Junos® OS

MPLS Feature Guide for Security Devices

Release

15.1X49-D70

Modified: 2016-11-23

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

Junos® OS MPLS Feature Guide for Security Devices
15.1X49-D70

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

YEAR 2000 NOTICE

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

END USER LICENSE AGREEMENT

Table of Contents

Part 1 Overview

Part 2 Configuring Traffic Engineering

Part 3 Configuring MPLS VPNs

Part 4 Configuring CLNS VPNs

Part 5 Configuring VPLS

Part 6 Configuration Statements and Operational Commands

List of Figures

Part 1 Overview

Part 2 Configuring Traffic Engineering

Part 3 Configuring MPLS VPNs

Part 5 Configuring VPLS

List of Tables

Part 6 Configuration Statements and Operational Commands

About the Documentation

Documentation and Release Notes

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

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

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

Supported Platforms

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

- SRX345
- SRX340
- SRX320
- SRX300
- SRX550M
- SRX1500
- vSRX

Using the Examples in This Manual

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

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

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

Merging a Full Example

To merge a full example, follow these steps:

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

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

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

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

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

Merging a Snippet

To merge a snippet, follow these steps:

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

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

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

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

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

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

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

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

Documentation Conventions

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

Table 1: Notice Icons

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

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

Table 2: Text and Syntax Conventions

Convention	Description	Examples
Bold text like this	Represents text that you type.	To enter configuration mode, type the **configure** command: user@host> **configure**
`Fixed-width text like this`	Represents output that appears on the terminal screen.	`user@host>` **show chassis alarms** `No alarms currently active`
Italic text like this	• Introduces or emphasizes important new terms. • Identifies guide names. • Identifies RFC and Internet draft titles.	• A policy *term* is a named structure that defines match conditions and actions. • *Junos OS CLI User Guide* • RFC 1997, *BGP Communities Attribute*
Italic text like this	Represents variables (options for which you substitute a value) in commands or configuration statements.	Configure the machine's domain name: [edit] root@# **set system domain-name** *domain-name*

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

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

```
[edit]
routing-options {
  static {
    route default {
      nexthop address;
      retain;
    }
  }
}
```

GUI Conventions

Convention	Description	Examples
Bold text like this	Represents graphical user interface (GUI) items you click or select.	• In the Logical Interfaces box, select **All Interfaces**. • To cancel the configuration, click **Cancel**.
> (bold right angle bracket)	Separates levels in a hierarchy of menu selections.	In the configuration editor hierarchy, select **Protocols>Ospf**.

Documentation Feedback

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

- Online feedback rating system—On any page of the Juniper Networks TechLibrary site at http://www.juniper.net/techpubs/index.html, simply click the stars to rate the content, and use the pop-up form to provide us with information about your experience. Alternately, you can use the online feedback form at http://www.juniper.net/techpubs/feedback/.

- E-mail—Send your comments to **techpubs-comments@juniper.net**. Include the document or topic name, URL or page number, and software version (if applicable).

Requesting Technical Support

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

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

Self-Help Online Tools and Resources

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

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

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

Opening a Case with JTAC

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

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

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

PART 1

Overview

CHAPTER 1

Introduction to MPLS

MPLS Overview

Supported Platforms SRX1500, SRX300, SRX320, SRX340, SRX345, SRX550M, vSRX

Multiprotocol Label Switching (MPLS) is a method for engineering traffic patterns by assigning short labels to network packets that describe how to forward them through the network. MPLS is independent of routing tables or any routing protocol and can be used for unicast packets.

The MPLS framework supports traffic engineering and the creation of virtual private networks (VPNs). Traffic is engineered (controlled) primarily by the use of signaling protocols to establish label-switched paths (LSPs). VPN support includes Layer 2 and Layer 3 VPNs and Layer 2 circuits.

When you enable your device to allow MPLS traffic, the device performs packet-based processing and functions as a standard Junos router.

CAUTION: When packet forwarding mode is changed to MPLS, all flow-based security features are deactivated, and the device performs packet-based processing only. Flow-based services such as security policies, zones, NAT, ALGs, chassis clustering, screens, firewall authentication, and IPsec VPNs are unavailable on the device. However, MPLS can be enabled in flow-based packet forwarding mode for selected traffic using firewall filters.

This overview contains the following topics:

Label Switching

In a traditional IP network, packets are transmitted with an IP header that includes a source and destination address. When a router receives such a packet, it examines its forwarding tables for the next-hop address associated with the packet's destination address and forwards the packet to the next-hop location.

In an MPLS network, each packet is encapsulated with an MPLS header. When a router receives the packet, it copies the header as an index into a separate MPLS forwarding table. The MPLS forwarding table consists of pairs of inbound interfaces and path information. Each pair includes forwarding information that the router uses to forward the traffic and modify, when necessary, the MPLS header.

Because the MPLS forwarding table has far fewer entries than the more general forwarding table, the lookup consumes less processing time and processing power. The resultant savings in time and processing are a significant benefit for traffic that uses the network to transit between outside destinations only.

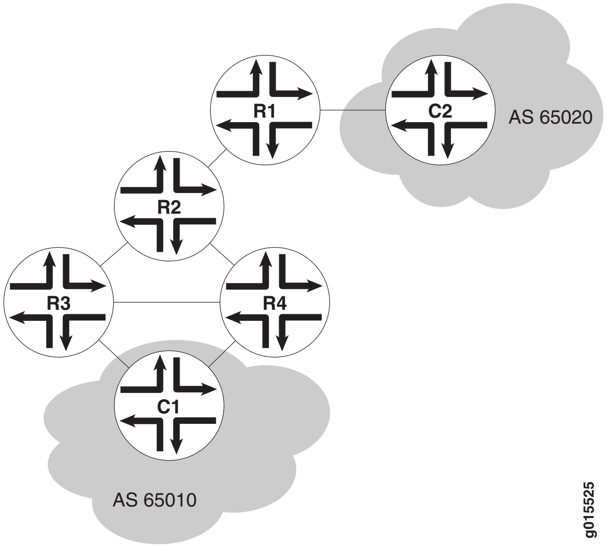
Label-Switched Paths

Label-switched paths (LSPs) are unidirectional routes through a network or autonomous system (AS). In normal IP routing, the packet has no predetermined path. Instead, each router forwards a packet to the next-hop address stored in its forwarding table, based only on the packet's destination address. Each subsequent router then forwards the packet using its own forwarding table.

In contrast, MPLS routers within an AS determine paths through a network through the exchange of MPLS traffic engineering information. Using these paths, the routers direct traffic through the network along an established route. Rather than selecting the next hop along the path as in IP routing, each router is responsible for forwarding the packet to a predetermined next-hop address.

Figure 1 on page 5 shows a typical LSP topology.

Figure 1: Typical LSP Topology



In the topology shown in Figure 1 on page 5, traffic is forwarded from Host C1 to the transit network with standard IP forwarding. When the traffic enters the transit network, it is switched across a preestablished LSP through the network. In this example, an LSP might switch the traffic from Router R4 to Router R2 to Router R1. When the traffic exits the network, it is forwarded to its destination by IP routing protocols.

Label-Switching Routers

Routers that are part of the LSP are label-switching routers (LSRs). Each LSR must be configured with MPLS so that it can interpret MPLS headers and perform the MPLS operations required to pass traffic through the network. An LSP can include four types of LSRs:

- Inbound router—The only entry point for traffic into MPLS. Native IPv4 packets are encapsulated into the MPLS protocol by the inbound router. Each LSP can have only one inbound router. Inbound routers are also known as ingress routers.
- Transit router—Any router in the middle of an LSP. An individual LSP can contain between 0 and 253 transit routers. Transit routers forward MPLS traffic along the LSP, using only the MPLS header to determine how the packet is routed.
- Penultimate router—The second-to-last router in the LSP. The penultimate router in an LSP is responsible for stripping the MPLS header from the packet before forwarding it to the outbound router.
- Outbound router—The endpoint for the LSP. The outbound router receives MPLS packets from the penultimate router and performs an IP route lookup. The router then forwards the packet to the next hop of the route. Each LSP can have only one outbound router. Outbound routers are also known as egress routers.

Labels

To forward traffic through an MPLS network, MPLS routers encapsulate packets and assign and manage headers known as *labels*. A label is a 20–bit unsigned integer in the range 0 through 1,048,575. The routers use the labels to index the MPLS forwarding tables that determine how packets are routed through the network.

When a network's inbound router receives traffic, it inserts an MPLS label between the IP packet and the appropriate Layer 2 header for the physical link. The label contains an index value that identifies a next-hop address for the particular LSP. When the next-hop transit router receives the packet, it uses the index in the MPLS label to determine the next-hop address for the packet and forwards the packet to the next router in the LSP.

As each packet travels through the transit network, every router along the way performs a lookup on the MPLS label and forwards the packet accordingly. When the outbound router receives a packet, it examines the header to determine that it is the final router in the LSP. The outbound router then removes the MPLS header, performs a regular IP route lookup, and forwards the packet with its IP header to the next-hop address.

Label Operations

Each LSR along an LSP is responsible for examining the MPLS label, determining the LSP next hop, and performing the required label operations. LSRs can perform five label operations:

- Push—Adds a new label to the top of the packet. For IPv4 packets arriving at the inbound router, the new label is the first label in the label stack. For MPLS packets with an existing label, this operation adds a label to the stack and sets the stacking bit to 0, indicating that more MPLS labels follow the first.

  When it receives the packet, the inbound router performs an IP route lookup on the packet. Because the route lookup yields an LSP next hop, the inbound router performs a label push on the packet, and then forwards the packet to the LSP next hop.

- Swap—Replaces the label at the top of the label stack with a new label.

  When a transit router receives the packet, it performs an MPLS forwarding table lookup. The lookup yields the LSP next hop and the path index of the link between the transit router and the next router in the LSP.

- Pop—Removes the label from the top of the label stack. For IPv4 packets arriving at the penultimate router, the entire MPLS label is removed from the label stack. For MPLS packets with an existing label, this operation removes the top label from the label stack and modifies the stacking bit as necessary—sets it to 1, for example, if only a single label remains in the stack.

  If multiple LSPs terminate at the same outbound router, the router performs MPLS label operations for all outbound traffic on the LSPs. To share the operations among multiple routers, most LSPs use penultimate hop popping (PHP).

- Multiple push—Adds multiple labels to the top of the label stack. This action is equivalent to performing multiple push operations.

The multiple push operation is used with label stacking, which is beyond the scope of this topic.

- Swap and push—Replaces the top label with a new label and then pushes a new label to the top of the stack.

  The swap and push operation is used with label stacking, which is beyond the scope of this topic.

Penultimate Hop Popping

Multiple LSPs terminating at a single outbound router load the router with MPLS label operations for all their outbound traffic. Penultimate hop popping (PHP) transfers the operation from the outbound router to penultimate routers.

With PHP, the penultimate router is responsible for popping the MPLS label and forwarding the traffic to the outbound router. The outbound router then performs an IP route lookup and forwards the traffic. For example, if four LSPs terminate at the same outbound router and each has a different penultimate router, label operations are shared across four routers.

LSP Establishment

An MPLS LSP is established by one of two methods: static LSPs and dynamic LSPs.

Static LSPs

Like a static route, a static LSP requires each router along the path to be configured explicitly. You must manually configure the path and its associated label values. Static LSPs require less processing by the LSRs because no signaling protocol is used. However, because paths are statically configured, they cannot adapt to network conditions. Topology changes and network outages can create black holes in the LSP that exist until you manually reconfigure the LSP.

Dynamic LSPs

Dynamic LSPs use signaling protocols to establish themselves and propagate LSP information to other LSRs in the network. You configure the inbound router with LSP information that is transmitted throughout the network when you enable the signaling protocols across the LSRs. Because the LSRs must exchange and process signaling packets and instructions, dynamic LSPs consume more resources than static LSPs. However, dynamic LSPs can avoid the network black holes of static LSPs by detecting topology changes and outages and propagating them throughout the network.

Related Documentation

- MPLS Configuration Overview on page 8
- Example: Deleting Security Services on page 8
- Example: Enabling MPLS on page 10
- MPLS Traffic Engineering and Signaling Protocols Overview on page 15
- MPLS VPN Overview on page 53

MPLS Configuration Overview

Supported Platforms SRX1500, SRX300, SRX320, SRX340, SRX345, SRX550M, vSRX

When you first install Junos OS on your device, MPLS is disabled by default. You must explicitly configure your device to allow MPLS traffic to pass through. Complete the following steps for all devices in your MPLS network that are running Junos OS.

To enable MPLS:

1. Delete all configured security services from the device. If you do not complete this step, you will get a commit failure. See "Example: Deleting Security Services" on page 8.
2. Enable MPLS on the device. See "Example: Enabling MPLS" on page 10.
3. Commit the configuration.
4. Reboot the device.
5. Configure MPLS features such as traffic engineering, VPNs, and VPLS. See:
   - MPLS Traffic Engineering and Signaling Protocols Overview on page 15
   - MPLS VPN Overview on page 53
   - CLNS Overview on page 77
   - VPLS Overview on page 103

CAUTION: When packet forwarding mode is changed to MPLS, all flow-based security features are deactivated, and the device performs packet-based processing only. Flow-based services such as security policies, zones, NAT, ALGs, chassis clustering, screens, firewall authentication, and IPsec VPNs are unavailable on the device. However, MPLS can be enabled in flow-based packet forwarding mode for selected traffic using firewall filters.

Related Documentation

- MPLS Overview on page 3
- Example: Deleting Security Services on page 8
- Example: Enabling MPLS on page 10

Example: Deleting Security Services

Supported Platforms SRX1500, SRX300, SRX320, SRX340, SRX345, SRX550M, vSRX

This example shows how to delete configured services in the security level of the configuration hierarchy.

- Requirements on page 9
- Overview on page 9

- Configuration on page 9
- Verification on page 9

Requirements

Before you begin, save your current configuration to a temporary file. Do this prior to removing all configurations from the security level of the configuration hierarchy and deleting the inherited configurations.

Overview

In this example, you save your current configuration in the var/tmp/ directory with an appropriate filename and .cfg extension—for example, *curfeb08.cfg*. Then you remove all configurations from the **security** level of the configuration hierarchy, and delete all global groups and inherited configurations.

Configuration

Step-by-Step Procedure

To delete the configured services in the security level of the configuration hierarchy:

1. Save your current configuration.

```
[edit]
user@host# save /var/tmp/curfeb08.cfg
```

2. Remove all configurations in the **security** level of the configuration hierarchy.

```
[edit]
user@host# delete security
```

3. Remove all inherited configurations in the security level of the configuration hierarchy.

```
[edit]
user@host# delete groups global security
```

CAUTION: Do not commit after deleting the security configurations. A commit without any security configurations leaves the router unreachable through the management port.

Verification

To verify the configuration is working properly, enter the **show groups global security** command.

Related Documentation

- MPLS Overview on page 3
- MPLS Configuration Overview on page 8
- Example: Enabling MPLS on page 10

Example: Enabling MPLS

Supported Platforms SRX1500, SRX300, SRX320, SRX340, SRX345, SRX550M, vSRX

This example shows how to enable MPLS for packet-based processing. It also shows how to enable the MPLS family and MPLS process on all of the transit interfaces in the network.

NOTE: When MPLS is enabled, all flow-based security features are deactivated and the device performs packet-based processing. Flow-based services such as security policies, zones, NAT, ALGs, chassis clustering, screens, firewall authentication, IP packets, and IPsec VPNs are unavailable on the device.

Before changing from flow mode to packet mode, you must remove all security policies remaining under flow mode. To prevent management connection loss, you must bind the management interface to zones and enable host-inbound traffic to prevent the device from losing connectivity.

For information about configuring zones, see *Building Blocks Feature Guide for Security Devices*.

Requirements

Before you begin, delete all configured security services. See "Example: Deleting Security Services" on page 8.

Overview

The instructions in this topic describe how to enable MPLS on the device. You must enable MPLS on the device before including a device running Junos OS in an MPLS network.

Configuration

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

```
set security forwarding-options family mpls mode packet-based
set interfaces ge-1/0/0 unit 0 family mpls
set protocols mpls ge-1/0/0 unit 0
```

Step-by-Step Procedure The following example requires you to navigate various levels in the configuration hierarchy. For instructions on how to do that, see *Using the CLI Editor in Configuration Mode*.

To enable MPLS:

1. Enable MPLS for packet-based processing.

```
[edit security forwarding-options]
user@host# set family mpls mode packet-based
```

2. Enable the MPLS family on each transit interface that you want to include in the MPLS network.

```
[edit interfaces]
user@host# set interfaces ge-1/0/0 unit 0 family mpls
```

3. Enable the MPLS process on all of the transit interfaces in the MPLS network.

```
[edit protocols mpls]
user@host# set interface ge-1/0/0 unit 0
```

Results From configuration mode, confirm your configuration by entering the **show security forwarding-options** command. If the output does not display the intended configuration, repeat the configuration instructions in this example to correct it.

NOTE: If you enable MPLS for packet-based processing by using the command **set security forward-option family mpls mode packet**, the mode will not change immediately and the system will display the following messages:

warning: Reboot may required when try reset flow inet mode

warning: Reboot may required when try reset mpls flow mode please check security flow status for detail.

You need to reboot your device for the configuration to take effect.

CAUTION: If you disable MPLS and switch back to using the security services (flow-based processing), the mode will not change immediately and the system will display warning messages instructing you to restart your device. You must reboot your device for the configuration to take effect. This will also result in management sessions being reset and transit traffic getting interrupted.

```
[edit]
user@host# show security forwarding-options
family {
  mpls {
    mode packet-based;
  }
}
```

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

Verification

Confirm that the configuration is working properly.

- Verifying MPLS Is Enabled at the Protocols Level on page 12
- Verifying MPLS Is Enabled at the Interfaces Level on page 12

Verifying MPLS Is Enabled at the Protocols Level

Purpose Verify that MPLS is enabled at the protocols level.

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

Verifying MPLS Is Enabled at the Interfaces Level

Purpose Verify that MPLS is enabled at the interfaces level.

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

Related Documentation

- MPLS Overview on page 3
- MPLS Configuration Overview on page 8
- Example: Deleting Security Services on page 8

PART 2

Configuring Traffic Engineering

CHAPTER 2

Configuring MPLS Traffic Engineering and Signaling Protocols

MPLS Traffic Engineering and Signaling Protocols Overview

Supported Platforms	SRX100, SRX110, SRX210, SRX220, SRX240, SRX650

Traffic engineering facilitates efficient and reliable network operations while simultaneously optimizing network resources and traffic performance. Traffic engineering provides the ability to move traffic flow away from the shortest path selected by the interior gateway protocol (IGP) to a potentially less congested physical path across a network. To support traffic engineering, besides source routing, the network must do the following:

- Compute a path at the source by taking into account all the constraints, such as bandwidth and administrative requirements.
- Distribute the information about network topology and link attributes throughout the network once the path is computed.
- Reserve network resources and modify link attributes.

When transit traffic is routed through an IP network, MPLS is often used to engineer its passage. Although the exact path through the transit network is of little importance to either the sender or the receiver of the traffic, network administrators often want to route traffic more efficiently between certain source and destination address pairs. By adding a short label with specific routing instructions to each packet, MPLS switches packets from router to router through the network rather than forwarding packets based on next-hop lookups. The resulting routes are called *label-switched paths (LSPs)*. LSPs control the passage of traffic through the network and speed traffic forwarding.

You can create LSPs manually, or through the use of signaling protocols. Signaling protocols are used within an MPLS environment to establish LSPs for traffic across a transit network. Junos OS supports two signaling protocols—LDP and the Resource Reservation Protocol (RSVP).

MPLS traffic engineering uses the following components:

- MPLS LSPs for packet forwarding
- IGP extensions for distributing information about the network topology and link attributes
- Constrained Shortest Path First (CSPF) for path computation and path selection
- RSVP extensions to establish the forwarding state along the path and to reserve resources along the path

Junos OS also supports traffic engineering across different OSPF regions.

Related Documentation

- MPLS Applications Feature Guide for Routing Devices
- Understanding the LDP Signaling Protocol on page 16
- Understanding the RSVP Signaling Protocol on page 21
- Understanding Point-to-Multipoint LSPs on page 29

Understanding the LDP Signaling Protocol

Supported Platforms SRX1500, SRX300, SRX320, SRX340, SRX345, SRX550M, vSRX

LDP is a signaling protocol that runs on a device configured for MPLS support. The successful configuration of both MPLS and LDP initiates the exchange of TCP packets across the LDP interfaces. The packets establish TCP-based LDP sessions for the exchange of MPLS information within the network. Enabling both MPLS and LDP on the appropriate interfaces is sufficient to establish LSPs.

LDP is a simple, fast-acting signaling protocol that automatically establishes LSP adjacencies within an MPLS network. Routers then share LSP updates such as hello packets and LSP advertisements across the adjacencies. Because LDP runs on top of an IGP such as IS-IS or OSPF, you must configure LDP and the IGP on the same set of interfaces. After both are configured, LDP begins transmitting and receiving LDP messages through all LDP-enabled interfaces. Because of LDP's simplicity, it cannot perform the true traffic engineering which RSVP can perform. LDP does not support bandwidth reservation or traffic constraints.

When you configure LDP on a label-switching router (LSR), the router begins sending LDP discovery messages out all LDP-enabled interfaces. When an adjacent LSR receives LDP discovery messages, it establishes an underlying TCP session. An LDP session is then created on top of the TCP session. The TCP three-way handshake ensures that the

LDP session has bidirectional connectivity. After they establish the LDP session, the LDP neighbors maintain, and terminate, the session by exchanging messages. LDP advertisement messages allow LSRs to exchange label information to determine the next hops within a particular LSP. Any topology changes, such as a router failure, generate LDP notifications that can terminate the LDP session or generate additional LDP advertisements to propagate an LSP change.

Related Documentation

- MPLS Traffic Engineering and Signaling Protocols Overview on page 15
- Example: Configuring LDP-Signaled LSPs on page 17

Example: Configuring LDP-Signaled LSPs

Supported Platforms SRX1500, SRX300, SRX320, SRX340, SRX345, SRX550M, vSRX

This example shows how to create and configure LDP instances within an MPLS network.

- Requirements on page 17
- Overview on page 17
- Configuration on page 18
- Verification on page 19

Requirements

Before you begin:

- Configure network interfaces. See *Interfaces Feature Guide for Security Devices*.
- Configure an IGP across your network. (The LDP configuration is added to the existing IGP configuration and included in the MPLS configuration.)
- Configure a network to use LDP for LSP establishment by enabling MPLS on all transit interfaces in the MPLS network.

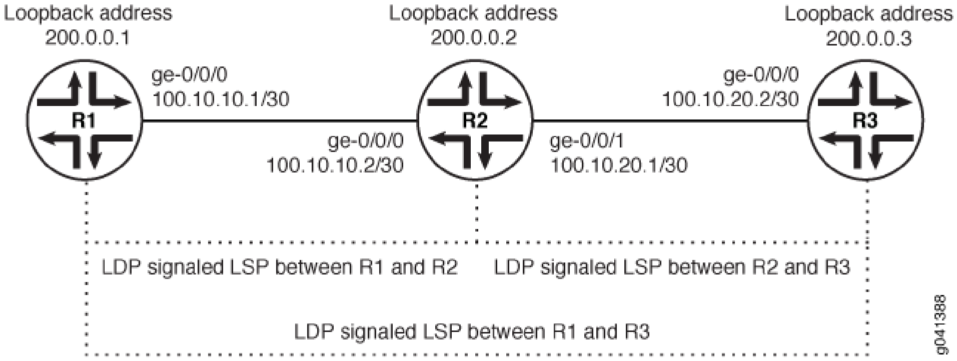
NOTE: **Because LDP runs on top of an IGP such as IS-IS or OSPF, you must configure LDP and the IGP on the same set of interfaces.**

Overview

To configure LDP-signaled LSPs, you must enable the MPLS family on all transit interfaces in the MPLS network and include all the transit interfaces under the [**protocols mpls**] and [**protocols ldp**] hierarchy levels.

In this example, you enable the MPLS family and create an LDP instance on all the transit interfaces. Additionally, you enable the MPLS process on all the transit interfaces in the MPLS network. In this example, you configure a sample network as shown in Figure 2 on page 18.

Figure 2: Typical LDP-Signaled LSP



Configuration

CLI Quick Configuration

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

For Router R1, perform the following:

```
set interfaces ge-0/0/0 unit 0 family mpls
set protocols mpls ge-0/0/0 unit 0
set protocols ldp interface ge-0/0/0.0 unit 0
```

For Router R2, perform the following:

```
set interfaces ge-0/0/0 unit 0 family mpls
set protocols mpls ge-0/0/0 unit 0
set protocols ldp interface ge-0/0/0.0 unit 0
set interfaces ge-0/0/1 unit 0 family mpls
set protocols mpls ge-0/0/1 unit 0
set protocols ldp interface ge-0/0/0.1 unit 0
```

For Router R3, perform the following:

```
set interfaces ge-0/0/0 unit 0 family mpls
set protocols mpls ge-0/0/0 unit 0
set protocols ldp interface ge-0/0/0.0 unit 0
```

Step-by-Step Procedure

To enable LDP instances within an MPLS network:

1. Enable the MPLS family on the transit interface on Router R1.

```
[edit]
user@R1# set interfaces ge-0/0/0 unit 0 family mpls
```

2. Enable the MPLS process on the transit interface.

```
[edit]
user@R1# set protocols mpls interface ge-0/0/0 unit 0
```

3. Create the LDP instance on the transit interface.

```
[edit]
user@R1# set protocols ldp interface ge-0/0/0 unit 0
```

Results Confirm your configuration by entering the **show** command from configuration mode. If the output does not display the intended configuration, repeat the configuration instructions in this example to correct it.

For brevity, this **show** output includes only the configuration that is relevant to this example. Any other configuration on the system has been replaced with ellipses (...).

```
user@R1# show
...
  interfaces {
    ge-0/0/0 {
      unit 0 {
        family inet {
          address 10.100.37.20/24;
        }
        family mpls;
      }
    }
    }
  ...
    protocols {
      mpls {
        interface all;
      }
      ldp {
        interface ge-0/0/0.0;
      }
    }
```

If you are done configuring the device, enter the **commit** command from the configuration mode to activate the configuration.

Verification

Confirm that the configuration is working properly.

- Verifying LDP Neighbors on page 19
- Verifying LDP Sessions on page 20
- Verifying the Presence of LDP-Signaled LSPs on page 21
- Verifying Traffic Forwarding over the LDP-Signaled LSP on page 21

Verifying LDP Neighbors

Supported Platforms SRX1500, SRX300, SRX320, SRX340, SRX345

Purpose Verify that each device shows the appropriate LDP neighbors.

Action From the CLI, enter the **show ldp neighbor** command.

```
user@r2> show ldp neighbor
```

```
Address      Interface        Label space ID      Hold time
100.10.10.1     ge-0/0/0.0      200.0.0.1:0           11
100.10.20.2    ge-0/0/1.0      200.0.0.2:0            14
```

The output shows the IP addresses of the neighboring interfaces along with the interface through which the neighbor adjacency is established. Verify the following information:

- Each interface on which LDP is enabled is listed.
- Each neighboring LDP interface address is listed with the appropriate corresponding LDP interface.
- Under **Label space ID**, the appropriate loopback address for each neighbor appears.

Verifying LDP Sessions

Supported Platforms SRX1500, SRX300, SRX320, SRX340, SRX345

Purpose Verify that a TCP-based LDP session has been established between all LDP neighbors. Also, verify that the modified keepalive value is active.

Action From the CLI, enter the **show ldp session detail** command.

```
user@r1> show ldp session detail
200.0.0.2, State: Operational, Connection: Open, Hold time: 22
  Session ID: 200.0.0.1:0--200.0.0.2:0
  Next keepalive in 9 seconds
  Active, Maximum PDU: 4096, Hold time: 30, Neighbor count: 1
  Neighbor types: discovered
  Keepalive interval: 10, Connect retry interval: 1
  Local address: 200.0.0.1, Remote address: 200.0.0.2
  Up for 01:58:49
  Capabilities advertised: none
  Capabilities received: none
  Protection: disabled
  Local - Restart: disabled, Helper mode: enabled
  Remote - Restart: disabled, Helper mode: enabled
  Local maximum neighbor reconnect time: 120000 msec
  Local maximum neighbor recovery time: 240000 msec
  Nonstop routing state: Not in sync
  Next-hop addresses received:
 100.10.10.2
 100.10.20.1
```

The output shows the detailed information, including session IDs, keepalive interval, and next-hop addresses, for each established LDP session. Verify the following information:

- Each LDP neighbor address has an entry, listed by loopback address.
- The state for each session is **Operational**, and the connection for each session is **Open**. A state of **Nonexistent** or a connection of **Closed** indicates a problem with one of the following:
  - LDP configuration
  - Passage of traffic between the two devices
  - Physical link between the two routers

- For **Keepalive interval**, the appropriate value, **10**, appears.

Verifying the Presence of LDP-Signaled LSPs

Supported Platforms SRX1500, SRX300, SRX320, SRX340, SRX345

Purpose Verify that each Juniper Networks device's **inet.3** routing table has an LSP for the loopback address on each of the other routers.

Action From the CLI, enter the **show route table inet.3** command.

```
user@r1> run show route table inet.3
inet.3: 2 destinations, 2 routes (2 active, 0 holddown, 0 hidden)
+ = Active Route, - = Last Active, * = Both
200.0.0.2/32       *[LDP/9] 05:20:20, metric 1
                    > via ge-0/0/0.0, Push 300640
200.0.0.3/32       *[LDP/9] 05:20:20, metric 1
                    > via ge-0/0/0.0, Push 300704
```

The output shows the LDP routes that exist in the **inet.3** routing table. Verify that an LDP-signaled LSP is associated with the loopback addresses of the other routers in the MPLS network.

Verifying Traffic Forwarding over the LDP-Signaled LSP

Supported Platforms SRX1500, SRX300, SRX320, SRX340, SRX345

Purpose Verify that the LDP path between R1 and R3 is complete over the LDP-signaled LSP.

Action From the CLI on R1, enter the **traceroute mpls ldp 200.0.0.3** command.

```
user@r1> traceroute mpls ldp 200.0.0.3
Probe options: ttl 64, retries 3, wait 10, paths 16, exp 7, fanout 16
ttl Label Protocol Address Previous Hop Probe Status
1 300704 LDP 200.0.0.1 (null) Unhelpful
2 200.0.0.2 200.0.0.1 Unhelpful
3 200.0.0.3 200.0.0.2 Egress
Path 1 via ge-0/0/0.0 destination 127.0.0.64
```

The output shows the path between R1 and R3 using an LDP signalled LSP.

Related Documentation

- Understanding the LDP Signaling Protocol on page 16
- Routing Protocols Overview

Understanding the RSVP Signaling Protocol

Supported Platforms SRX1500, SRX300, SRX320, SRX340, SRX345, SRX550M, vSRX

RSVP is a signaling protocol that handles bandwidth allocation and true traffic engineering across an MPLS network. Like LDP, RSVP uses discovery messages and advertisements to exchange LSP path information between all hosts. However, RSVP also includes a set of features that control the flow of traffic through an MPLS network. Whereas LDP is restricted to using the configured IGP's shortest path as the transit path through the network, RSVP uses a combination of the Constrained Shortest Path First (CSPF) algorithm and Explicit Route Objects (EROs) to determine how traffic is routed through the network.

Basic RSVP sessions are established in exactly the same way that LDP sessions are established. By configuring both MPLS and RSVP on the appropriate transit interfaces, you enable the exchange of RSVP packets and the establishment of LSPs. However, RSVP also lets you configure link authentication, explicit LSP paths, and link coloring.

This topic contains the following sections:

RSVP Fundamentals

RSVP uses unidirectional and simplex flows through the network to perform its function. The inbound router initiates an RSVP path message and sends it downstream to the outbound router. The path message contains information about the resources needed for the connection. Each router along the path begins to maintain information about a reservation.

When the path message reaches the outbound router, resource reservation begins. The outbound router sends a reservation message upstream to the inbound router. Each router along the path receives the reservation message and sends it upstream, following the path of the original path message. When the inbound router receives the reservation message, the unidirectional network path is established.

The established path remains open as long as the RSVP session is active. The session is maintained by the transmission of additional path and reservation messages that report the session state every 30 seconds. If a router does not receive the maintenance messages for 3 minutes, it terminates the RSVP session and reroutes the LSP through another active router.

Bandwidth Reservation Requirement

When a bandwidth reservation is configured, reservation messages propagate the bandwidth value throughout the LSP. Routers must reserve the bandwidth specified across the link for the particular LSP. If the total bandwidth reservation exceeds the available bandwidth for a particular LSP segment, the LSP is rerouted through another LSR. If no segments can support the bandwidth reservation, LSP setup fails and the RSVP session is not established.

Explicit Route Objects

Explicit Route Objects (EROs) limit LSP routing to a specified list of LSRs. By default, RSVP messages follow a path that is determined by the network IGP's shortest path.

However, in the presence of a configured ERO, the RSVP messages follow the path specified.

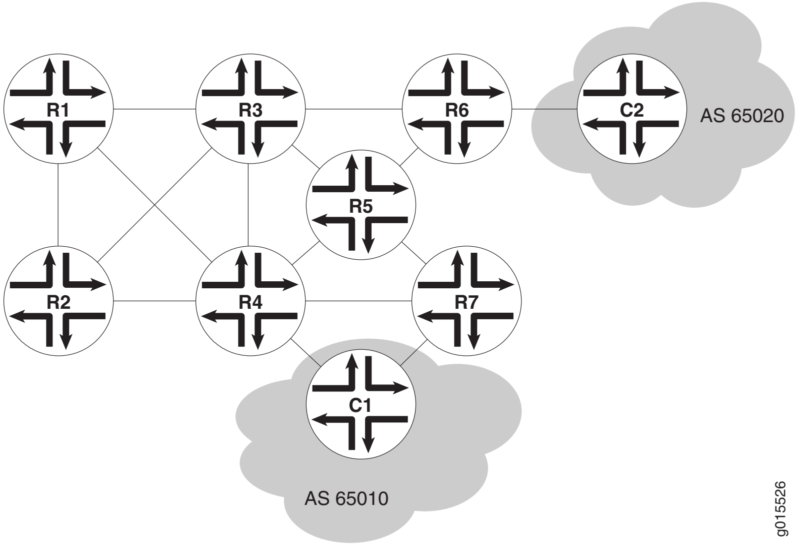
EROs consist of two types of instructions: loose hops and strict hops. When a loose hop is configured, it identifies one or more transit LSRs through which the LSP must be routed. The network IGP determines the exact route from the inbound router to the first loose hop, or from one loose hop to the next. The loose hop specifies only that a particular LSR be included in the LSP.

When a strict hop is configured, it identifies an exact path through which the LSP must be routed. Strict-hop EROs specify the exact order of the routers through which the RSVP messages are sent.

You can configure loose-hop and strict-hop EROs simultaneously. In this case, the IGP determines the route between loose hops, and the strict-hop configuration specifies the exact path for particular LSP path segments.

Figure 3 on page 23 shows a typical RSVP-signaled LSP that uses EROs.

Figure 3: Typical RSVP-Signaled LSP with EROs



In the topology shown in Figure 3 on page 23, traffic is routed from Host C1 to Host C2. The LSP can pass through Routers R4 or Router R7. To force the LSP to use R4, you can set up either a loose-hop or strict-hop ERO that specifies R4 as a hop in the LSP. To force a specific path through Router R4, R3, and R6, configure a strict-hop ERO through the three LSRs.

Constrained Shortest Path First

Whereas IGPs use the Shortest Path First (SPF) algorithm to determine how traffic is routed within a network, RSVP uses the Constrained Shortest Path First (CSPF) algorithm to calculate traffic paths that are subject to the following constraints:

- LSP attributes—Administrative groups such as link coloring, bandwidth requirements, and EROs

- Link attributes—Colors on a particular link and available bandwidth

These constraints are maintained in the traffic engineering database (TED). The database provides CSPF with up-to-date topology information, the current reservable bandwidth of links, and the link colors.

In determining which path to select, CSPF follows these rules:

- Computes LSPs one at a time, beginning with the highest priority LSP—the one with the lowest setup priority value. Among LSPs of equal priority, CSPF starts with those that have the highest bandwidth requirement.
- Prunes the traffic engineering database of links that are not full duplex and do not have sufficient reservable bandwidth.
- If the LSP configuration includes the **include** statement, prunes all links that do not share any included colors.
- If the LSP configuration includes the **exclude** statement, prunes all links that contain excluded colors. If a link does not have a color, it is accepted.
- Finds the shortest path toward the LSP's outbound router, taking into account any EROs. For example, if the path must pass through Router A, two separate SPF algorithms are computed: one from the inbound router to Router A and one from Router A to the outbound router.
- If several paths have equal cost, chooses the one with a last-hop address the same as the LSP's destination.
- If several equal-cost paths remain, selects the path with the least number of hops.
- If several equal-cost paths remain, applies CSPF load-balancing rules configured on the LSP.

Link Coloring

RSVP allows you to configure administrative groups for CSPF path selection. An administrative group is typically named with a color, assigned a numeric value, and applied to the RSVP interface for the appropriate link. Lower numbers indicate higher priority.

After configuring the administrative group, you can either exclude, include, or ignore links of that color in the TED:

- If you exclude a particular color, all segments with an administrative group of that color are excluded from CSPF path selection.
- If you include a particular color, only those segments with the appropriate color are selected.
- If you neither exclude nor include the color, the metrics associated with the administrative groups and applied on the particular segments determine the path cost for that segment.

The LSP with the lowest total path cost is selected into the TED.

Related Documentation

- MPLS Traffic Engineering and Signaling Protocols Overview on page 15
- Example: Configuring RSVP-Signaled LSPs on page 25

Example: Configuring RSVP-Signaled LSPs

Supported Platforms SRX100, SRX110, SRX210, SRX220, SRX240, SRX650

This example shows how to create an LSP between routers in an IP network using RSVP as the signaling protocol.

- Requirements on page 25
- Overview and Topology on page 25
- Configuration on page 26
- Verification on page 27

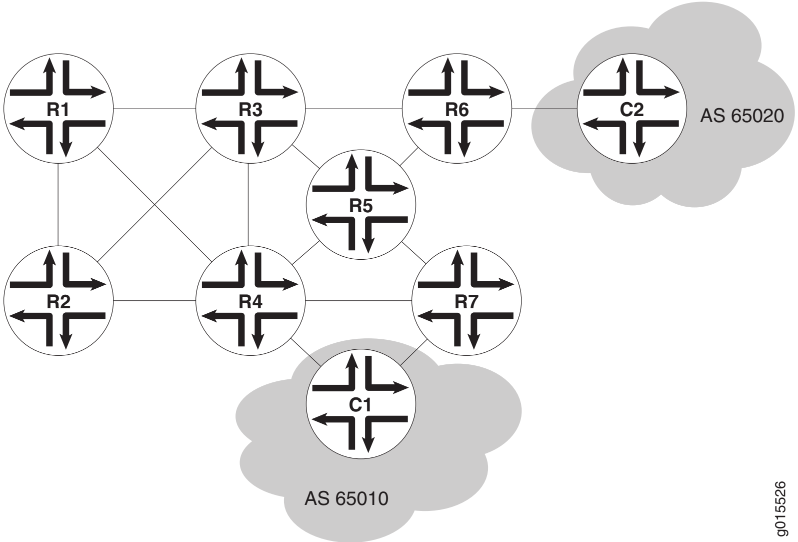
Requirements

Before you begin, delete security services from the device. See "Example: Deleting Security Services" on page 8.

Overview and Topology

Using RSVP as a signaling protocol, you can create LSPs between routers in an IP network. In this example, you configure a sample network as shown in Figure 4 on page 25.

Figure 4: Typical RSVP-Signaled LSP



To establish an LSP between routers, you must individually enable the MPLS family and configure RSVP on each of the transit interfaces in the MPLS network. This example shows how to enable MPLS and configure RSVP on the ge-0/0/0 transit interface. Additionally, you must enable the MPLS process on all of the MPLS interfaces in the network.

This example shows how to define an LSP from R1 to R7 on the ingress router (R1) using R7's loopback address (10.0.9.7). The configuration reserves 10 Mbps of bandwidth. Additionally, the configuration disables the CSPF algorithm, ensuring that Hosts C1 and C2 use the RSVP-signaled LSP that correspond to the network IGP's shortest path.

Configuration

CLI Quick Configuration

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

```
set interfaces ge-0/0/0 unit 0 family mpls
set protocols rsvp interface ge-0/0/0.0
set protocols mpls label-switched-path r1-r7 to 10.0.9.7
set protocols mpls label-switched-path r1-r7 bandwidth 10m
set protocols mpls interface all
```

Step-by-Step Procedure

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

To configure RSVP:

1. Enable the MPLS family on all transit interfaces in the MPLS network.

   ```
   [edit]
   user@host# set interfaces ge-0/0/0 unit 0 family mpls
   ```

2. Enable RSVP on each transit interface in the MPLS network.

   ```
   [edit]
   user @host# set protocols rsvp interface ge-0/0/0
   ```

3. Enable the MPLS process on the transit interface in the MPLS network.

   ```
   [edit]
   user@host# set protocols mpls interface ge-0/0/0
   ```

4. Define the LSP on the ingress router.

   ```
   [edit protocols mpls]
   user@host# set label-switched-path r1-r7 to 10.0.9.7
   ```

5. Reserve 10 Mbps of bandwidth on the LSP.

   ```
   [edit protocols mpls]
   user @host# set label-switched-path r1-r7 bandwidth 10m
   ```

Results

Confirm your configuration by entering the **show** command from configuration mode. If the output does not display the intended configuration, repeat the configuration instructions in this example to correct it.

For brevity, this **show** command output includes only the configuration that is relevant to this example. Any other configuration on the system has been replaced with ellipses (...).

```
user@host# show
...
  interfaces {
  ge-0/0/0 {
    family mpls;
    }
  }
  }
  ...
  protocols {
    rsvp {
      interface ge-0/0/0.0;
    }
    mpls {
      label-switched-path r1-r7 {
        to 10.0.9.7;
        bandwidth 10m;
      }
      interface all;
    }
  }
...
```

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

Verification

To confirm that the configuration is working properly, perform these tasks:

- Verifying RSVP Neighbors on page 27
- Verifying RSVP Sessions on page 27
- Verifying the Presence of RSVP-Signaled LSPs on page 28

Verifying RSVP Neighbors

Purpose Verify that each device shows the appropriate RSVP neighbors—for example, that Router R1 in Figure 4 on page 25 lists both Router R3 and Router R2 as RSVP neighbors.

Action From the CLI, enter the **show rsvp neighbor** command.

```
user@r1> show rsvp neighbor
RSVP neighbor: 2 learned
Address            Idle Up/Dn LastChange HelloInt HelloTx/Rx
10.0.6.2              0  3/2       13:01        3   366/349
10.0.3.3              0  1/0       22:49        3   448/448
```

The output shows the IP addresses of the neighboring routers. Verify that each neighboring RSVP router loopback address is listed.

Verifying RSVP Sessions

Purpose Verify that an RSVP session has been established between all RSVP neighbors. Also, verify that the bandwidth reservation value is active.

Action From the CLI, enter the **show rsvp session detail** command.

```
user@r1> show rsvp session detail
Ingress RSVP: 1 sessions

10.0.9.7
  From: 10.0.6.1, LSPstate: Up, ActiveRoute: 0
  LSPname: r1–r7, LSPpath: Primary
  Bidirectional, Upstream label in: –, Upstream label out: -
  Suggested label received: -, Suggested label sent: –
  Recovery label received: -, Recovery label sent: 100000
  Resv style: 1 FF, Label in: -, Label out: 100000
  Time left:    -,  Since: Thu Jan 26 17:57:45 2002
  Tspec: rate 10Mbps size 10Mbps peak Infbps m 20 M 1500
  Port number: sender 3 receiver 17 protocol 0
  PATH rcvfrom: localclient
  PATH sentto: 10.0.4.13 (ge-0/0/1.0) 1467 pkts
  RESV rcvfrom: 10.0.4.13  (ge-0/0/1.0) 1467 pkts
  Record route: <self> 10.0.4.13  10.0.2.1  10.0.8.10
```

The output shows the detailed information, including session IDs, bandwidth reservation, and next-hop addresses, for each established RSVP session. Verify the following information:

- Each RSVP neighbor address has an entry for each neighbor, listed by loopback address.
- The state for each LSP session is **Up**.
- For **Tspec**, the appropriate bandwidth value, **10Mbps**, appears.

Verifying the Presence of RSVP-Signaled LSPs

Purpose Verify that the routing table of the entry (ingress) router has a configured LSP to the loopback address of the other router. For example, verify that the **inet.3** routing table of the R1 entry router in Figure 4 on page 25 has a configured LSP to the loopback address of Router R7.

Action From the CLI, enter the **show route table inet.3** command.

```
user@r1> show route table inet.3
inet.3: 2 destinations, 2 routes (2 active, 0 holddown, 0 hidden)
+ = Active Route, - = Last Active, * = Both

10.0.9.7/32         *[RSVP/7] 00:05:29, metric 10
                    > to 10.0.4.17 via ge-0/0/0.0, label-switched-path r1–r7
```

The output shows the RSVP routes that exist in the **inet.3** routing table. Verify that an RSVP-signaled LSP is associated with the loopback address of the exit (egress) router, R7, in the MPLS network.

Related Documentation

- Understanding the RSVP Signaling Protocol on page 21
- CLI Explorer

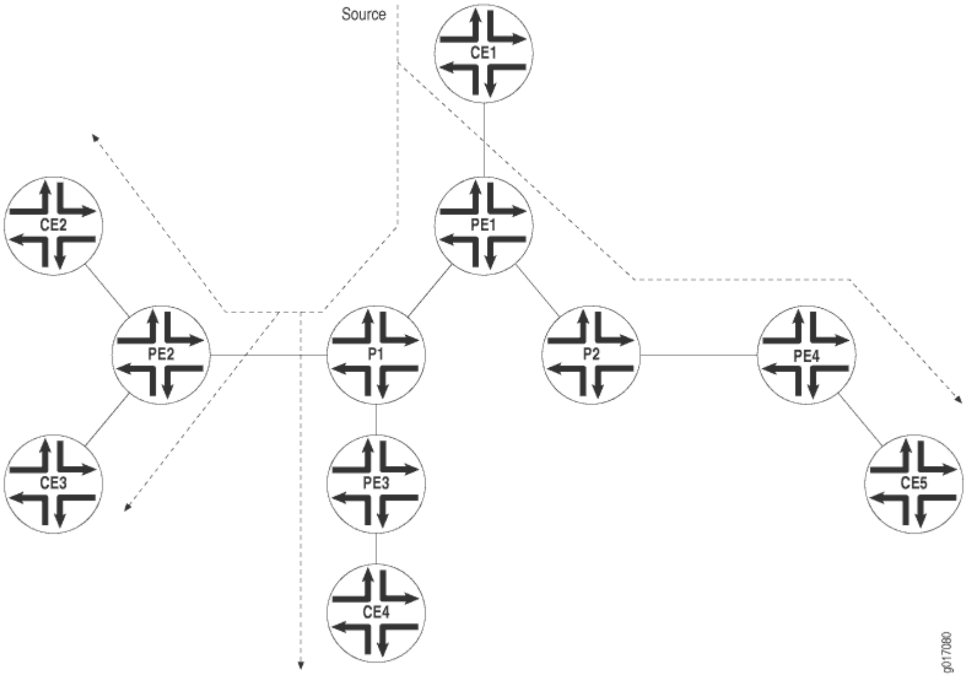
Understanding Point-to-Multipoint LSPs

Supported Platforms ACX Series, M Series, MX Series, SRX100, SRX110, SRX210, SRX220, SRX240, T Series

A point-to-multipoint MPLS label-switched path (LSP) is an LDP-signaled or RSVP-signaled LSP with a single source and multiple destinations. By taking advantage of the MPLS packet replication capability of the network, point-to-multipoint LSPs avoid unnecessary packet replication at the inbound (ingress) router. Packet replication takes place only when packets are forwarded to two or more different destinations requiring different network paths.

This process is illustrated in Figure 5 on page 29. Device PE1 is configured with a point-to-multipoint LSP to Routers PE2, PE3, and PE4. When Device PE1 sends a packet on the point-to-multipoint LSP to Routers P1 and P2, Device P1 replicates the packet and forwards it to Routers PE2 and PE3. Device P2 sends the packet to Device PE4.

Figure 5: Point-to-Multipoint LSPs



Following are some of the properties of point-to-multipoint LSPs:

- A point-to-multipoint LSP allows you to use MPLS for point-to-multipoint data distribution. This functionality is similar to that provided by IP multicast.
- You can add and remove branch LSPs from a main point-to-multipoint LSP without disrupting traffic. The unaffected parts of the point-to-multipoint LSP continue to function normally.
- You can configure a node to be both a transit and an outbound (egress) router for different branch LSPs of the same point-to-multipoint LSP.

- You can enable link protection on a point-to-multipoint LSP. Link protection can provide a bypass LSP for each of the branch LSPs that make up the point-to-multipoint LSP. If any primary paths fail, traffic can be quickly switched to the bypass.
- You can configure subpaths either statically or dynamically.
- You can enable graceful restart on point-to-multipoint LSPs.

Related Documentation

- MPLS Traffic Engineering and Signaling Protocols Overview on page 15
- Point-to-Multipoint LSP Configuration Overview on page 30

Point-to-Multipoint LSP Configuration Overview

Supported Platforms ACX Series, MX Series, PTX Series, SRX100, SRX110, SRX210, SRX220, SRX240, SRX650, T Series

To set up a point-to-multipoint LSP:

1. Configure the primary LSP from the ingress router and the branch LSPs that carry traffic to the egress routers.
2. Specify a pathname on the primary LSP and this same path name on each branch LSP.

NOTE: By default, the branch LSPs are dynamically signaled by means of Constrained Shortest Path First (CSPF) and require no configuration. You can alternatively configure the branch LSPs as static paths.

Related Documentation

- Understanding Point-to-Multipoint LSPs on page 29
- Junos OS MPLS Applications Library for Routing Devices

Example: Configuring a Collection of Paths to Create an RSVP-Signaled Point-to-Multipoint LSP

Supported Platforms ACX Series, M Series, MX Series, PTX Series, SRX Series, T Series

This example shows how to configure a collection of paths to create an RSVP-signaled point-to-multipoint label-switched path (LSP).

- Requirements on page 30
- Overview on page 31
- Configuration on page 31
- Verification on page 47

Requirements

In this example, no special configuration beyond device initialization is required.

Overview

In this example, multiple routing devices serve as the transit, branch, and leaf nodes of a single point-to-multipoint LSP. On the provider edge (PE), Device PE1 is the ingress node. The branches go from PE1 to PE2, PE1 to PE3, and PE1 to PE4. Static unicast routes on the ingress node (PE1) point to the egress nodes.

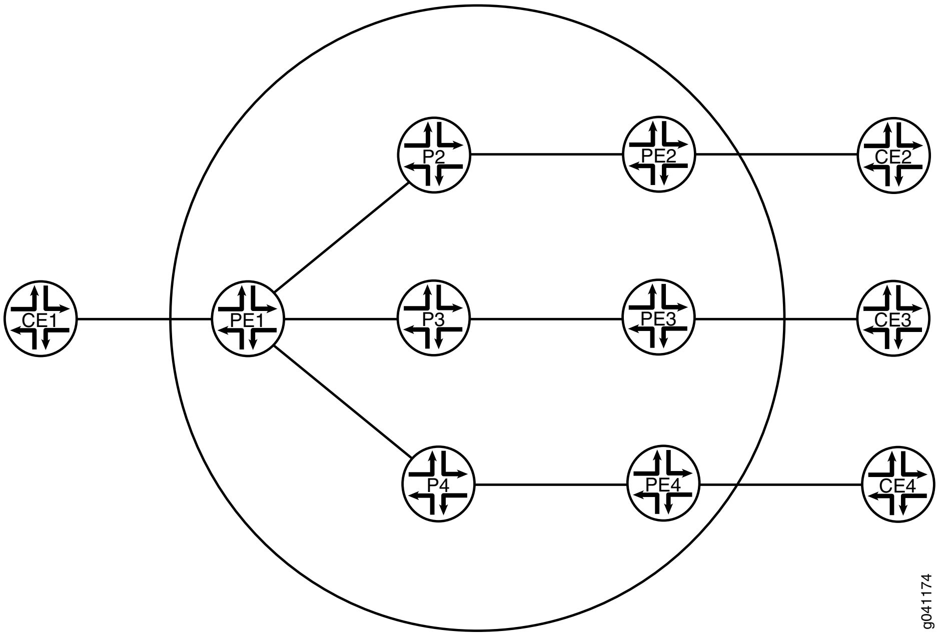
This example also demonstrates static routes with a next hop that is a point-to-multipoint LSP, using the **p2mp-lsp-next-hop** statement. This is useful when implementing filter-based forwarding.

NOTE: Another option is to use the **lsp-next-hop** statement to configure a regular point-to-point LSP to be the next hop. Though not shown in this example, you can optionally assign an independent preference and metric to the next hop.

Topology Diagram

Figure 6 on page 31 shows the topology used in this example.

Figure 6: RSVP-Signaled Point-to-Multipoint LSP



Configuration

CLI Quick Configuration

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

Device PE1

```
set interfaces ge-2/0/2 unit 0 description PE1-to-CE1
set interfaces ge-2/0/2 unit 0 family inet address 10.0.244.10/30
set interfaces fe-2/0/10 unit 1 description PE1-to-P2
set interfaces fe-2/0/10 unit 1 family inet address 2.2.2.1/24
set interfaces fe-2/0/10 unit 1 family mpls
set interfaces fe-2/0/9 unit 8 description PE1-to-P3
set interfaces fe-2/0/9 unit 8 family inet address 6.6.6.1/24
set interfaces fe-2/0/9 unit 8 family mpls
set interfaces fe-2/0/8 unit 9 description PE1-to-P4
set interfaces fe-2/0/8 unit 9 family inet address 3.3.3.1/24
set interfaces fe-2/0/8 unit 9 family mpls
set interfaces lo0 unit 1 family inet address 100.10.10.10/32
set protocols rsvp interface fe-2/0/10.1
set protocols rsvp interface fe-2/0/9.8
set protocols rsvp interface fe-2/0/8.9
set protocols rsvp interface lo0.1
set protocols mpls traffic-engineering bgp-igp
set protocols mpls label-switched-path PE1-PE2 to 100.50.50.50
set protocols mpls label-switched-path PE1-PE2 link-protection
set protocols mpls label-switched-path PE1-PE2 p2mp p2mp1
set protocols mpls label-switched-path PE1-PE3 to 100.70.70.70
set protocols mpls label-switched-path PE1-PE3 link-protection
set protocols mpls label-switched-path PE1-PE3 p2mp p2mp1
set protocols mpls label-switched-path PE1-PE4 to 100.40.40.40
set protocols mpls label-switched-path PE1-PE4 link-protection
set protocols mpls label-switched-path PE1-PE4 p2mp p2mp1
set protocols mpls interface fe-2/0/10.1
set protocols mpls interface fe-2/0/9.8
set protocols mpls interface fe-2/0/8.9
set protocols mpls interface lo0.1
set protocols ospf traffic-engineering
set protocols ospf area 0.0.0.0 interface ge-2/0/2.0
set protocols ospf area 0.0.0.0 interface fe-2/0/10.1
set protocols ospf area 0.0.0.0 interface fe-2/0/9.8
set protocols ospf area 0.0.0.0 interface fe-2/0/8.9
set protocols ospf area 0.0.0.0 interface lo0.1
set routing-options static route 5.5.5.0/24 p2mp-lsp-next-hop p2mp1
set routing-options static route 7.7.7.0/24 p2mp-lsp-next-hop p2mp1
set routing-options static route 4.4.4.0/24 p2mp-lsp-next-hop p2mp1
set routing-options router-id 100.10.10.10
```

Device CE1

```
set interfaces ge-1/3/2 unit 0 family inet address 10.0.244.9/30
set interfaces ge-1/3/2 unit 0 description CE1-to-PE1
set routing-options static route 10.0.104.8/30 next-hop 10.0.244.10
set routing-options static route 10.0.134.8/30 next-hop 10.0.244.10
set routing-options static route 10.0.224.8/30 next-hop 10.0.244.10
```

Device CE2

```
set interfaces ge-1/3/3 unit 0 family inet address 10.0.224.9/30
set interfaces ge-1/3/3 unit 0 description CE2-to-PE2
set routing-options static route 10.0.244.8/30 next-hop 10.0.224.10
```

Device CE3

```
set interfaces ge-2/0/1 unit 0 family inet address 10.0.134.9/30
set interfaces ge-2/0/1 unit 0 description CE3-to-PE3
set routing-options static route 10.0.244.8/30 next-hop 10.0.134.10
```

Device CE4

```
set interfaces ge-3/1/3 unit 0 family inet address 10.0.104.10/30
set interfaces ge-3/1/3 unit 0 description CE4-to-PE4
set routing-options static route 10.0.244.8/30 next-hop 10.0.104.9
```

Configuring the Ingress Label-Switched Router (LSR) (Device PE1)

Step-by-Step Procedure

To configure Device PE1:

1. Configure the interfaces, interface encapsulation, and protocol families.

```
[edit interfaces]
user@PE1# set ge-2/0/2 unit 0 description PE1-to-CE1
user@PE1# set ge-2/0/2 unit 0 family inet address 10.0.244.10/30
user@PE1# set fe-2/0/10 unit 1 description PE1-to-P2
user@PE1# set fe-2/0/10 unit 1 family inet address 2.2.2.1/24
user@PE1# set fe-2/0/10 unit 1 family mpls
user@PE1# set fe-2/0/9 unit 8 description PE1-to-P3
user@PE1# set fe-2/0/9 unit 8 family inet address 6.6.6.1/24
user@PE1# set fe-2/0/9 unit 8 family mpls
user@PE1# set fe-2/0/8 unit 9 description PE1-to-P4
user@PE1# set fe-2/0/8 unit 9 family inet address 3.3.3.1/24
user@PE1# set fe-2/0/8 unit 9 family mpls
user@PE1# set lo0 unit 1 family inet address 100.10.10.10/32
```

2. Enable RSVP, MPLS, and OSPF on the interfaces.

```
[edit protocols]
user@PE1# set rsvp interface fe-2/0/10.1
user@PE1# set rsvp interface fe-2/0/9.8
user@PE1# set rsvp interface fe-2/0/8.9
user@PE1# set rsvp interface lo0.1
user@PE1# set mpls interface fe-2/0/10.1
user@PE1# set mpls interface fe-2/0/9.8
user@PE1# set mpls interface fe-2/0/8.9
user@PE1# set mpls interface lo0.1
user@PE1# set ospf area 0.0.0.0 interface ge-2/0/2.0
user@PE1# set ospf area 0.0.0.0 interface fe-2/0/10.1
user@PE1# set ospf area 0.0.0.0 interface fe-2/0/9.8
user@PE1# set ospf area 0.0.0.0 interface fe-2/0/8.9
user@PE1# set ospf area 0.0.0.0 interface lo0.1
```

3. Configure the MPLS point-to-multipoint LSPs.

```
[edit protocols]
user@PE1# set mpls label-switched-path PE1-PE2 to 100.50.50.50
user@PE1# set mpls label-switched-path PE1-PE2 p2mp p2mp1
user@PE1# set mpls label-switched-path PE1-PE3 to 100.70.70.70
user@PE1# set mpls label-switched-path PE1-PE3 p2mp p2mp1
user@PE1# set mpls label-switched-path PE1-PE4 to 100.40.40.40
user@PE1# set mpls label-switched-path PE1-PE4 p2mp p2mp1
```

4. (Optional) Enable link protection on the LSPs.

   Link protection helps to ensure that traffic sent over a specific interface to a neighboring router can continue to reach the router if that interface fails.

```
[edit protocols]
user@PE1# set mpls label-switched-path PE1-PE2 link-protection
user@PE1# set mpls label-switched-path PE1-PE3 link-protection
```

```
user@PE1# set mpls label-switched-path PE1-PE4 link-protection
```

5. Enable MPLS to perform traffic engineering for OSPF.

   ```
   [edit protocols]
   user@PE1# set mpls traffic-engineering bgp-igp
   ```

   This causes the ingress routes to be installed in the inet.0 routing table. By default, MPLS performs traffic engineering for BGP only. You need to enable MPLS traffic engineering on the ingress LSR only.

6. Enable traffic engineering for OSPF.

   ```
   [edit protocols]
   user@PE1# set ospf traffic-engineering
   ```

   This causes the shortest-path first (SPF) algorithm to take into account the LSPs configured under MPLS.

7. Configure the router ID.

   ```
   [edit routing-options]
   user@PE1# set router-id 100.10.10.10
   ```

8. Configure static IP unicast routes with the point-to-multipoint LSP name as the next hop for each route.

   ```
   [edit routing-options]
   user@PE1# set static route 5.5.5.0/24p2mp-lsp-next-hop p2mp1
   user@PE1# set static route 7.7.7.0/24 p2mp-lsp-next-hop p2mp1
   user@PE1# set static route 4.4.4.0/24 p2mp-lsp-next-hop p2mp1
   ```

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

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

Configuring the Transit and Egress LSRs (Devices P2, P3, P4, PE2, PE3, and PE4)

Step-by-Step Procedure

To configure the transit and egress LSRs:

1. Configure the interfaces, interface encapsulation, and protocol families.

   ```
   [edit]
   user@P2# set interfaces fe-2/0/10 unit 2 description P2-to-PE1
   user@P2# set interfaces fe-2/0/10 unit 2 family inet address 2.2.2.2/24
   user@P2# set interfaces fe-2/0/10 unit 2 family mpls
   user@P2# set interfaces fe-2/0/9 unit 10 description P2-to-PE2
   user@P2# set interfaces fe-2/0/9 unit 10 family inet address 5.5.5.1/24
   user@P2# set interfaces fe-2/0/9 unit 10 family mpls
   user@P2# set interfaces lo0 unit 2 family inet address 100.20.20.20/32

   user@PE2# set interfaces ge-2/0/3 unit 0 description PE2-to-CE2
   user@PE2# set interfaces ge-2/0/3 unit 0 family inet address 10.0.224.10/30
   user@PE2# set interfaces fe-2/0/10 unit 5 description PE2-to-P2
   user@PE2# set interfaces fe-2/0/10 unit 5 family inet address 5.5.5.2/24
   user@PE2# set interfaces fe-2/0/10 unit 5 family mpls
   user@PE2# set interfaces lo0 unit 5 family inet address 100.50.50.50/32
   ```

```
user@P3# set interfaces fe-2/0/10 unit 6 description P3-to-PE1
user@P3# set interfaces fe-2/0/10 unit 6 family inet address 6.6.6.2/24
user@P3# set interfaces fe-2/0/10 unit 6 family mpls
user@P3# set interfaces fe-2/0/9 unit 11 description P3-to-PE3
user@P3# set interfaces fe-2/0/9 unit 11 family inet address 7.7.7.1/24
user@P3# set interfaces fe-2/0/9 unit 11 family mpls
user@P3# set interfaces lo0 unit 6 family inet address 100.60.60.60/32

user@PE3# set interfaces ge-2/0/1 unit 0 description PE3-to-CE3
user@PE3# set interfaces ge-2/0/1 unit 0 family inet address 10.0.134.10/30
user@PE3# set interfaces fe-2/0/10 unit 7 description PE3-to-P3
user@PE3# set interfaces fe-2/0/10 unit 7 family inet address 7.7.7.2/24
user@PE3# set interfaces fe-2/0/10 unit 7 family mpls
user@PE3# set interfaces lo0 unit 7 family inet address 100.70.70.70/32

user@P4# set interfaces fe-2/0/10 unit 3 description P4-to-PE1
user@P4# set interfaces fe-2/0/10 unit 3 family inet address 3.3.3.2/24
user@P4# set interfaces fe-2/0/10 unit 3 family mpls
user@P4# set interfaces fe-2/0/9 unit 12 description P4-to-PE4
user@P4# set interfaces fe-2/0/9 unit 12 family inet address 4.4.4.1/24
user@P4# set interfaces fe-2/0/9 unit 12 family mpls
user@P4# set interfaces lo0 unit 3 family inet address 100.30.30.30/32

user@PE4# set interfaces ge-2/0/0 unit 0 description PE4-to-CE4
user@PE4# set interfaces ge-2/0/0 unit 0 family inet address 10.0.104.9/30
user@PE4# set interfaces fe-2/0/10 unit 4 description PE4-to-P4
user@PE4# set interfaces fe-2/0/10 unit 4 family inet address 4.4.4.2/24
user@PE4# set interfaces fe-2/0/10 unit 4 family mpls
user@PE4# set interfaces lo0 unit 4 family inet address 100.40.40.40/32
```

2. Enable RSVP, MPLS, and OSPF on the interfaces.

```
[edit]
user@P2# set protocols rsvp interface fe-2/0/10.2
user@P2# set protocols rsvp interface fe-2/0/9.10
user@P2# set protocols rsvp interface lo0.2
user@P2# set protocols mpls interface fe-2/0/10.2
user@P2# set protocols mpls interface fe-2/0/9.10
user@P2# set protocols mpls interface lo0.2
user@P2# set protocols ospf area 0.0.0.0 interface fe-2/0/10.2
user@P2# set protocols ospf area 0.0.0.0 interface fe-2/0/9.10
user@P2# set protocols ospf area 0.0.0.0 interface lo0.2

user@PE2# set protocols rsvp interface fe-2/0/10.5
user@PE2# set protocols rsvp interface lo0.5
user@PE2# set protocols mpls interface fe-2/0/10.5
user@PE2# set protocols mpls interface lo0.5
user@PE2# set protocols ospf area 0.0.0.0 interface ge-2/0/3.0
user@PE2# set protocols ospf area 0.0.0.0 interface fe-2/0/10.5
user@PE2# set protocols ospf area 0.0.0.0 interface lo0.5

user@P3# set protocols rsvp interface fe-2/0/10.6
user@P3# set protocols rsvp interface fe-2/0/9.11
user@P3# set protocols rsvp interface lo0.6
```

```
user@P3# set protocols mpls interface fe-2/0/10.6
user@P3# set protocols mpls interface fe-2/0/9.11
user@P3# set protocols mpls interface lo0.6
user@P3# set protocols ospf area 0.0.0.0 interface fe-2/0/10.6
user@P3# set protocols ospf area 0.0.0.0 interface fe-2/0/9.11
user@P3# set protocols ospf area 0.0.0.0 interface lo0.6

user@PE3# set protocols rsvp interface fe-2/0/10.7
user@PE3# set protocols rsvp interface lo0.7
user@PE3# set protocols mpls interface fe-2/0/10.7
user@PE3# set protocols mpls interface lo0.7
user@PE3# set protocols ospf area 0.0.0.0 interface ge-2/0/1.0
user@PE3# set protocols ospf area 0.0.0.0 interface fe-2/0/10.7
user@PE3# set protocols ospf area 0.0.0.0 interface lo0.7

user@P4# set protocols rsvp interface fe-2/0/10.3
user@P4# set protocols rsvp interface fe-2/0/9.12
user@P4# set protocols rsvp interface lo0.3
user@P4# set protocols mpls interface fe-2/0/10.3
user@P4# set protocols mpls interface fe-2/0/9.12
user@P4# set protocols mpls interface lo0.3
user@P4# set protocols ospf area 0.0.0.0 interface fe-2/0/10.3
user@P4# set protocols ospf area 0.0.0.0 interface fe-2/0/9.12
user@P4# set protocols ospf area 0.0.0.0 interface lo0.3

user@PE4# set protocols rsvp interface fe-2/0/10.4
user@PE4# set protocols rsvp interface lo0.4
user@PE4# set protocols mpls interface fe-2/0/10.4
user@PE4# set protocols mpls interface lo0.4
user@PE4# set protocols ospf area 0.0.0.0 interface ge-2/0/0.0
user@PE4# set protocols ospf area 0.0.0.0 interface fe-2/0/10.4
user@PE4# set protocols ospf area 0.0.0.0 interface lo0.4
```

3. Enable traffic engineering for OSPF.

```
[edit]
user@P2# set protocols ospf traffic-engineering

user@P3# set protocols ospf traffic-engineering

user@P4# set protocols ospf traffic-engineering

user@PE2# set protocols ospf traffic-engineering

user@PE3# set protocols ospf traffic-engineering

user@PE4# set protocols ospf traffic-engineering
```

   This causes the shortest-path first (SPF) algorithm to take into account the LSPs configured under MPLS.

4. Configure the router IDs.

```
[edit]
```

```
user@P2# set routing-options router-id 100.20.20.20
```

```
user@P3# set routing-options router-id 100.60.60.60
```

```
user@P4# set routing-options router-id 100.30.30.30
```

```
user@PE2# set routing-options router-id 100.50.50.50
```

```
user@PE3# set routing-options router-id 100.70.70.70
```

```
user@PE4# set routing-options router-id 100.40.40.40
```

5. If you are done configuring the devices, commit the configuration.

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

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

Device PE1

```
user@PE1# show interfaces
ge-2/0/2 {
  unit 0 {
    description R1-to-CE1;
    family inet {
      address 10.0.244.10/30;
    }
  }
}
fe-2/0/10 {
  unit 1 {
    description PE1-to-P2;
    family inet {
      address 2.2.2.1/24;
    }
    family mpls;
  }
}
fe-2/0/9 {
  unit 8 {
    description PE1-to-P2;
    family inet {
      address 6.6.6.1/24;
    }
    family mpls;
  }
}
fe-2/0/8 {
  unit 9 {
    description PE1-to-P3;
    family inet {
      address 3.3.3.1/24;
    }
```

```
        family mpls;
    }
}
lo0 {
    unit 1 {
        family inet {
            address 100.10.10.10/32;
        }
    }
}
```

```
user@PE1# show protocols
rsvp {
    interface fe-2/0/10.1;
    interface fe-2/0/9.8;
    interface fe-2/0/8.9;
    interface lo0.1;
}
mpls {
    traffic-engineering bgp-igp;
    label-switched-path PE1-to-PE2 {
        to 100.50.50.50;
        link-protection;
        p2mp p2mp1;
    }
    label-switched-path PE1-to-PE3 {
        to 100.70.70.70;
        link-protection;
        p2mp p2mp1;
    }
    label-switched-path PE1-to-PE4 {
        to 100.40.40.40;
        link-protection;
        p2mp p2mp1;
    }
    interface fe-2/0/10.1;
    interface fe-2/0/9.8;
    interface fe-2/0/8.9;
    interface lo0.1;
}
ospf {
    traffic-engineering;
    area 0.0.0.0 {
        interface ge-2/0/2.0;
        interface fe-2/0/10.1;
        interface fe-2/0/9.8;
        interface fe-2/0/8.9;
        interface lo0.1;
    }
}
```

```
user@PE1# show routing-options
static {
    route 5.5.5.0/24 {
        p2mp-lsp-next-hop p2mp1;
    }
```

```
  route 7.7.7.0/24 {
    p2mp-lsp-next-hop p2mp1;
  }
  route 4.4.4.0/24 {
    p2mp-lsp-next-hop p2mp1;
  }
}
router-id 100.10.10.10;
```

Device P2

```
user@P2# show interfaces
fe-2/0/10 {
  unit 2 {
    description P2-to-PE1;
    family inet {
      address 2.2.2.2/24;
    }
    family mpls;
  }
fe-2/0/9 {
  unit 10 {
    description P2-to-PE2;
    family inet {
      address 5.5.5.1/24;
    }
    family mpls;
  }
}
lo0 {
  unit 2 {
    family inet {
      address 100.20.20.20/32;
    }
  }
}
```

```
user@P2# show protocols
rsvp {
  interface fe-2/0/10.2;
  interface fe-2/0/9.10;
  interface lo0.2;
}
mpls {
  interface fe-2/0/10.2;
  interface fe-2/0/9.10;
  interface lo0.2;
}
ospf {
  traffic-engineering;
  area 0.0.0.0 {
    interface fe-2/0/10.2;
    interface fe-2/0/9.10;
    interface lo0.2;
  }
}
```

```
user@P2# show routing-options
```

```
router-id 100.20.20.20;
```

Device P3

```
user@P3# show interfaces
fe-2/0/10 {
   unit 6 {
      description P3-to-PE1;
      family inet {
         address 6.6.6.2/24;
      }
      family mpls;
   }
}
fe-2/0/9 {
   unit 11 {
      description P3-to-PE3;
      family inet {
         address 7.7.7.1/24;
      }
      family mpls;
   }
}
lo0 {
   unit 6 {
      family inet {
         address 100.60.60.60/32;
      }
   }
}

user@P3# show protocols
rsvp {
   interface fe-2/0/10.6;
   interface fe-2/0/9.11;
   interface lo0.6;
}
mpls {
   interface fe-2/0/10.6;
   interface fe-2/0/9.11;
   interface lo0.6;
}
ospf {
   traffic-engineering;
   area 0.0.0.0 {
      interface fe-2/0/10.6;
      interface fe-2/0/9.11;
      interface lo0.6;
   }
}

user@P2# show routing-options
router-id 100.60.60.60;
```

Device P4

```
user@P4# show interfaces
fe-2/0/10 {
   unit 3 {
      description P4-to-PE1;
      family inet {
            address 3.3.3.2/24;
        }
        family mpls;
    }
}
fe-2/0/9 {
    unit 12 {
        description P4-to-PE4;
        family inet {
            address 4.4.4.1/24;
        }
        family mpls;
    }
}
lo0 {
    unit 3 {
        family inet {
            address 100.30.30.30/32;
        }
    }
}

user@P4# show protocols
rsvp {
    interface fe-2/0/10.3;
    interface fe-2/0/9.12;
    interface lo0.3;
}
mpls {
    interface fe-2/0/10.3;
    interface fe-2/0/9.12;
    interface lo0.3;
}
ospf {
    traffic-engineering;
    area 0.0.0.0 {
        interface fe-2/0/10.3;
        interface fe-2/0/9.12;
        interface lo0.3;
    }
}

user@P3# show routing-options
router-id 100.30.30.30;
```

Device PE2

```
user@PE2# show interfaces
    ge-2/0/3 {
        unit 0 {
            description PE2-to-CE2;
            family inet {
                address 10.0.224.10/30;
            }
        }
    }
    fe-2/0/10 {
        unit 5 {
            description PE2-to-P2;
            family inet {
                address 5.5.5.2/24;
            }
            family mpls;
        }
    }
    lo0 {
        unit 5 {
            family inet {
                address 100.50.50.50/32;
            }
        }
    }
}

user@PE2# show protocols
rsvp {
    interface fe-2/0/10.5;
    interface lo0.5;
}
mpls {
    interface fe-2/0/10.5;
    interface lo0.5;
}
ospf {
    traffic-engineering;
    area 0.0.0.0 {
        interface ge-2/0/3.0;
        interface fe-2/0/10.5;
        interface lo0.5;
    }
}

user@PE2# show routing-options
router-id 100.50.50.50;
```

Device PE3

```
user@PE3# show interfaces
    ge-2/0/1 {
        unit 0 {
            description PE3-to-CE3;
            family inet {
                address 10.0.134.10/30;
            }
        }
    }
    fe-2/0/10 {
        unit 7 {
            description PE3-to-P3;
            family inet {
                address 7.7.7.2/24;
            }
            family mpls;
        }
    }
    lo0 {
      unit 7 {
         family inet {
            address 100.70.70.70/32;
         }
      }
   }
}
```

```
user@PE3# show protocols
rsvp {
   interface fe-2/0/10.7;
   interface lo0.7;
}
mpls {
   interface fe-2/0/10.7;
   interface lo0.7;
}
ospf {
   traffic-engineering;
   area 0.0.0.0 {
      interface ge-2/0/1.0;
      interface fe-2/0/10.7;
      interface lo0.7;
   }
}
```

```
user@PE3# show routing-options
router-id 100.70.70.70;
```

Device PE4

```
user@PE4# show interfaces
   ge-2/0/0 {
      unit 0 {
         description PE4-to-CE4;
         family inet {
            address 10.0.104.9/30;
         }
      }
   }
   fe-2/0/10 {
      unit 4 {
         description PE4-to-P4;
         family inet {
            address 4.4.4.2/24;
         }
         family mpls;
      }
   }
   lo0 {
      unit 4 {
         family inet {
            address 100.40.40.40/32;
         }
      }
   }
}
```

```
user@PE4# show protocols
rsvp {
   interface fe-2/0/10.4;
   interface lo0.4;
}
mpls {
   interface fe-2/0/10.4;
   interface lo0.4;
}
ospf {
   traffic-engineering;
   area 0.0.0.0 {
      interface ge-2/0/0.0;
      interface fe-2/0/10.4;
      interface lo0.4;
   }
}
```

```
user@PE4# show routing-options
router-id 100.40.40.40;
```

Configuring Device CE1

Step-by-Step Procedure

To configure Device CE1:

1. Configure an interface to Device PE1.

   ```
   [edit interfaces]
   user@CE1# set ge-1/3/2 unit 0 family inet address 10.0.244.9/30
   user@CE1# set ge-1/3/2 unit 0 description CE1-to-PE1
   ```

2. Configure static routes from Device CE1 to the three other customer networks, with Device PE1 as the next hop.

   ```
   [edit routing-options]
   user@CE1# set static route 10.0.104.8/30 next-hop 10.0.244.10
   user@CE1# set static route 10.0.134.8/30 next-hop 10.0.244.10
   user@CE1# set static route 10.0.224.8/30 next-hop 10.0.244.10
   ```

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

   ```
   [edit]
   user@CE1# commit
   ```

Results

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

```
user@CE1# show interfaces
ge-1/3/2 {
   unit 0 {
      family inet {
         address 10.0.244.9/30;
         description CE1-to-PE1;
      }
   }
}
```

```
user@CE1# show routing-options
static {
   route 10.0.104.8/30 next-hop 10.0.244.10;
   route 10.0.134.8/30 next-hop 10.0.244.10;
   route 10.0.224.8/30 next-hop 10.0.244.10;
}
```

Configuring Device CE2

Step-by-Step Procedure

To configure Device CE2:

1. Configure an interface to Device PE2.

   ```
   [edit interfaces]
   user@CE2# set ge-1/3/3 unit 0 family inet address 10.0.224.9/30
   user@CE2# set ge-1/3/3 unit 0 description CE2-to-PE2
   ```

2. Configure a static route from Device CE2 to CE1, with Device PE2 as the next hop.

   ```
   [edit routing-options]
   user@CE2# set static route 10.0.244.8/30 next-hop 10.0.224.10
   ```

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

   ```
   [edit]
   user@CE2# commit
   ```

Results

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

```
user@CE2# show interfaces
ge-1/3/3 {
   unit 0 {
      family inet {
         address 10.0.224.9/30;
         description CE2-to-PE2;
      }
   }
}
```

```
user@CE2# show routing-options
static {
   route 10.0.244.8/30 next-hop 10.0.224.10;
}
```

Configuring Device CE3

Step-by-Step Procedure

To configure Device CE3:

1. Configure an interface to Device PE3.

   ```
   [edit interfaces]
   user@CE3# set ge-2/0/1 unit 0 family inet address 10.0.134.9/30
   user@CE3# set ge-2/0/1 unit 0 description CE3-to-PE3
   ```

2. Configure a static route from Device CE3 to CE1, with Device PE3 as the next hop.

   ```
   [edit routing-options]
   user@CE3# set static route 10.0.244.8/30 next-hop 10.0.134.10
   ```

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

   ```
   [edit]
   user@CE3# commit
   ```

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

```
user@CE3# show interfaces
ge-2/0/1 {
   unit 0 {
      family inet {
         address 10.0.134.9/30;
         description CE3-to-PE3;
      }
   }
}
```

```
user@CE3# show routing-options
static {
   route 10.0.244.8/30 next-hop 10.0.134.10;
}
```

Configuring Device CE4

Step-by-Step Procedure To configure Device CE4:

1. Configure an interface to Device PE4.

   ```
   [edit interfaces]
   user@CE4# set ge-3/1/3 unit 0 family inet address 10.0.104.10/30
   user@CE4# set ge-3/1/3 unit 0 description CE4-to-PE4
   ```

2. Configure a static route from Device CE4 to CE1, with Device PE4 as the next hop.

   ```
   [edit routing-options]
   user@CE4# set static route 10.0.244.8/30 next-hop 10.0.104.9
   ```

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

   ```
   [edit]
   user@CE4# commit
   ```

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

```
user@CE4# show interfaces
ge-3/1/3 {
   unit 0 {
```

```
            family inet {
               address 10.0.104.10/30;
               description CE4-to-PE4;
            }
         }
      }
```

```
user@CE4# show routing-options
static {
   route 10.0.244.8/30 next-hop 10.0.104.9;
}
```

Verification

Confirm that the configuration is working properly.

- Verifying Connectivity on page 47
- Verifying the State of the Point-to-Multipoint LSP on page 48
- Checking the Forwarding Table on page 49

Verifying Connectivity

Purpose Make sure that the devices can ping each other.

Action Run the **ping** command from CE1 to the interface on CE2 connecting to PE2.

```
user@CE1> ping 10.0.224.9
PING 10.0.224.9 (10.0.224.9): 56 data bytes
64 bytes from 10.0.224.9: icmp_seq=0 ttl=61 time=1.387 ms
64 bytes from 10.0.224.9: icmp_seq=1 ttl=61 time=1.394 ms
64 bytes from 10.0.224.9: icmp_seq=2 ttl=61 time=1.506 ms
^C
--- 10.0.224.9 ping statistics ---
3 packets transmitted, 3 packets received, 0% packet loss
round-trip min/avg/max/stddev = 1.387/1.429/1.506/0.055 ms
```

Run the **ping** command from CE1 to the interface on CE3 connecting to PE3.

```
user@CE1> ping 10.0.134.9
PING 10.0.134.9 (10.0.134.9): 56 data bytes
64 bytes from 10.0.134.9: icmp_seq=0 ttl=61 time=1.068 ms
64 bytes from 10.0.134.9: icmp_seq=1 ttl=61 time=1.062 ms
64 bytes from 10.0.134.9: icmp_seq=2 ttl=61 time=1.053 ms
^C
--- 10.0.134.9 ping statistics ---
3 packets transmitted, 3 packets received, 0% packet loss
round-trip min/avg/max/stddev = 1.053/1.061/1.068/0.006 ms
```

Run the **ping** command from CE1 to the interface on CE4 connecting to PE4.

```
user@CE1> ping 10.0.104.10
PING 10.0.104.10 (10.0.104.10): 56 data bytes
64 bytes from 10.0.104.10: icmp_seq=0 ttl=61 time=1.079 ms
64 bytes from 10.0.104.10: icmp_seq=1 ttl=61 time=1.048 ms
64 bytes from 10.0.104.10: icmp_seq=2 ttl=61 time=1.070 ms
^C
--- 10.0.104.10 ping statistics ---
3 packets transmitted, 3 packets received, 0% packet loss
round-trip min/avg/max/stddev = 1.048/1.066/1.079/0.013 ms
```

Verifying the State of the Point-to-Multipoint LSP

Purpose Make sure that the ingress, transit, and egress LSRs are in the Up state.

Action Run the **show mpls lsp p2mp** command on all of the LSRs. Only the ingress LSR is shown here.

```
user@PE1> show mpls lsp p2mp
Ingress LSP: 1 sessions
P2MP name: p2mp1, P2MP branch count: 3
To              From            State Rt P     ActivePath       LSPname
100.40.40.40    100.10.10.10    Up     0 *                      PE1-PE4
100.70.70.70    100.10.10.10    Up     0 *                      PE1-PE3
100.50.50.50    100.10.10.10    Up     0 *                      PE1-PE2
Total 3 displayed, Up 3, Down 0
...
```

Checking the Forwarding Table

Purpose Make sure that the routes are set up as expected by running the **show route forwarding-table** command. Only the routes to the remote customer networks are shown here.

Action

```
user@PE1> show route forwarding-table
Routing table: default.inet
Internet:
Destination        Type RtRef Next hop           Type Index NhRef Netif
...
10.0.104.8/30      user     0 3.3.3.2            ucst  1006     6 fe-2/0/8.9
10.0.134.8/30      user     0 6.6.6.2            ucst  1010     6 fe-2/0/9.8
10.0.224.8/30      user     0 2.2.2.2            ucst  1008     6 fe-2/0/10.1
...
```

Related Documentation

- *MPLS Applications Feature Guide for Routing Devices*

PART 3

Configuring MPLS VPNs

CHAPTER 3

Introduction to MPLS VPNs

MPLS VPN Overview

Supported Platforms SRX1500, SRX300, SRX320, SRX340, SRX345, SRX550M, vSRX

Virtual private networks (VPNs) are private networks that use a public network to connect two or more remote sites. Instead of dedicated connections between networks, VPNs use virtual connections routed (tunneled) through public networks that are typically service provider networks. VPNs are a cost-effective alternative to expensive dedicated lines. The type of VPN is determined by the connections it uses and whether the customer network or the provider network performs the virtual tunneling.

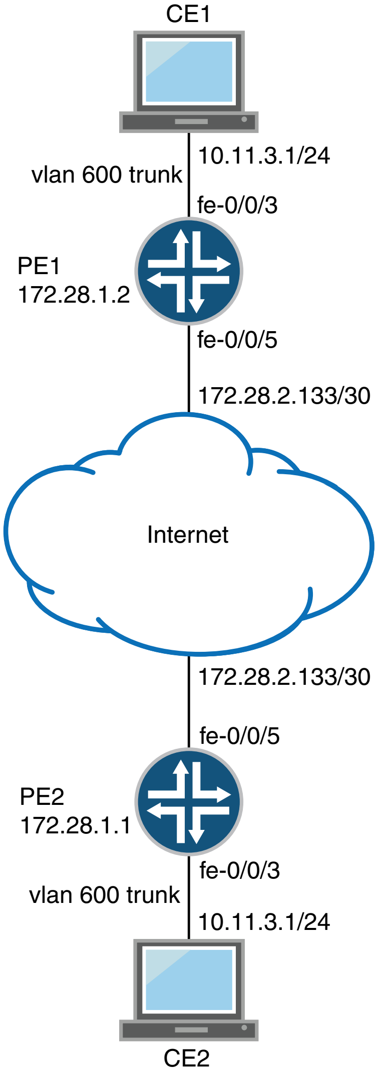
You can configure a router running Junos OS to participate in several types of VPNs. This topic discusses MPLS VPNs.

This topic contains the following sections:

MPLS VPN Topology

There are many ways to set up an MPLS VPN and direct traffic through it. Figure 7 on page 54 shows a typical MPLS VPN topology.

Figure 7: Typical VPN Topology



There are three primary types of MPLS VPNs: Layer 2 VPNs, Layer 2 circuits, and Layer 3 VPNs. All types of MPLS VPNs share certain components:

- The provider edge (PE) routers in the provider's network connect to the customer edge (CE) routers located at customer sites. PE routers support VPN and MPLS label functionality. Within a single VPN, pairs of PE routers are connected through a virtual tunnel, typically a label-switched path (LSP).
- Provider routers within the core of the provider's network are not connected to any routers at a customer site but are part of the tunnel between pairs of PE routers. Provider routers support LSP functionality as part of the tunnel support, but do not support VPN functionality.
- CE routers are the routers or switches located at the customer site that connect to the provider's network. CE routers are typically IP routers, but they can also be Asynchronous Transfer Mode (ATM), Frame Relay, or Ethernet switches.

All VPN functions are performed by the PE routers. Neither CE routers nor provider routers are required to perform any VPN functions.

MPLS VPN Routing

VPNs tunnel traffic as follows from one customer site to another customer site, using a public network as a transit network, when certain requirements are met:

1. Traffic is forwarded by standard IP forwarding from the CE routers to the PE routers.
2. The PE routers establish an LSP through the provider network.
3. The inbound PE router receives traffic, and it performs a route lookup. The lookup yields an LSP next hop, and the traffic is forwarded along the LSP.
4. The traffic reaches the outbound PE router, and the PE router pops the MPLS label and forwards the traffic with standard IP routing.

VRF Instances

A routing instance is a collection of routing tables, interfaces, and routing protocol parameters. The interfaces belong to the routing tables, and the routing protocol parameters control the information in the routing tables. In the case of MPLS VPNs, each VPN has a VPN routing and forwarding (VRF) instance.

A VRF instance consists of one or more routing tables, a derived forwarding table, the interfaces that use the forwarding table, and the policies and routing protocols that determine what goes into the forwarding table. Because each instance is configured for a particular VPN, each VPN has separate tables, rules, and policies that control its operation.

A separate VRF table is created for each VPN that has a connection to a CE router. The VRF table is populated with routes received from directly connected CE sites associated with the VRF instance, and with routes received from other PE routers in the same VPN.

Route Distinguishers

Because a typical transit network is configured to handle more than one VPN, the provider routers are likely to have multiple VRF instances configured. As a result, depending on the origin of the traffic and any filtering rules applied to the traffic, the BGP routing tables can contain multiple routes for a particular destination address. Because BGP requires that exactly one BGP route per destination be imported into the forwarding table, BGP must have a way to distinguish between potentially identical network layer reachability information (NLRI) messages received from different VPNs.

A route distinguisher is a locally unique number that identifies all route information for a particular VPN. Unique numeric identifiers allow BGP to distinguish between routes that are otherwise identical.

Each routing instance that you configure on a PE router must have a unique route distinguisher. There are two possible formats:

- ***as-number*:*number***, where ***as-number*** is an autonomous system (AS) number (a 2–byte value) in the range 1 through 65,535, and ***number*** is any 4–byte value. We recommend that you use an Internet Assigned Numbers Authority (IANA)-assigned, nonprivate AS number, preferably the ISP or the customer AS number.
- ***ip-address*:*number***, where ***ip-address*** is an IP address (a 4–byte value) and ***number*** is any 2–byte value. The IP address can be any globally unique unicast address. We recommend that you use the address that you configure in the **router-id** statement, which is a public IP address in your assigned prefix range.

The route target defines which route is part of a VPN. A unique route target helps distinguish between different VPN services on the same router. Each VPN also has a policy that defines how routes are imported into the VRF table on the router. A Layer 2 VPN is configured with import and export policies. A Layer 3 VPN uses a unique route target to distinguish between VPN routes.

The PE router then exports the route in IBGP sessions to the other provider routers. Route export is governed by any routing policy that has been applied to the particular VRF table. To propagate the routes through the provider network, the PE router must also convert the route to VPN format, which includes the route distinguisher.

When the outbound PE router receives the route, it strips off the route distinguisher and advertises the route to the connected CE router, typically through standard BGP IPv4 route advertisements.

Related Documentation

- Understanding MPLS Layer 2 VPNs on page 61
- Understanding MPLS Layer 3 VPNs on page 71
- Understanding MPLS Layer 2 Circuits on page 67

Configuring a BGP Session for MPLS VPNs (CLI Procedure)

Supported Platforms SRX1500, SRX300, SRX320, SRX340, SRX345, SRX550M, vSRX

NOTE: **This section is valid for Layer 2 VPNs and Layer 3 VPNs, but not Layer 2 circuits.**

To configure an IBGP session, perform the following steps on each PE router:

1. Configure BGP.

```
[edit]
user@host# edit protocols bgp group group-name
```

2. Set the BGP type to internal.

```
[edit protocols bgp group group-name]
user@host# set type internal
```

3. Specify the loopback interface.

```
[edit protocols bgp group group-name]
```

```
user@host# set local-address loopback-interface-ip-address
```

4. Set the Layer 2 or Layer 3 VPN family type to unicast.

   ```
   [edit protocols bgp group group-name]
   user@host# set family family-type unicast
   ```

   Replace ***family-type*** with **l2vpn** for a Layer 2 VPN or **inet–vpn** for a Layer 3 VPN.

5. Enter the loopback address of the neighboring PE router.

   ```
   [edit protocols bgp]
   user@host# set neighbor ip-address
   ```

6. Commit the configuration if you are finished configuring the device.

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

Related Documentation

- MPLS Layer 2 VPN Configuration Overview on page 61
- MPLS Layer 3 VPN Configuration Overview on page 72

Configuring an IGP and the RSVP Signaling Protocol (CLI Procedure)

Supported Platforms SRX1500, SRX300, SRX320, SRX340, SRX345, SRX550M, vSRX

To configure RSVP and OSPF:

1. Configure OSPF with traffic engineering support on the PE routers.

   ```
   [edit]
   user@host# edit protocols ospf traffic-engineering shortcuts
   ```

   NOTE: **You must configure the IGP at the [edit protocols] level, not within the routing instance at the [edit routing-instances] level.**

2. Enable RSVP on interfaces that participate in the LSP. For PE routers, enable interfaces on the source and destination points. For provider routers, enable interfaces that connect the LSP between the PE routers.

   ```
   [edit]
   user@host# edit protocols rsvp interface interface-name
   ```

3. Commit the configuration if you are finished configuring the device.

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

Related Documentation

- MPLS Layer 2 VPN Configuration Overview on page 61
- MPLS Layer 3 VPN Configuration Overview on page 72
- MPLS Layer 2 Circuit Configuration Overview on page 67

Configuring Routing Options for MPLS VPNs (CLI Procedure)

Supported Platforms SRX1500, SRX300, SRX320, SRX340, SRX345, SRX550M, vSRX

To configure routing options for a VPN:

1. Configure the AS number.

   ```
   [edit]
   user@host# set routing-options autonomous-system as-number
   ```

2. Commit the configuration if you are finished configuring the device.

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

Related Documentation

- MPLS Layer 2 VPN Configuration Overview on page 61
- MPLS Layer 3 VPN Configuration Overview on page 72
- MPLS Layer 2 Circuit Configuration Overview on page 67

Configuring a Routing Instance for MPLS VPNs (CLI Procedure)

Supported Platforms SRX1500, SRX300, SRX320, SRX340, SRX345, SRX550M, vSRX

To configure a VPN routing instance on each PE router:

1. Create the routing instance.

   ```
   [edit]
   user@host# edit routing-instances routing-instance-name
   ```

2. Create a routing instance description. (This text appears in the output of the **show route instance detail** command.)

   ```
   [edit routing-instances routing-instance-name]
   user@host# set description “text”
   ```

3. Specify the instance type, either **l2vpn** for Layer 2 VPNs or **vrf** for Layer 3 VPNs.

   ```
   [edit routing-instances routing-instance-name]
   user@host# set instance-type instance-type
   ```

4. Specify the interface of the remote PE router.

   ```
   [edit routing-instances routing-instance-name]
   user@host# set interface interface-name
   ```

5. Specify the route distinguisher using one of the following commands:

   ```
   [edit routing-instances routing-instance-name]
   user@host# set route-distinguisheras-number:number
   user@host# set route-distinguisher ip-address:number
   ```

6. Specify the policy for the Layer 2 VRF table.

   ```
   [edit routing-instances routing-instance-name]
   user@host# set vrf-import import-policy-name vrf-export export-policy-name
   ```

7. Specify the policy for the Layer 3 VRF table.

   ```
   [edit routing-instances routing-instance-name]
   user@host# set vrf-target target:community-id
   ```

   Where *community-id* is either *as-number*:*number* or *ip-address*:*number*.

8. Commit the configuration if you are finished configuring the device.

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

Related Documentation

- MPLS Layer 2 VPN Configuration Overview on page 61
- MPLS Layer 3 VPN Configuration Overview on page 72

CHAPTER 4

Configuring MPLS Layer 2 VPNs

Understanding MPLS Layer 2 VPNs

Supported Platforms SRX1500, SRX300, SRX320, SRX340, SRX345, SRX550M, vSRX

In an MPLS Layer 2 VPN, traffic is forwarded to the provider edge (PE) router in Layer 2 format, carried by MPLS through an label-switched path (LSP) over the service provider network, and then converted back to Layer 2 format at the receiving customer edge (CE) router.

Routing occurs on the customer routers, typically on the CE router. The CE router connected to a service provider on a Layer 2 VPN must select the appropriate circuit on which to send traffic. The PE router receiving the traffic sends it across the network to the PE router on the outbound side. The PE routers need no information about the customer's routes or routing topology, and need only to determine the virtual tunnel through which to send the traffic.

Implementing a Layer 2 VPN on the router is similar to implementing a VPN using a Layer 2 technology such as Asynchronous Transfer Mode (ATM) or Frame Relay.

Related Documentation

- MPLS VPN Overview on page 53
- MPLS Layer 2 VPN Configuration Overview on page 61

MPLS Layer 2 VPN Configuration Overview

Supported Platforms SRX1500, SRX300, SRX320, SRX340, SRX345, SRX550M, vSRX

To configure MPLS Layer 2 VPN functionality on a router running Junos OS, you must enable support on the provider edge (PE) router and configure the PE router to distribute routing information to other routers in the VPN, as explained in the following steps. However, because the tunnel information is maintained at both PE routers, neither the provider core routers nor the customer edge (CE) routers need to maintain any VPN information in their configuration databases.

To configure an MPLS Layer 2 VPN:

1. Determine all of the routers that you want to participate in the VPN, and then complete the initial configuration of their interfaces. See *Interfaces Feature Guide for Security Devices*.
2. For all of the routers in the VPN configuration, update the interface configurations to enable participation in the Layer 2 VPN. As part of the interface configuration, you must configure the MPLS address family for each interface that uses LDP or RSVP. See "Configuring Interfaces for Layer 2 VPNs (CLI Procedure)" on page 64.
3. For all of the routers in the VPN configuration, configure the appropriate protocols.
    a. MPLS—For PE routers and provider routers, use MPLS to advertise the Layer 2 VPN interfaces that communicate with other PE routers and provider routers.

    b. BGP and internal BGP (IBGP)—For PE routers, configure a BGP session to enable the routers to exchange information about routes originating and terminating in the VPN. (The PE routers use this information to determine which labels to use for traffic destined to the remote sites. The IBGP session for the VPN runs through the loopback address.) In addition, CE routers require a BGP connection to the PE routers. See "Configuring a BGP Session for MPLS VPNs (CLI Procedure)" on page 56.

    c. IGP and a signaling protocol—For PE routers, configure a signaling protocol (either LDP or RSVP) to dynamically set up label-switched paths (LSPs) through the provider network. (LDP routes traffic using IGP metrics. RSVP has traffic engineering that lets you override IGP metrics as needed.) You must use LDP or RSVP between PE routers and provider routers, but you cannot use them for interfaces between PE routers and CE routers.

       In addition, configure an IGP such as OSPF or static routes for PE routers to enable exchanges of routing information between the PE routers and provider routers. Each PE router's loopback address must appear as a separate route. Do not configure any summarization of the PE router's loopback addresses at the area boundary. Configure the provider network to run OSPF or IS-IS as an IGP, as well as IBGP sessions through either a full mesh or route reflector.

       See "Configuring an IGP and the LDP Signaling Protocol (CLI Procedure)" on page 70 and "Configuring an IGP and the RSVP Signaling Protocol (CLI Procedure)" on page 57.
4. For all of the routers in the VPN configuration, configure routing options. The only required routing option for VPNs is the AS number. You must specify it on each router involved in the VPN. See "Configuring Routing Options for MPLS VPNs (CLI Procedure)" on page 58.
5. For each PE router in the VPN configuration, configure a routing instance for each VPN. The routing instance should have the same name on each PE router. Each routing instance must have a unique route distinguisher associated with it. (VPN routing

instances need a route distinguisher to help BGP distinguish between potentially identical network layer reachable information [NLRI] messages received from different VPNs.) See “Configuring a Routing Instance for MPLS VPNs (CLI Procedure)” on page 58.

6. For each PE router in the VPN configuration, configure a VPN routing policy if you are not using a route target. Within the policy, describe which packets are sent and received across the VPN and specify how routes are imported into and exported from the router's VRF table. Each advertisement must have an associated route target that uniquely identifies the VPN for which the advertisement is valid. If the routing instance uses a policy for accepting and rejecting packets instead of a route target, you must specify the import and export routing policies and the community on each PE router. See “Configuring a Routing Policy for MPLS Layer 2 VPNs (CLI Procedure)” on page 63.

Related Documentation

- Verifying an MPLS Layer 2 VPN Configuration on page 65

Configuring a Routing Policy for MPLS Layer 2 VPNs (CLI Procedure)

Supported Platforms SRX1500, SRX300, SRX320, SRX340, SRX345, SRX550M, vSRX

These instructions show how to configure a Layer 2 VPN routing policy on the PE routers in the VPN.

After configuring an import routing policy for a Layer 2 VPN, configure an export routing policy for the Layer 2 VPN. Configure this export policy on the PE routers in the VPN. The export routing policy defines how routes are exported from the PE router routing table. An export policy is applied to routes sent to other PE routers in the VPN. The export policy must also evaluate all routes received over the routing protocol session with the CE router. The export policy must also contain a second term for rejecting all other routes.

To configure a Layer 2 VPN routing policy on a PE router:

1. Configure the import routing policy.

```
[edit]
user@host# edit policy-options policy-statement import-policy-name
```

2. Define the import policy’s term for accepting packets.

```
[edit edit policy-options policy-statement import-policy-name]
user@host# set term term-name-accept from protocol bgp community
   community-name
user@host# set term term-name-accept then accept
```

3. Define the import policy’s term for rejecting packets.

```
[edit edit policy-options policy-statement import-policy-name]
user@host# set term term-name-reject then reject
```

4. Configure the export routing policy.

```
[edit]
user@host# edit policy-options policy-statement export-policy-name
```

5. Define the export policy’s term for accepting packets.

```
[edit policy-options policy-statement export-policy-name]
user@host# set term term-name-accept from community add community-name
user@host# set termterm-name-accept then accept
```

6. Define the export policy's term for rejecting packets.

```
[edit policy-options policy-statement export-policy-name]
user@host# set term term-name-reject from community add community-name
user@host# set term term-name-reject then reject
```

7. Define the export policy's community using one of the following commands.

```
[edit policy-options policy-statement export-policy-name]
user@host# community community-name target: as-number
user@host# community community-name target: ip-address:number
```

8. Commit the configuration if you are finished configuring the device.

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

Related Documentation

- MPLS Layer 2 VPN Configuration Overview on page 61
- MPLS Layer 3 VPN Configuration Overview on page 72

Configuring Interfaces for Layer 2 VPNs (CLI Procedure)

Supported Platforms SRX1500, SRX300, SRX320, SRX340, SRX345, SRX550M, vSRX

Configuring the router interfaces that participate in the VPN is similar to configuring them for other uses, with a few requirements for the VPN. Perform the following tasks for each interface involved in the VPN, except Layer 3 loopback interfaces, which do not require other configuration.

To configure an interface for an MPLS VPN:

1. Configure IPv4 on all of the routers' interfaces.
    - For all interfaces except loopback interfaces and Layer 2 VPN interfaces facing a CE router:

      ```
      [edit]
      user@host# edit interfaces interface-name unit logical_interface family inet address ipv4_address
      ```

    - For a loopback address on a Layer 2 configuration:

      ```
      [edit]
      user@host# edit interfaces lo0 unit logical_interface family inet address ipv4_address primary
      ```

    - For a Layer 2 VPN interface facing a CE router:

      ```
      [edit]
      user@host# set interfaces interface-name vlan-tagging encapsulation vlan-ccc unit logical_interface encapsulation vlan-ccc vlan-id id-number
      ```

2. Configure the MPLS address family on the PE router or provider router interfaces that communicate with other PE routers or provider routers (and not loopback addresses).

    ```
    [edit interfaces interface]
    user@host# set unit logical_interface family mpls
    ```

3. Configure encapsulation for the interfaces on the PE routers that communicate with the CE routers in Layer 2 VPNs and Layer 2 circuits. If multiple logical units are configured, the encapsulation type is needed at the interface level only. It is always required at the unit level.

    ```
    [edit interfaces interface]
    user@host# set encapsulation encapsulation_type
    user@host# set unit logical_interface encapsulation encapsulation_type
    ```

4. Enable protocol mpls on CE facing interface.

    ```
    [edit interfaces interface]
    user@host# set protocols mpls interface interface-name
    ```

5. Commit the configuration if you are finished configuring the device.

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

Related Documentation

- MPLS Layer 2 VPN Configuration Overview on page 61
- MPLS Layer 3 VPN Configuration Overview on page 72
- MPLS Layer 2 Circuit Configuration Overview on page 67

Verifying an MPLS Layer 2 VPN Configuration

Supported Platforms SRX1500, SRX300, SRX320, SRX340, SRX345, SRX550M, vSRX

Purpose Verify the connectivity of MPLS Layer 2 VPNs using the **ping mpls** command. This command helps to verify that a VPN has been enabled by testing the integrity of the VPN connection between the PE routers. It does not test the connection between a PE router and CE router.

Action

- To ping an interface configured for the Layer 2 VPN on the PE router, use the following command:

  ping mpls l2vpn interface *interface-name*

- To ping a combination of the Layer 2 VPN routing instance name, the local site identifier, and the remote site identifier to test the integrity of the Layer 2 VPN connection (specified by identifiers) between the two PE routers, use the following command:

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

Related Documentation

- MPLS Layer 2 VPN Configuration Overview on page 61

CHAPTER 5

Configuring MPLS Layer 2 Circuit VPNs

Understanding MPLS Layer 2 Circuits

Supported Platforms SRX1500, SRX300, SRX320, SRX340, SRX345, SRX550M, vSRX

An MPLS Layer 2 circuit is a point-to-point Layer 2 connection that transports traffic by means of MPLS or another tunneling technology on the service provider network. The Layer 2 circuit creates a virtual connection to direct traffic between two customer edge (CE) routers across a service provider network. The main difference between a Layer 2 VPN and a Layer 2 circuit is the method of setting up the virtual connection. As with a leased line, a Layer 2 circuit forwards all packets received from the local interface to the remote interface.

Each Layer 2 circuit is represented by the logical interface connecting the local provider edge (PE) router to the local CE router. All Layer 2 circuits using a particular remote PE router neighbor is identified by its IP address and is usually the endpoint destination for the label-switched path (LSP) tunnel transporting the Layer 2 circuit.

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. Based on the virtual circuit ID and the neighbor relationship, an LDP label is bound to an LDP circuit. LDP uses the binding for sending traffic on that Layer 2 circuit to the remote CE router.

Related Documentation

- MPLS VPN Overview on page 53
- MPLS Layer 2 Circuit Configuration Overview on page 67

MPLS Layer 2 Circuit Configuration Overview

Supported Platforms SRX1500, SRX300, SRX320, SRX340, SRX345, SRX550M, vSRX

To configure an MPLS Layer 2 circuit:

1. Determine all of the routers that you want to participate in the circuit, and then complete the initial configuration of their interfaces. See the *Interfaces Feature Guide for Security Devices*.
2. For all of the routers in the circuit configuration, update the interface configurations to enable participation in the Layer 2 circuit.
   a. On the interface communicating with the other provider edge (PE) router, specify MPLS and IPv4, and include the IP address. For the loopback interface, specify **inet**, and include the IP address. For IPv4, designate the loopback interface as primary so it can receive control packets. (Because it is always operational, the loopback interface is best able to perform the control function.)

   b. On the PE router interface facing the customer edge (CE) router, specify a circuit cross-connect (CCC) encapsulation type. The type of encapsulation depends on the interface type. For example, an Ethernet interface uses **ethernet-ccc**. (The encapsulation type determines how the packet is constructed for that interface.)

   c. On the CE router interface that faces the PE router, specify **inet** (for IPv4), and include the IP address. In addition, specify a routing protocol such as Open Shortest Path First (OSPF), which specifies the area and IP address of the router interface.

   See "Configuring Interfaces for Layer 2 VPNs (CLI Procedure)" on page 64.
3. For all of the routers in the circuit configuration, configure the appropriate protocols.
   a. MPLS—For PE routers and provider routers, use MPLS to advertise the Layer 2 circuit interfaces that communicate with other PE routers and provider routers.

   b. BGP—For PE routers, configure a BGP session.

   c. IGP and a signaling protocol—For PE routers, configure a signaling protocol (either LDP or RSVP) to dynamically set up label-switched paths (LSPs) through the provider network. (LDP routes traffic using IGP metrics. RSVP has traffic engineering that lets you override IGP metrics as needed.) You must use LDP or RSVP between PE routers and provider routers, but cannot use them for interfaces between PE routers and CE routers.

      In addition, configure an IGP such as OSPF or static routes on the PE routers to enable exchanges of routing information between the PE routers and provider routers. Each PE router's loopback address must appear as a separate route. Do not configure any summarization of the PE router's loopback addresses at the area boundary. Configure the provider network to run OSPF or IS-IS as an IGP, as well as IBGP sessions through either a full mesh or route reflector.

      See "Configuring an IGP and the LDP Signaling Protocol (CLI Procedure)" on page 70 and "Configuring an IGP and the RSVP Signaling Protocol (CLI Procedure)" on page 57.
4. For all of the routers in the circuit configuration, configure routing options. The only required routing option for circuits is the autonomous system (AS) number. You must

specify it on each router involved in the circuit. See "Configuring Routing Options for MPLS VPNs (CLI Procedure)" on page 58.

5. For PE routers, configure Layer 2 circuits on the appropriate interfaces. See "Configuring an MPLS Layer 2 Circuit (CLI Procedure)" on page 69.

Related Documentation

- Verifying an MPLS Layer 2 Circuit Configuration on page 69

Configuring an MPLS Layer 2 Circuit (CLI Procedure)

Supported Platforms SRX1500, SRX300, SRX320, SRX340, SRX345, SRX550M, vSRX

To configure a Layer 2 circuit on a PE router:

1. Enable a Layer 2 circuit on the appropriate interface.

   ```
   [edit]
   user@host# edit protocols l2circuit neighbor interface-name interface interface-name
   ```

2. Enter the circuit ID number.

   ```
   [edit protocols l2circuit neighbor interface-name interface interface-name]
   user@host# set virtual-circuit-id id-number
   ```

   For **neighbor**, specify the local loopback address, and for **interface**, specify the interface name of the remote PE router.

3. Commit the configuration if you are finished configuring the device.

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

Related Documentation

- MPLS Layer 2 Circuit Configuration Overview on page 67

Verifying an MPLS Layer 2 Circuit Configuration

Supported Platforms SRX1500, SRX300, SRX320, SRX340, SRX345, SRX550M, vSRX

Purpose To verify the connectivity of MPLS Layer 2 circuits, use the **ping mpls** command. This command helps to verify that the circuit has been enabled by testing the integrity of the Layer 2 circuit between the source and destination routers.

Action

- To ping an interface configured for the Layer 2 circuit on the PE router, enter the following command:

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

- To ping a combination of the IPv4 prefix and the virtual circuit ID on the destination PE router, enter the following command:

  ```
  ping mpls l2circuit virtual-circuitprefix virtual-circuit-id
  ```

Related Documentation

- MPLS Layer 2 Circuit Configuration Overview on page 67

Configuring an IGP and the LDP Signaling Protocol (CLI Procedure)

Supported Platforms SRX1500, SRX300, SRX320, SRX340, SRX345, SRX550M, vSRX

The following instructions show how to configure LDP and OSPF on PE routers and provider routers. Within the task, you specify which interfaces to enable for LDP. Perform this step on each PE router interface and provider router interface that communicates with other PE routers and provider routers. For OSPF, you configure at least one area on at least one of the router's interfaces. (An AS can be divided into multiple areas.) These instructions use the backbone area **0.0.0.0** and show how to enable traffic engineering for Layer 2 VPN circuits.

To configure LDP and OSPF:

1. Enable the ldp protocol.

```
[edit]
user@host# edit protocols ldp
```

NOTE: You must configure the IGP at the **[protocols]** level of the configuration hierarchy, not within the routing instance at the **[routing-instances]** level of the configuration hierarchy.

2. Specify which interfaces to enable for LDP.

```
[edit protocols ldp]
user@host# edit interface interface-name
```

3. Configure OSPF for each interface that uses LDP.

```
[edit]
user@host# edit protocols ospf area 0.0.0.0 interface interface-name
```

4. (Layer 2 VPN circuits only) Enable traffic engineering.

```
[edit protocols ospf]
user@host# set traffic engineering
```

5. Commit the configuration if you are finished configuring the device.

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

Related Documentation

- MPLS Layer 2 VPN Configuration Overview on page 61
- MPLS Layer 3 VPN Configuration Overview on page 72
- MPLS Layer 2 Circuit Configuration Overview on page 67

CHAPTER 6

Configuring MPLS Layer 3 VPNs

Understanding MPLS Layer 3 VPNs

Supported Platforms SRX1500, SRX300, SRX320, SRX340, SRX345, SRX550M, vSRX

An MPLS Layer 3 VPN operates at the Layer 3 level of the OSI model, the Network layer. The VPN is composed of a set of sites that are connected over a service provider's existing public Internet backbone. The sites share common routing information and the connectivity of the sites is controlled by a collection of policies.

In an MPLS Layer 3 VPN, routing occurs on the service provider's routers. The provider routers route and forward VPN traffic at the entry and exit points of the transit network. The service provider network must learn the IP addresses of devices sending traffic across the VPN and the routes must be advertised and filtered throughout the provider network. As a result, Layer 3 VPNs require information about customer routes and a more extensive VPN routing and forwarding (VRF) policy configuration than a Layer 2 VPN. This information is used to share and filter routes that originate or terminate in the VPN.

The MPLS Layer 3 VPN requires more processing power on the provider edge (PE) routers than a Layer 2 VPN, because the Layer 3 VPN has larger routing tables for managing network traffic on the customer sites. Route advertisements originate at the customer edge (PE) routers and are shared with the inbound PE routers through standard IP routing protocols, typically BGP. Based on the source address, the PE router filters route advertisements and imports them into the appropriate VRF table.

The provider router uses OSPF and LDP to communicate with the PE routers. For OSPF, the provider router interfaces that communicate with the PE routers are specified, as well as the loopback interface. For the PE routers, the loopback interface is in passive mode, meaning it does not send OSPF packets to perform the control function.

Related Documentation

- MPLS VPN Overview on page 53

MPLS Layer 3 VPN Configuration Overview

Supported Platforms EX4600, QFX Series, SRX100, SRX110, SRX210, SRX220, SRX240, SRX650

To configure MPLS Layer 3 VPN functionality on a router running Junos OS, you must enable support on the provider edge (PE) router and configure the PE router to distribute routing information to other routers in the VPN, as explained in the following steps. However, because the tunnel information is maintained at both PE routers, neither the provider core routers nor the customer edge (CE) routers need to maintain any VPN information in their configuration databases.

To configure an MPLS Layer 3 VPN:

1. Determine all of the routers that you want to participate in the VPN, and then complete the initial configuration of their interfaces. See the *Junos OS Interfaces Configuration Guide for Security Devices*.
2. For all of the routers in the VPN configuration, update the interface configurations to enable participation in the Layer 3 VPN. As part of the interface configuration, you must configure the MPLS address family for each interface that uses LDP or RSVP. See "Configuring Interfaces for Layer 2 VPNs (CLI Procedure)" on page 64.
3. For all of the routers in the VPN configuration, configure the appropriate protocols.
   a. MPLS—If you are using RSVP, use MPLS to advertise the Layer 3 VPN interfaces on the PE routers and provider routers that communicate with other PE routers and provider routers. See Configuring MPLS for Layer 2 VPNs (CLI Procedure).

   b. BGP, EBGP, and internal BGP (IBGP)—For PE routers, configure a BGP session to enable the routers to exchange information about routes originating and terminating in the VPN. (The PE routers use this information to determine which labels to use for traffic destined to the remote sites. The IBGP session for the VPN runs through the loopback address.) In addition, CE routers require a BGP connection to the PE routers. See "Configuring a BGP Session for MPLS VPNs (CLI Procedure)" on page 56.

   c. IGP and a signaling protocol—For PE routers and provider, configure a signaling protocol (either LDP or RSVP) to dynamically set up label-switched paths (LSPs) through the provider network. (LDP routes traffic using IGP metrics. RSVP has traffic engineering that lets you override IGP metrics as needed.) You must use LDP or RSVP between PE routers and provider routers, but cannot use them for interfaces between PE routers and CE routers.

      In addition, configure an IGP such as OSPF or static routes on the PE routers in order to enable exchanges of routing information between the PE routers and provider routers. Each PE router's loopback address must appear as a separate route. Do not configure any summarization of the PE router's loopback addresses at the area boundary. Configure the provider network to run OSPF or IS-IS as an IGP, as well as IBGP sessions through either a full mesh or route reflector.

See “Configuring an IGP and the LDP Signaling Protocol (CLI Procedure)” on page 70 and “Configuring an IGP and the RSVP Signaling Protocol (CLI Procedure)” on page 57.

4. For all of the routers in the VPN configuration, configure routing options. The only required routing option for VPNs is the autonomous system (AS) number. You must specify it on each router involved in the VPN. See “Configuring Routing Options for MPLS VPNs (CLI Procedure)” on page 58.
5. For each PE router in the VPN configuration, configure a routing instance for each VPN. The routing instance should have the same name on each PE router. Each routing instance must have a unique route distinguisher associated with it. (VPN routing instances need a route distinguisher to help BGP distinguish between potentially identical network layer reachable information [NLRI] messages received from different VPNs.) See “Configuring a Routing Instance for MPLS VPNs (CLI Procedure)” on page 58.
6. For CE routers, configure a routing policy. In addition, if you are not using a route target, configure a VPN routing policy for each PE router in the VPN configuration. Within the policy, describe which packets are sent and received across the VPN and specify how routes are imported into and exported from the router's VRF table. Each advertisement must have an associated route target that uniquely identifies the VPN for which the advertisement is valid. See “Configuring a Routing Policy for MPLS Layer 3 VPNs (CLI Procedure)” on page 73.

Related Documentation

- Verifying an MPLS Layer 3 VPN Configuration on page 74

Configuring a Routing Policy for MPLS Layer 3 VPNs (CLI Procedure)

Supported Platforms SRX1500, SRX300, SRX320, SRX340, SRX345, SRX550M, vSRX

To configure a Layer 3 VPN routing policy on a CE router:

1. Configure the routing policy for the loopback interface.

```
[edit]
user@host# edit policy-options policy-statement policy-name
```

2. Define the term for accepting packets.

```
[edit policy-options policy-statement policy-name]
user@host# set term term-name-accept from protocol direct route-filter
   local-loopback-address/netmask exact
user@host# set termterm-name-accept then accept
```

3. Define the term for rejecting packets.

```
[edit policy-options policy-statement policy-name]
user@host# set termterm-name-reject then reject
```

4. Commit the configuration if you are finished configuring the device.

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

Related Documentation

- MPLS Layer 3 VPN Configuration Overview on page 72

Verifying an MPLS Layer 3 VPN Configuration

Supported Platforms SRX1500, SRX300, SRX320, SRX340, SRX345, SRX550M, vSRX

Purpose Verify the connectivity of MPLS Layer 3 VPNs using the **ping mpls** command. This command helps to verify that a VPN has been enabled by testing the integrity of the VPN connection between the source and destination routers. The destination prefix corresponds to a prefix in the Layer 3 VPN. However, ping tests only whether the prefix is present in a PE VRF table.

Action To a combination of a IPv4 destination prefix and a Layer 3 VPN name on the destination PE router, use the following command:

ping mpls l3vpn *l3vpn-name* prefix *prefix* count *count*

Related Documentation

- MPLS Layer 3 VPN Configuration Overview on page 72

PART 4

Configuring CLNS VPNs

CHAPTER 7

Introduction to CLNS

CLNS Overview

Supported Platforms SRX100, SRX110, SRX210, SRX220, SRX240, SRX550M, SRX650

Connectionless Network Service (CLNS) is a Layer 3 protocol similar to IP version 4 (IPv4) for linking hosts (end systems) with routers (intermediate systems) in an Open Systems Interconnection (OSI) network. CLNS and its related OSI protocols, Intermediate System-to-Intermediate System (IS-IS) and End System-to-Intermediate System (ES-IS), are International Organization for Standardization (ISO) standards.

You can configure devices running Junos OS as provider edge (PE) routers within a CLNS network. CLNS networks can be connected over an IP MPLS network core using Border Gateway Protocol (BGP) and MPLS Layer 3 virtual private networks (VPNs). See RFC 2547, *BGP/MPLS VPNs*.

CLNS uses network service access points (NSAPs), similar to IP addresses found in IPv4, to identify end systems (hosts) and intermediate systems (routers). ES-IS enables the hosts and routers to discover each other. IS-IS is the interior gateway protocol (IGP) that carries ISO CLNS routes through a network.

For more information about CLNS, see the ISO 8473 standards.

Related Documentation

- CLNS Configuration Overview on page 77
- Understanding ES-IS for CLNS on page 81
- Understanding IS-IS for CLNS on page 85
- Understanding Static Routes for CLNS on page 89
- Understanding BGP for CLNS VPNs on page 93

CLNS Configuration Overview

Supported Platforms SRX100, SRX110, SRX210, SRX220, SRX240, SRX550M, SRX650

To configure CLNS:

1. Configure the network interfaces. See the *Junos OS Interfaces Configuration Guide for Security Devices*.
2. If applicable, configure BGP and VPNs. See:
   - Example: Configuring BGP for CLNS VPNs on page 93
   - MPLS Layer 2 VPN Configuration Overview on page 61
   - MPLS Layer 3 VPN Configuration Overview on page 72
3. Configure a VPN routing instance. You typically configure ES-IS, IS-IS, and CLNS static routes using a VPN routing instance. See "Example: Configuring a VPN Routing Instance for CLNS" on page 95.
4. Configure one or more of the following protocols for CLNS (depending on your network).
   - ES-IS—If a device is a PE router within a CLNS island that contains any end systems, you must configure ES-IS on the device. If a CLNS island does not contain any end systems, you do not need to configure ES-IS on a device. See "Example: Configuring ES-IS for CLNS" on page 81.

     NOTE: ES-IS is enabled only if either ES-IS or IS-IS is configured on the router. ES-IS must not be disabled. If ES-IS is not explicitly configured, the interface sends and receives only intermediate system hello (ISH) messages. If ES-IS is explicitly configured and disabled, the interface does not send or receive ES-IS packets. If ES-IS is explicitly configured and not disabled, the interface sends and receives ISH messages as well as ES-IS packets.

     One of the interfaces that is configured for ES-IS must be configured with an ISO address for hello messages. The ISO address family must be configured on an interface to support ES-IS on that interface.
   - IS-IS—You can configure IS-IS to exchange CLNS routes within a CLNS island. See "Example: Configuring IS-IS for CLNS" on page 85.

     NOTE: If you have a pure CLNS island—an island that does not contain any IP devices—you must disable IPv4 and IPv6 routing. Also, to export BGP routes into IS-IS, you must configure and apply an export policy.
   - Static routes—If some devices in your network do not support IS-IS, you must configure CLNS static routes. You can use static routing with or without IS-IS. You might also consider using static routes if your network is simple. See "Example: Configuring Static Routes for CLNS" on page 89.
   - BGP—See "Example: Configuring BGP for CLNS VPNs" on page 93.

NOTE: Many of the configuration statements used to configure CLNS and routing protocols can be included at different hierarchy levels in the configuration.

Related Documentation

- CLNS Overview on page 77
- Verifying a CLNS VPN Configuration on page 97

CHAPTER 8

Configuring ES-IS for CLNS

Understanding ES-IS for CLNS

Supported Platforms SRX100, SRX110, SRX210, SRX220, SRX240, SRX550M, SRX650

End System-to-Intermediate System (ES-IS) is a protocol that resolves Layer 3 ISO network service access points (NSAP) to Layer 2 addresses. ES-IS has an equivalent role as Address Resolution Protocol (ARP) in IP version 4 (IPv4).

ES-IS provides the basic interaction between Connectionless Network Service (CLNS) hosts (end systems) and routers (intermediate systems). ES-IS allows hosts to advertise NSAP addresses to other routers and hosts attached to the network. Those routers can then advertise the address to the rest of the network by using Intermediate System-to-Intermediate System (IS-IS). Routers use ES-IS to advertise their network entity title (NET) to hosts and routers that are attached to that network.

ES-IS routes are exported to Layer 1 IS-IS by default. You can also export ES-IS routes into Layer 2 IS-IS by configuring a routing policy. ES-IS generates and receives end system hello (ESH) hello messages when the protocol is configured on an interface. ES-IS is a resolution protocol that allows a network to be fully ISO integrated at both the network layer and the data layer.

The resolution of Layer 3 ISO NSAPs to Layer 2 subnetwork point of attachments (SNPAs) by ES-IS is equivalent to ARP within an IPv4 network. If a device is a provider edge (PE) router within a CLNS island that contains any end systems, you must configure ES-IS on the device.

For more information about ES-IS, see the ISO 9542 standard.

Related Documentation

- CLNS Overview on page 77
- Example: Configuring ES-IS for CLNS on page 81

Example: Configuring ES-IS for CLNS

Supported Platforms SRX1500, SRX300, SRX320, SRX340, SRX345, SRX550M, vSRX

This example shows how to create a routing instance and enable ES-IS for CLNS on all interfaces.

- Requirements on page 82
- Overview on page 82
- Configuration on page 82
- Verification on page 83

Requirements

Before you begin, configure the network interfaces. See *Interfaces Feature Guide for Security Devices*.

Overview

The configuration instructions in this topic describe how to create a routing-instance called aaaa, set the end system configuration timer for the interfaces to 180, and set a preference value to 30 for ES-IS.

Configuration

CLI Quick Configuration

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

```
set routing-instances aaaa protocols esis interface all end-system-configuration-timer
  180
set routing-instances aaaa protocols esis preference 30
```

Step-by-Step Procedure

The following example requires you to navigate various levels in the configuration hierarchy. For instructions on how to do that, see *Using the CLI Editor in Configuration Mode*.

To configure ES-IS for CLNS:

1. Configure the routing instance.

```
[edit]
user@host# edit routing-instances aaaa
```

2. Enable ES-IS on all interfaces.

```
[edit routing-instances aaaa]
user@host# set protocols esis interface all
```

3. Configure the end system configuration timer.

```
[edit routing-instances aaaa]
user@host# set protocols esis interface all end-system-configuration-timer 180
```

4. Configure the preference value.

```
[edit routing-instances aaaa]
user@host# set protocols esis preference 30
```

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

```
[edit]
user@host# show routing-instances
aaaa {
   protocols {
      esis {
         preference 30;
         interface all {
            end-system-configuration-timer 180;
         }
      }
   }
}
```

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

Verification

Confirm that the configuration is working properly.

- Verifying Routing-Instance for CLNS on page 83
- Verifying ES-IS for CLNS on page 83

Verifying Routing-Instance for CLNS

Purpose Verify that the policy options are enabled for the routing instance.

Action From operational mode, enter the **show routing-instances** command.

Verifying ES-IS for CLNS

Purpose Verify that ES-IS is enabled.

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

Related Documentation

- CLNS Configuration Overview on page 77
- Understanding ES-IS for CLNS on page 81
- Verifying a CLNS VPN Configuration on page 97

CHAPTER 9

Configuring IS-IS for CLNS

Understanding IS-IS for CLNS

Supported Platforms EX4600, SRX100, SRX110, SRX210, SRX220, SRX240, SRX550M, SRX650

IS-IS extensions provide the basic interior gateway protocol (IGP) support for collecting intradomain routing information for Connectionless Network Service (CLNS) destinations within a CLNS network. Routers that learn host addresses through End System-to-Intermediate System (ES-IS) can advertise the addresses to other routers (intermediate systems) by using IS-IS.

For more information about IS-IS, see the ISO 10589 standard.

Related Documentation

- CLNS Overview on page 77
- Example: Configuring IS-IS for CLNS on page 85

Example: Configuring IS-IS for CLNS

Supported Platforms SRX1500, SRX300, SRX320, SRX340, SRX345, SRX550M, vSRX

This example shows how to create a routing instance and enable IS-IS protocol on all interfaces.

Requirements

Before you begin, configure the network interfaces. See Interfaces Feature Guide for Security Devices.

Overview

The configuration instructions in this topic describe how to create a routing-instance called aaaa, enable IS-IS on all interfaces, and define BGP export policy name (dist-bgp), family (ISO), and protocol (BP), and apply the export policy to IS-IS.

Configuration

CLI Quick Configuration

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

```
set routing-instances aaaa protocols isis clns-routing
set routing-instances aaaa protocols isis interface all
set routing-instances aaaa protocols isis no-ipv4-routing no-ipv6-routing
set policy-options policy-statement dist-bgp from family iso protocol bgp
set policy-options policy-statement dist-bgp then accept
set routing-instances aaaa protocols isis export dist-bgp
```

Step-by-Step Procedure

The following example requires you to navigate various levels in the configuration hierarchy. For instructions on how to do that, see *Using the CLI Editor in Configuration Mode*.

To configure IS-IS for CLNS:

1. Configure the routing instance.

   ```
   [edit]
   user@host# edit routing-instances aaaa
   ```

2. Enable CLNS routing.

   ```
   [edit routing-instances aaaa]
   user@host# set protocols isis clns-routing
   ```

3. Enable IS-IS on all interfaces.

   ```
   [edit routing-instances aaaa]
   user@host# set protocols isis interface all
   ```

4. (Optional) Disable IPv4 and IPv6 routing to configure a pure CLNS network .

   ```
   [edit routing-instances aaaa]
   user@host# set protocols isis no-ipv4-routing no-ipv6-routing
   ```

5. Define the BGP export policy name, family, and protocol.

   ```
   [edit policy-options]
   user@host# set policy-statement dist-bgp from family iso protocol bgp
   ```

6. Define the action for the export policy.

   ```
   [edit policy-options]
   user@host# set policy-statement dist-bgp then accept
   ```

7. Apply the export policy to IS-IS.

   ```
   [edit routing-instances aaaa]
   ```

```
user@host# set protocols isis export dist-bgp
```

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

```
[edit]
user@host# show routing-instances
aaaa {
  protocols {
    isis {
      export dist-bgp;
      no-ipv4-routing;
      no-ipv6-routing;
      clns-routing;
      interface all;
    }
```

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

Verification

Confirm that the configuration is working properly.

- Verifying Routing-Instance for CLNS on page 87
- Verifying IS-IS for CLNS on page 87

Verifying Routing-Instance for CLNS

Purpose Verify that the policy options are enabled for the routing instance.

Action From operational mode, enter the **show routing-instances** command.

Verifying IS-IS for CLNS

Purpose Verify that IS-IS is enabled.

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

Related Documentation

- CLNS Configuration Overview on page 77
- Understanding IS-IS for CLNS on page 85
- Verifying a CLNS VPN Configuration on page 97

CHAPTER 10

Configuring Static Routes for CLNS

Understanding Static Routes for CLNS

Supported Platforms MX Series, SRX100, SRX110, SRX210, SRX220, SRX240, SRX550M, SRX650

The Connectionless Network Service (CLNS) is an ISO Layer 3 protocol that uses network service access point (NSAP) reachability information instead of IPv4 or IPv6 prefixes.

You can configure static routes to exchange CLNS routes within a CLNS island. A *CLNS island* is typically an IS-IS level 1 area that is part of a single IGP routing domain. An island can contain more than one area. CLNS islands can be connected by VPNs.

Related Documentation

- Example: Configuring Static Routes for CLNS on page 89

Example: Configuring Static Routes for CLNS

Supported Platforms MX Series, SRX1500, SRX300, SRX320, SRX340, SRX345, SRX550M, vSRX

This example shows how to configure static routes for CLNS.

Requirements

Before you begin, configure the network interfaces. See *Interfaces Feature Guide for Security Devices*.

Overview

In this example, you configure static routes for CLNS. In the absence of an interior gateway protocol (IGP) on a certain link, a routing device might need to be configured with static routes for CLNS prefixes to be reachable by way of that link. This might be useful, for example, at an autonomous system (AS) boundary.

When you configure static routes for CLNS, consider the following tasks:

- Specify the **iso.0** routing table option to configure a primary instance CLNS static route.
- Specify the **instance-name.iso.0** routing table option to configure a CLNS static route for a particular routing instance.
- Specify the **route** ***nsap-prefix*** statement to configure the destination for the CLNS static route.
- Specify the **next-hop (*interface-name* | *iso-net*)** statement to configure the next hop, specified as an ISO network entity title (NET) or interface name.
- Include the **qualified-next-hop (*interface-name* | *iso-net*)** statement to configure a secondary backup next hop, specified as an ISO network entity title or interface name.

Configuration

CLI Quick Configuration

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

```
set routing-options rib iso.0 static iso-route 47.0005.80ff.f800.0000.ffff.ffff/152 next-hop
  47.0005.80ff.f800.0000.0108.0001.1921.6800.4212
set routing-options rib iso.0 static iso-route
  47.0005.80ff.f800.0000.0108.0001.1921.6800.4212/152 next-hop t1-0/2/2.0
set routing-options rib iso.0 static iso-route 47.0005.80ff.f800.0000.eee0/152
  qualified-next-hop 47.0005.80ff.f800.0000.0108.0001.1921.6800.4002 preference
  20
set routing-options rib iso.0 static iso-route 47.0005.80ff.f800.0000.eee0/152
  qualified-next-hop 47.0005.80ff.f800.0000.0108.0001.1921.6800.4002 metric 10
```

Step-by-Step Procedure

To configure static routes for CLNS:

1. Configure the routes.

```
[edit routing-options rib iso.0 static]
user@host# set iso-route 47.0005.80ff.f800.0000.ffff.ffff/152 next-hop
  47.0005.80ff.f800.0000.0108.0001.1921.6800.4212
user@host# set iso-route 47.0005.80ff.f800.0000.0108.0001.1921.6800.4212/152
  next-hop t1-0/2/2.0
user@host# set iso-route 47.0005.80ff.f800.0000.eee0/152 qualified-next-hop
  47.0005.80ff.f800.0000.0108.0001.1921.6800.4002 preference 20
user@host# set iso-route 47.0005.80ff.f800.0000.eee0/152 qualified-next-hop
  47.0005.80ff.f800.0000.0108.0001.1921.6800.4002 metric 10
```

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

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

Results Confirm your configuration by issuing the **show routing-options** command. If the output does not display the intended configuration, repeat the instructions in this example to correct the configuration.

```
user@host# show routing-options
rib iso.0 {
  static {
    iso-route 47.0005.80ff.f800.0000.ffff.ffff/152 next-hop
      47.0005.80ff.f800.0000.0108.0001.1921.6800.4212;
    iso-route 47.0005.80ff.f800.0000.0108.0001.1921.6800.4212/152 next-hop t1-0/2/2.0;
    iso-route 47.0005.80ff.f800.0000.eee0/152 {
      qualified-next-hop 47.0005.80ff.f800.0000.0108.0001.1921.6800.4002 {
        preference 20;
        metric 10;
      }
    }
  }
}
```

Verification

Checking the Routing Table

Purpose Make sure that the expected routes appear in the routing table.

Action

```
user@host> show route table iso.0

iso.0: 7 destinations, 7 routes (7 active, 0 holddown, 0 hidden)
+ = Active Route, - = Last Active, * = Both

47.0005.80ff.f800.0000.0108.0001.1921.6800.4212/152
                   *[Static/5] 00:00:25
                    > via t1-0/2/2.0
47.0005.80ff.f800.0000.eee0/84
                   *[Static/20] 00:04:01, metric 10, metric2 10
                    > to #75 0.12.0.34.0.56 via fe-0/0/1.0
47.0005.80ff.f800.0000.ffff.ffff/104
                   *[Static/5] 00:04:01, metric2 0
                    > via t1-0/2/2.0
```

Meaning The static routes appear in the routing table.

Related Documentation

- CLNS Configuration Overview on page 77
- Understanding Static Routes for CLNS on page 89

CHAPTER 11

Configuring BGP for CLNS

Understanding BGP for CLNS VPNs

Supported Platforms MX Series, SRX100, SRX110, SRX210, SRX220, SRX240, SRX550M, SRX650

BGP extensions allow BGP to carry Connectionless Network Service (CLNS) virtual private network (VPN) network layer reachability information (NLRI) between provider edge (PE) routers. Each CLNS route is encapsulated into a CLNS VPN NLRI and propagated between remote sites in a VPN.

CLNS is a Layer 3 protocol similar to IP version 4 (IPv4). CLNS uses network service access points (NSAPs) to address end systems. This allows for a seamless autonomous system (AS) based on International Organization for Standardization (ISO) NSAPs.

A single routing domain consisting of ISO NSAP devices are considered to be CLNS islands. CLNS islands are connected together by VPNs.

You can configure BGP to exchange ISO CLNS routes between PE routers connecting various CLNS islands in a VPN using multiprotocol BGP extensions. These extensions are the ISO VPN NLRIs.

Each CLNS network island is treated as a separate VPN routing and forwarding instance (VRF) instance on the PE router.

You can configure CLNS on the global level, group level, and neighbor level.

Related Documentation

- CLNS Overview on page 77
- Example: Configuring BGP for CLNS VPNs on page 93

Example: Configuring BGP for CLNS VPNs

Supported Platforms SRX1500, SRX300, SRX320, SRX340, SRX345, SRX550M, vSRX

This example shows how to create a BGP group for CLNS VPNs, define the BGP peer neighbor address for the group, and define the family.

- Requirements on page 94
- Overview on page 94
- Configuration on page 94
- Verification on page 94

Requirements

Before you begin, configure the network interfaces. See the *Interfaces Feature Guide for Security Devices*.

Overview

In this example, you create the BGP group called pedge-pedge, define the BGP peer neighbor address for the group as 10.255.245.215, and define the BGP family.

Configuration

CLI Quick Configuration

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

```
set protocols bgp group pedge-pedge neighbor 10.255.245.213
set protocols bgp family iso-vpn unicast
```

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 BGP for CLNS VPNs:

1. Configure the BGP group and define the BGP peer neighbor address.

   ```
   [edit protocols bgp]
   user@host# set group pedge-pedge neighbor 10.255.245.213
   ```

2. Define the family.

   ```
   [edit protocols bgp]
   user@host# set family iso-vpn unicast
   ```

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

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

Verification

Verifying the Neighbor Status

Purpose Display information about the BGP peer.

Action From operational mode, run the **show bgp neighbor 10.255.245.213** command. Look for **iso-vpn-unicast** in the output.

```
user@host> show bgp neighbor 10.255.245.213
Peer: 10.255.245.213+179 AS 200 Local: 10.255.245.214+3770 AS 100
Type: External State: Established Flags: <ImportEval Sync>
Last State: OpenConfirm Last Event: RecvKeepAlive
Last Error: None
Options: <Multihop Preference LocalAddress HoldTime AddressFamily PeerAS
Rib-group Refresh>
Address families configured: iso-vpn-unicast
Local Address: 10.255.245.214 Holdtime: 90 Preference: 170
Number of flaps: 0
Peer ID: 10.255.245.213 Local ID: 10.255.245.214 Active Holdtime: 90
Keepalive Interval: 30 Peer index: 0
NLRI advertised by peer: iso-vpn-unicast
NLRI for this session: iso-vpn-unicast
Peer supports Refresh capability (2)
Table bgp.isovpn.0 Bit: 10000
RIB State: BGP restart is complete
RIB State: VPN restart is complete
Send state: in sync
Active prefixes: 3
Received prefixes: 3
Suppressed due to damping: 0
Advertised prefixes: 3
Table aaaa.iso.0
RIB State: BGP restart is complete
RIB State: VPN restart is complete
Send state: not advertising
Active prefixes: 3
Received prefixes: 3
Suppressed due to damping: 0
Last traffic (seconds): Received 6 Sent 5 Checked 5
Input messages: Total 1736 Updates 4 Refreshes 0 Octets 33385
Output messages: Total 1738 Updates 3 Refreshes 0 Octets 33305
Output Queue[0]: 0
Output Queue[1]: 0
```

Related Documentation

- CLNS Configuration Overview on page 77
- Understanding BGP for CLNS VPNs on page 93
- Verifying a CLNS VPN Configuration on page 97

Example: Configuring a VPN Routing Instance for CLNS

Supported Platforms SRX1500, SRX300, SRX320, SRX340, SRX345, SRX550M, vSRX

This example shows how to create a CLNS routing instance and set the instance type for Layer 3 VPNs.

- Requirements on page 96
- Overview on page 96
- Configuration on page 96
- Verification on page 97

Requirements

Before you begin, configure the network interfaces. See *Interfaces Feature Guide for Security Devices*.

Overview

The following example shows how to create a CLNS routing instance called aaaa and set the instance type to VRF for Layer 3 VPNs. Within the example, you specify that the lo0.1 interface, e1–2/0/0.0 interface, and t1–3/0/0.0 interface all belong to the routing instance. The route distinguisher is set as 10.255.245.1:1 and the policy for the Layer 3 VRF table is set as target:11111:1.

Configuration

CLI Quick Configuration

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

```
set routing-instances aaaa instance-type vrf
set routing-instances aaaa interface lo0.1
set routing-instances aaaa interface ge-0/0/3
set routing-instances aaaa interface ge-0/0/2
set routing-instances aaaa route-distinguisher 10.255.245.1:1
set routing-instances aaaa vrf-target target:11111:1
```

Step-by-Step Procedure

The following example requires you to navigate various levels in the configuration hierarchy. For instructions on how to do that, see *Using the CLI Editor in Configuration Mode*.

To configure a VPN routing instance:

1. Create the routing instance.

   ```
   [edit]
   user@host# edit routing-instances aaaa
   ```

2. Specify the routing instance type.

   ```
   [edit routing-instances aaaa]
   user@host# set instance-type vrf
   ```

3. Specify the interfaces that belong to the routing instance.

   ```
   [edit routing-instances aaaa]
   user@host# set interface lo0.1
   user@host# set interface ge-0/0/3
   user@host# set interface ge-0/0/2
   ```

4. Specify the route distinguisher.

   ```
   [edit routing-instances aaaa]
   user@host# set route-distinguisher 10.255.245.1:1
   ```

5. Specify the policy for the Layer 3 VRF table.

```
[edit routing-instances aaaa]
user@host# set vrf-target target:11111:1
```

6. Enable family ISO on the interfaces edit interfaces interface-name unit-id.

```
[edit routing-instances aaaa]
user@host# set family ISO
```

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

```
[edit ]
user@host# show routing-instances
instance-type vrf;
interface ge-0/0/2.0;
interface ge-0/0/3.0;
interface lo0.1;
route-distinguisher 10.255.245.1:1;
vrf-target target:11111:1;
```

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

Verification

Verifying the Configured CLNS Routing Instance

Purpose Confirm that the configuration is working properly.

Verify that the CLNS routing instance is configured.

Action From operational mode, enter the **show routing-instances** command.

Related Documentation

- CLNS Configuration Overview on page 77
- Verifying a CLNS VPN Configuration on page 97

Verifying a CLNS VPN Configuration

Supported Platforms SRX1500, SRX300, SRX320, SRX340, SRX345, SRX550M, vSRX

Purpose Verify that the device is configured correctly for CLNS VPNs.

Action From configuration mode in the CLI, enter the **show** command.

```
[edit]
user@host# show
interfaces {
  e1–2/0/0.0 {
    unit 0 {
      family inet {
        address 192.168.37.51/31;
      }
```

```
                family iso;
                family mpls;
            }
        }
        t1–3/0/0.0 {
            unit 0 {
                family inet {
                    address 192.168.37.24/32;
                }
                family iso;
                family mpls;
            }
        }
        lo0 {
            unit 0 {
                family inet {
                    address 127.0.0.1/32;
                    address 10.255.245.215/32;
                }
                family iso {
                    address 47.0005.80ff.f800.0000.0108.0001.1921.6800.4215.00;
                }
            }
            unit 1 {
                family iso {
                    address 47.0005.80ff.f800.0000.0108.aaa2.1921.6800.4215.00;
                }
            }
        }
    }
    routing-options {
        autonomous-system 230;
    }
    protocols {
        bgp {
            group pedge-pedge {
                type internal;
                local-address 10.255.245.215;
                neighbor 10.255.245.212 {
                    family iso-vpn {
                        unicast;
                    }
                }
            }
        }
    }
    policy-options {
        policy-statement dist-bgp {
            from {
                protocol bgp;
                family iso;
            }
            then accept;
        }
    }
    routing-instances {
        aaaa {
            instance-type vrf;
            interface lo0.1;
            interface e1–2/0/0.0;
            interface t1–3/0/0.0;
            route-distinguisher 10.255.245.1:1;
            vrf-target target:11111:1;
            routing-options {
                rib aaaa.iso.0 {
                    static {
                        iso-route 47.0005.80ff.f800.0000.bbbb.1022/104
                            next-hop 47.0005.80ff.f800.0000.aaaa.1000.1921.6800.4196.00;
                    }
                }
            }
            protocols {
                esis {
                    interface all;
                }
                isis {
                    export dist-bgp;
                    no-ipv4–routing;
                    no-ip64–routing;
                    clns–routing;
                    interface all;
                }
            }
        }
    }
```

Related Documentation

- CLNS Configuration Overview on page 77

PART 5

Configuring VPLS

CHAPTER 12

Introduction to VPLS

VPLS Overview

Supported Platforms SRX1500, SRX300, SRX320, SRX340, SRX345, SRX550M, vSRX

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

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

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

NOTE: The RSVP automatic mesh feature with multiple RSVP neighbors on a single LAN is not supported on branch SRX Series devices because RSVP runs on WAN links in a service provider network. Most of these WAN interfaces are point-to-point and are rarely seen in LAN networks.

This topic contains the following sections:

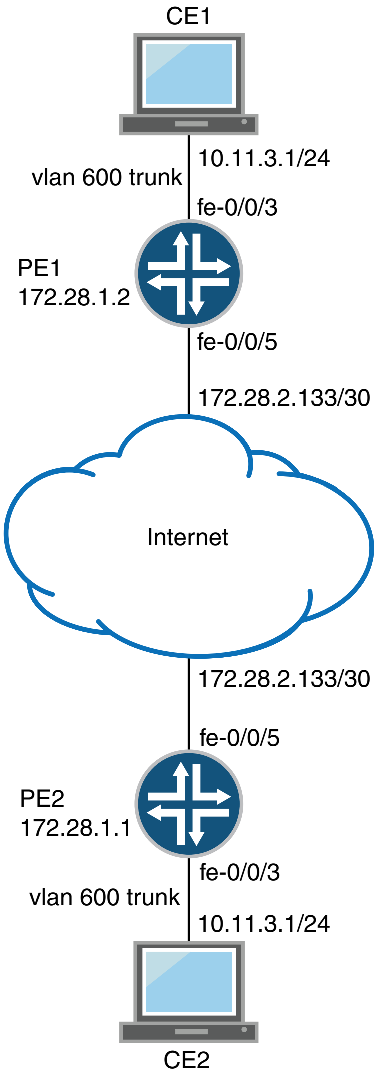
- Sample VPLS Topology on page 104
- VPLS on PE Routers on page 105

- Using an Ethernet Switch as the VPLS CE Device on page 107
- VPLS Exceptions on SRX Series Devices on page 107

Sample VPLS Topology

Figure 8 on page 104 shows a basic VPLS topology.

Figure 8: Basic VPLS Topology



In this sample, the PE routers use the same autonomous system (AS). Within the AS, routing information is communicated through an interior gateway protocol (IGP). Outside the AS, routing information is shared with other ASs through BGP. The PE routers must use the same signaling protocols to communicate.

VPLS on PE Routers

Within a VPLS configuration, a device running Junos OS can act as a PE router. Junos OS passes the VPLS traffic through the following ports and PIMs on the Juniper Networks device to CE routers in the VPLS network:

- Built-in Ethernet ports on front panel
- Gigabit Ethernet uPIMs
- Gigabit Ethernet ePIMs
- Fast Ethernet PIMs
- Fast Ethernet ePIMs

NOTE: **Ports on uPIMs and ePIMs must be in routing mode before you can configure the corresponding interfaces for VPLS.**

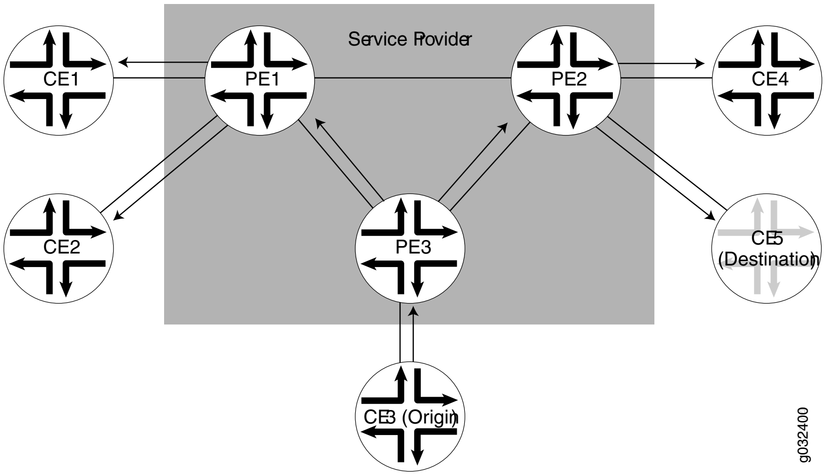
Because a VPLS carries Ethernet traffic across a service provider network, it must mimic an Ethernet network in some ways. When a PE router configured with a VPLS routing instance receives a packet from a CE device, it first determines whether it has the destination of the VPLS packet in the appropriate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

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

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

If the PE router cannot determine the destination of the VPLS packet, it floods the packet to all attached CE devices. Figure 9 on page 106 illustrates this process.

Figure 9: Flooding a Packet with an Unknown Destination



A VPLS interface can be directly connected to an Ethernet switch. Layer 2 information gathered by an Ethernet switch, for example, MAC addresses and interface ports, is included in the VPLS routing instance table.

An MPLS label-switched interface (LSI) label is used as the inner label for VPLS. This label maps to a VPLS routing instance on the ingress PE router. On the egress PE router, the LSI label is stripped and then mapped to a logical LSI interface. The Layer 2 Ethernet frame is then forwarded using the LSI interface to the correct VPLS routing instance.

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

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

Using an Ethernet Switch as the VPLS CE Device

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

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

VPLS Exceptions on SRX Series Devices

The VPLS implementation on SRX Series device is similar to VPLS implementations on M Series, T Series, and MX Series routers, with the following exceptions:

- SRX Series devices do not support aggregated Ethernet interfaces. Therefore, aggregated Ethernet interfaces between CE devices and PE routers are not supported for VPLS routing instances on SRX Series devices.
- SRX Series devices do not support aggregated Ethernet interfaces. Therefore, aggregated Ethernet interfaces between PE devices and PE routers are not supported for VPLS routing instances on SRX Series devices.
- VPLS multihoming, which allows connecting a CE device to multiple PE routers to provide redundant connectivity, is not supported on SRX Series devices.
- SRX Series devices do not support BGP mesh groups.
- SRX Series devices support only the following encapsulation types on VPLS interfaces that face CE devices: extended VLAN VPLS, Ethernet VPLS, and VLAN VPLS. Ethernet VPLS over ATM LLC encapsulation is not supported.
- Virtual ports are generated dynamically on a Tunnel Services PIC on some Juniper Networks routing platforms. SRX Series devices do not support Tunnel Services modules or virtual ports.
- The VPLS implementation on SRX Series devices does not support dual-tagged frames. Therefore, VLAN rewrite operations are not supported on dual-tagged frames. VLAN rewrite operations such as pop-pop, pop-swap, push-push, swap-push, and swap-swap, which are supported on M Series and T Series routing platforms, are not supported on SRX Series devices.

Related Documentation

- MPLS Layer 2 VPN Configuration Overview on page 61
- Understanding VPLS Interfaces on page 111
- Understanding VPLS Routing Instances on page 121

- Understanding VPLS VLAN Encapsulation on page 167
- Understanding VPLS VLAN Encapsulation on a Logical Interface on page 168
- VPLS Configuration Overview on page 108

VPLS Configuration Overview

Supported Platforms SRX1500, SRX300, SRX320, SRX340, SRX345, SRX550M, vSRX

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

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

To configure VPLS:

1. Determine which uPIM and ePIM ports correspond to the interfaces that will carry the VPLS traffic and enable routing mode on those ports.
2. Configure the interfaces that will carry the VPLS traffic between the PE router and CE devices. On the PE router interfaces that are facing the CE devices, specify a VPLS encapsulation type. The type of encapsulation depends on the interface type. See“Example: Configuring Routing Interfaces on the VPLS PE Router” on page 113 and “Example: Configuring the Interface to the VPLS CE Device” on page 114.
3. Create a VPLS routing instance on each PE router that is participating in the VPLS. For each VPLS routing instance, specify which interfaces will carry the VPLS traffic between the PE and CE devices. On the CE device interface that faces the PE router, you must specify inet (for IPv4), and include the IP address. Additionally, each routing instance must have a unique route distinguisher associated with it. (VPN routing instances need a route distinguisher to help BGP identify overlapping network layer reachability information (NLRI) messages from different VPNs.) See “Example: Configuring the VPLS Routing Instance” on page 124.
4. Configure routing options on the PE router. See “Example: Configuring Routing Options on the VPLS PE Router” on page 165.
5. Configure MPLS LSPs between the PE routers. See “Example: Configuring MPLS on the VPLS PE Router” on page 131.
6. Configure RSVP on the PE routers. Enable RSVP for all connections that participate in the MPLS LSP. See “Example: Configuring RSVP on the VPLS PE Router” on page 130.
7. Configure an IBGP session between PE routers so that the routers can exchange information about routes originating and terminating in the VPLS. See “Example: Configuring BGP on the VPLS PE Router” on page 164.

8. Configure an IGP on the PE routers to exchange routing information. See “Example: Configuring OSPF on the VPLS PE Router” on page 129.
9. Configure VLAN encapsulation. See “Example: Configuring VPLS VLAN Encapsulation on Gigabit Ethernet Interfaces” on page 171,“Example: Configuring VPLS VLAN Encapsulation” on page 168, and “Example: Configuring Extended VLAN VPLS Encapsulation” on page 172.

Related Documentation

- MPLS Layer 2 VPN Configuration Overview on page 61
- MPLS Layer 2 VPN Configuration Overview on page 61
- Example: Configuring MPLS on the VPLS PE Router on page 131
- Example: Configuring RSVP on the VPLS PE Router on page 130
- Example: Configuring BGP on the VPLS PE Router on page 164
- Example: Configuring OSPF on the VPLS PE Router on page 129

CHAPTER 13

Configuring Interfaces

Understanding VPLS Interfaces

Supported Platforms SRX1500, SRX300, SRX320, SRX340, SRX345, SRX550M, vSRX

For each VPLS routing instance on a PE router, you specify which interfaces are to be used to carry VPLS traffic between the PE and CE devices.

This topic contains the following sections:

Interface Name

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

physical.logical

For example, in ge-1/2/1.2, ge-1/0/1 is the physical portion of the interface name and 2 is the logical portion. If you do not specify the logical portion of the interface name, 0 is set by default. A logical interface can be associated with only one routing instance.

Encapsulation Type

The physical link-layer encapsulation type for a VPLS interface can be one of the following:

- **ethernet-vpls**—Use Ethernet VPLS encapsulation on Ethernet interfaces that have VPLS enabled and that must accept packets carrying standard Tag Protocol Identifier (TPID) values.
- **extended-vlan-vpls**—Use extended virtual LAN (VLAN) VPLS encapsulation on Ethernet interfaces that have VLAN 802.1Q tagging and VPLS enabled and that must accept packets carrying TPIDs 0x8100, 0x9100, and 0x9901. All VLAN IDs from 1 through 1023 are valid for VPLS VLANs on Fast Ethernet interfaces, and all VLAN IDs from 1 through 4094 are valid for VPLS VLANs on Gigabit Ethernet interfaces.
- **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. You must configure this encapsulation type on both the physical interface and the logical interface. VLAN IDs 1 through 511 are reserved for normal Ethernet VLANs, IDs 512 through 1023 are reserved for VPLS VLANs on Fast Ethernet interfaces, and IDs 512 through 4094 are reserved for VPLS VLANs on Gigabit Ethernet interfaces.
- **flexible-ethernet-services**—Use flexible Ethernet services encapsulation when you want to configure multiple per-unit Ethernet encapsulations. This encapsulation type allows you to configure any combination of route, TCC, CCC, and VPLS encapsulations on a single physical port. Aggregated Ethernet bundles cannot use this encapsulation type.

  For flexible Ethernet services encapsulation, VLAN IDs from 1 through 511 are no longer reserved for normal VLANs.

Flexible VLAN Tagging

For untagged packets to be accepted on an 802.1Q VLAN-tagged port, specify the native VLAN ID with the flexible VLAN tagging option. (No other flexible VLAN tagging features are supported.)

VLAN Rewrite

You can rewrite VLAN tags on VPLS interfaces. Rewriting VLAN tags allows you to use an additional (outer) VLAN tag to differentiate between CE devices that share a VLAN ID.

You can configure rewrite operations to stack (push), remove (pop), or rewrite (swap) tags on single-tagged frames. If a port is not configured for VLAN tagging, rewrite operations are not supported on any logical interface on that port.

You can configure the following VLAN rewrite operations:

- pop—Remove a VLAN tag from the top of the VLAN tag stack. The outer VLAN tag of the frame is removed.
- push—Add a new VLAN tag to the top of the VLAN stack. An outer VLAN tag is pushed in front of the existing VLAN tag.
- swap—Replace the VLAN tag at the top of the VLAN tag stack with a user-specified VLAN tag value.

You perform VLAN rewrite operations by applying input and output VLAN maps at the ingress and egress, respectively, of the interface. For incoming frames, use the input-vlan-map; for outgoing frames, use the output-vlan-map.

The VPLS implementation on SRX Series devices does not support dual-tagged frames. Therefore, VLAN rewrite operations are not supported on dual-tagged frames. VLAN rewrite operations such as pop-pop, pop-swap, push-push, swap-push, and swap-swap, which are supported on M Series and T Series routing platforms, are not supported on SRX Series devices.

Related Documentation

- Example: Configuring Routing Interfaces on the VPLS PE Router on page 113
- Example: Configuring the Interface to the VPLS CE Device on page 114
- VPLS Configuration Overview on page 108
- VPLS Overview on page 103
- Understanding VPLS VLAN Encapsulation on page 167

Example: Configuring Routing Interfaces on the VPLS PE Router

Supported Platforms SRX1500, SRX300, SRX320, SRX340, SRX345, SRX550M, vSRX

This example shows how to configure routing interfaces on the VPLS PE router.

- Requirements on page 113
- Overview on page 113
- Configuration on page 113
- Verification on page 114

Requirements

Before you begin, see *Understanding Selective Stateless Packet-Based Services* .

Overview

In this example, you configure the PE1 router loopback interface and the interface to the PE2 router ge-2/0/1.

Configuration

Step-by-Step Procedure

To configure the routing interface on the VPLS PE router:

1. Configure the loopback interface.

```
[edit]
user@host# set interfaces lo0 unit 0 family inet address 10.255.7.168/32 primary
```

2. Configure the IP address on the MPLS core interface.

```
[edit]
user@host# set interfaces ge-3/0/2 unit 0 family inet address 100.1.1.1/30
```

3. Configure the MPLS family.

```
[edit]
user@host# set interfaces ge-3/0/2 unit 0 family mpls
```

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

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

Verification

To verify the configuration is working properly, enter the **show interfaces** command.

Related Documentation

- *Interfaces Feature Guide for Security Devices*
- VPLS Configuration Overview on page 108
- Understanding VPLS Interfaces on page 111

Example: Configuring the Interface to the VPLS CE Device

Supported Platforms SRX1500, SRX300, SRX320, SRX340, SRX345, SRX550M, vSRX

This example shows how to configure the router interface that is connected to the CE device to include VPLS encapsulation.

- Requirements on page 114
- Overview on page 114
- Configuration on page 114
- Verification on page 115

Requirements

Before you begin, see *Understanding Selective Stateless Packet-Based Services* .

Overview

In this example, you configure the router interface ge-1/2/1 that is connected to the CE device to include VPLS encapsulation.

Configuration

Step-by-Step Procedure

To configure the interface to the VPLS CE device:

1. Configure VPLS encapsulation for the interface facing the CE router.

```
[edit]
user@host# set interfaces ge-1/2/1 encapsulation ethernet-vpls
```

2. Configure the interface for the VPLS family group.

```
[edit]
user@host# set interfaces ge-1/2/1 unit 0 family vpls
```

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

```
[edit]
```

```
user@host# commit
```

Verification

To verify the configuration is working properly, enter the **show interfaces ge-1/2/1** command.

Related Documentation

- VPLS Configuration Overview on page 108
- Understanding VPLS Interfaces on page 111

VPLS Filters and Policers Overview

Supported Platforms SRX1500, SRX300, SRX320, SRX340, SRX345, SRX550M, vSRX

This feature permits users to configure both firewall filters and policers for virtual private LAN service (VPLS). Firewall filters enable you to filter packets based on their components and perform an action on packets that match the filter. Policers enable you to limit the amount of traffic that passes into or out of an interface.

This feature can be enabled by configuring VPLS filters, policers, and accounting through various CLI commands. VPLS filters and policers act on a Layer 2 frame that includes the media access control (MAC) header (after any VLAN rewrite or other rules are applied), but that does not include the cyclical redundancy check (CRC) field.

NOTE: You can apply VPLS filters and policers on the PE routers only to customer-facing (PE-CE) interfaces.

Related Documentation

- Example: Configuring VPLS Policers on page 118
- Example: Configuring VPLS Filters on page 115

Example: Configuring VPLS Filters

Supported Platforms SRX1500, SRX300, SRX320, SRX340, SRX345, SRX550M, vSRX

This example shows how to configure VPLS filters.

- Requirements on page 116
- Overview on page 116
- Configuration on page 116
- Verification on page 117

Requirements

Before you begin:

- Configure the interfaces that will carry the VPLS traffic between the PE router and the CE devices. See "Example: Configuring Routing Interfaces on the VPLS PE Router" on page 113 and "Example: Configuring the Interface to the VPLS CE Device" on page 114.
- Create a VPLS routing instance on each PE router that is participating in the VPLS. See "Example: Configuring the VPLS Routing Instance" on page 124.
- Configure an IGP on the PE routers to exchange routing information. See "Example: Configuring OSPF on the VPLS PE Router" on page 129.
- Configure RSVP-TE on the PE routers. See "Example: Configuring RSVP on the VPLS PE Router" on page 130.

Overview

This example describes how to configure filtering and accounting for VPLS.

CAUTION: **MPLS is disabled by default on SRX Series devices. You must explicitly configure your device to allow MPLS traffic. However, when MPLS is enabled, all flow-based security features are deactivated and the device performs packet-based processing. Flow-based services such as security policies, zones, NAT, ALGs, chassis clustering, screens, firewall authentication, and IPsec VPNs are unavailable on the device.**

Configuration

CLI Quick Configuration

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

```
set firewall family vpls filter blue term term1 from interface ge-3/0/0.512
set firewall family vpls filter blue term term1 from interface fe-5/0/0.512
set firewall family vpls filter blue term term1 then count count1
set firewall family vpls filter blue accounting-profile fw_profile
set accounting-options file fw_acc size 500k
set accounting-options file fw_acc transfer-interval 5
set accounting-options filter-profile fw_profile file fw_acc
set accounting-options filter-profile fw_profile interval 1
set accounting-options filter-profile fw_profile counters count1
set interfaces ge-0/0/1 unit 512 family vpls filter input blue
```

Step-by-Step Procedure

To configure filters for VPLS:

1. Configure a filter with a GE interface as the match condition and count as the action.

```
[edit ]
user@host# set firewall family vpls filter blue term term1 from interface ge-3/0/0.512
```

2. Configure a filter with an FE interface as the match condition and count as the action.

   ```
   [edit ]
   user@host# set firewall family vpls filter blue term term1 from interface fe-5/0/0.512
   ```

3. Configure the count.

   ```
   [edit ]
   user@host# set firewall family vpls filter blue term term1 then count count1
   ```

4. Configure the accounting profile to refer it to the counter.

   ```
   [edit ]
   user@host# set firewall family vpls filter blue accounting-profile fw_profile
   ```

5. Configure the account file size.

   ```
   [edit ]
   user@host# set accounting-options file fw_acc size 500k
   ```

6. Configure the account transfer interval.

   ```
   [edit ]
   user@host# set accounting-options file fw_acc transfer-interval 5
   ```

7. Configure the filter for the accounting profile.

   ```
   [edit ]
   user@host# set accounting-options filter-profile fw_profile file fw_acc
   ```

8. Configure the filter for the interval.

   ```
   [edit ]
   user@host# set accounting-options filter-profile fw_profile interval 1
   ```

9. Configure the counter.

   ```
   [edit ]
   user@host# set accounting-options filter-profile fw_profile counters count1
   ```

10. Apply the filter to the interface.

    ```
    [edit ]
    user@host# set interfaces ge-0/0/1 unit 512 family vpls filter input blue
    ```

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

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

Verification

To verify the configuration is working properly, enter the **show firewall** and **show accounting records** commands.

Related Documentation

- VPLS Filters and Policers Overview on page 115
- VPLS Configuration Overview on page 108
- Example: Configuring VPLS Policers on page 118

Example: Configuring VPLS Policers

Supported Platforms SRX1500, SRX300, SRX320, SRX340, SRX345, SRX550M, vSRX

This example shows how to configure VPLS policers.

- Requirements on page 118
- Overview on page 118
- Configuration on page 118
- Verification on page 119

Requirements

Before you begin:

- Configure the interfaces that will carry the VPLS traffic between the PE router and the CE devices. See "Example: Configuring Routing Interfaces on the VPLS PE Router" on page 113 and "Example: Configuring the Interface to the VPLS CE Device" on page 114.
- Create a VPLS routing instance on each PE router that is participating in the VPLS. See "Example: Configuring the VPLS Routing Instance" on page 124.
- Configure an IGP on the PE routers to exchange routing information. See "Example: Configuring OSPF on the VPLS PE Router" on page 129.
- Configure RSVP-TE on the PE routers. See "Example: Configuring RSVP on the VPLS PE Router" on page 130.

Overview

This example describes how to configure policing and apply it on the interface for VPLS.

CAUTION: MPLS is disabled by default on SRX Series devices. You must explicitly configure your device to allow MPLS traffic. However, when MPLS is enabled, all flow-based security features are deactivated and the device performs packet-based processing. Flow-based services such as security policies, zones, NAT, ALGs, chassis clustering, screens, firewall authentication, and IPsec VPNs are unavailable on the device.

Configuration

CLI Quick Configuration To quickly configure VPLS policers, copy the following commands, paste them into a text file, remove any line breaks, change any details necessary to match your network configuration, copy and paste the commands into the CLI at the **[edit]** hierarchy level, and then enter **commit** from configuration mode.

```
set firewall policer police2 if-exceeding bandwidth-percent 10
set firewall policer police2 if-exceeding burst-size-limit 1500
set firewall policer police2 then discard
set interfaces ge-0/0/1 unit 512 family vpls policer input police2
```

Step-by-Step Procedure

To configure filters for VPLS:

1. Configure bandwidth percentage.

   ```
   [edit ]
   user@host# set firewall policer police2 if-exceeding bandwidth-percent 10
   ```

2. Configure the burst size limit.

   ```
   [edit ]
   user@host# set firewall policer police2 if-exceeding burst-size-limit 1500
   ```

3. Configure the terminal action on the packet.

   ```
   [edit ]
   user@host# set firewall policer police2 then discard
   ```

4. Apply the policer to the interface.

   ```
   [edit ]
   user@host# set interfaces ge-0/0/1 unit 512 family vpls policer input police2
   ```

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

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

Verification

To verify the configuration is working properly, enter the **show firewall** command.

Related Documentation

- VPLS Filters and Policers Overview on page 115
- VPLS Configuration Overview on page 108
- Example: Configuring VPLS Filters on page 115

CHAPTER 14

Configuring Routing Instances

Understanding VPLS Routing Instances

Supported Platforms SRX1500, SRX300, SRX320, SRX340, SRX345, SRX550M, vSRX

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

You create a VPLS routing instance on each PE router that is participating in the VPLS. The routing instance has the same name on each PE router. To configure the VPLS routing instance, you specify the following:

- Route distinguisher—Helps BGP distinguish between potentially identical network layer reachability information (NLRI) messages received from different VPLS instances. Each routing instance that you configure on a PE router must have a unique route distinguisher.
- Route target—Defines which route is part of a VPLS. A unique route target helps distinguish between different VPLS services on the same router.
- Site name—Provides unique name for the VPLS site.
- Site identifier—Provides unique numerical identifier for the VPLS site.
- Site range—Specifies total number of sites in the VPLS. The site range must be greater than the site identifier.
- Interface to the CE router—Specifies the physical interface to the CE router that carries VPLS traffic. The interface must be configured for a VPLS encapsulation type.

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

CAUTION: MPLS is disabled by default on SRX Series devices. You must explicitly configure your router to allow MPLS traffic. However, when MPLS is enabled, all flow-based security features are deactivated and the router performs packet-based processing. Flow-based services such as security policies, zones, NAT, ALGs, chassis clustering, screens, firewall authentication, and IPsec VPNs are unavailable on the router.

This topic contains the following sections:

- BGP Signaling on page 122
- VPLS Routing Table on page 122
- Trace Options on page 123

BGP Signaling

BGP is used to signal the paths between each of the PE routers participating in the VPLS routing instance. These paths carry VPLS traffic across the service provider's network between the VPLS sites.

NOTE: LDP signaling is not supported for the VPLS routing instance.

To configure BGP signaling, you specify the following:

- VPLS site name and site identifier—When you configure BGP signaling for the VPLS routing instance, you must specify each VPLS site that has a connection to the router. For each VPLS site, you must configure a site name and site identifier (a numerical identifier between 1 to 65,534 that uniquely identifies the VPLS site).
- Site range—When you enable BGP signaling for the VPLS routing instance, you need to configure a site range. The site range specifies the total number of sites in the VPLS.

  NOTE: The site range value must be greater than the largest site identifier.

- Site preference—You can specify the preference value advertised for a particular VPLS site. The site preference value is encoded in the BGP local preference attribute. When a PE router receives multiple advertisements with the same VPLS edge (VE) device identifier, the advertisement with the highest local preference value is preferred.

VPLS Routing Table

The VPLS routing table contains MAC addresses and interface information for both physical and virtual ports. You can configure the following characteristics for the table:

- Table size—You can modify the size of the VPLS MAC address table. The default table size is 512 MAC addresses; the minimum is 16 addresses, and the maximum is 65,536 addresses.

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

The interfaces affected include all of the interfaces within the VPLS routing instance, including the local interfaces and the LSI interfaces.

- Timeout interval—You can modify the timeout interval for the VPLS table. The default timeout interval is 300 seconds; the minimum is 10 seconds, and the maximum is 1,000,000 seconds. We recommend you configure longer values for small, stable VPLS networks and shorter values for large, dynamic VPLS networks. If the VPLS table does not receive any updates during the timeout interval, the router waits one additional interval before automatically clearing the MAC address entries from the VPLS table.
- Number of addresses learned from an interface—You can configure a limit on the number of MAC addresses learned by a VPLS routing instance by setting the MAC table size. The default is 512 addresses; the minimum is 16, and the maximum is 65,536 addresses. If the MAC table limit is reached, new MAC addresses can no longer be added to the table. Eventually the oldest MAC addresses are removed from the MAC address table automatically. This frees space in the table, allowing new entries to be added. However, as long as the table is full, new MAC addresses are dropped.

  Because this limit applies to each VPLS routing instance, the MAC addresses of a single interface can consume all the available space in the table, preventing the routing instance from acquiring addresses from other interfaces. You can limit the number of MAC addresses learned from all interfaces configured for a VPLS routing instance, as well as limit the number of MAC addresses learned from a specific interface.

  The MAC limit configured for an individual interface overrides the limit configured for all interfaces for the VPLS routing instance. Also, the table limit can override the limits configured for the interfaces.

  The MAC address limit applies only to interfaces to CE devices.

Trace Options

The following trace flags display operations associated with VPLS:

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

Related Documentation

- Example: Configuring the VPLS Routing Instance on page 124
- Example: Configuring Routing Options on the VPLS PE Router on page 165

- VPLS Configuration Overview on page 108
- VPLS Overview on page 103
- Understanding VPLS VLAN Encapsulation on page 167

Example: Configuring the VPLS Routing Instance

Supported Platforms SRX1500, SRX300, SRX320, SRX340, SRX345, SRX550M, vSRX

This example shows how to create a VPLS routing instance on each PE router that is participating in the VPLS.

Requirements

Before you begin:

- Before you begin, see *Understanding Selective Stateless Packet-Based Services* .
- Configure the interfaces that will carry the VPLS traffic between the PE router and the CE devices. See “Example: Configuring Routing Interfaces on the VPLS PE Router” on page 113 and “Example: Configuring the Interface to the VPLS CE Device” on page 114.

Overview

This example describes how to create a VPLS routing instance; configure VPLS site identifier, site range, no tunnel services option, route distinguisher, and route target for the VPLS routing instance; and specify the VPLS interface to the CE router.

NOTE: You must specify no tunnel services in the VPLS routing instance configuration, because SRX Series devices do not support tunnel serial PICs.

Configuration

CLI Quick Configuration

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

```
set routing-instances green instance-type vpls
set routing-instances green protocols vpls site-range 10 site R3 site-identifier 2
set routing-instances green protocols vpls no-tunnel-services
set routing-instances green route-distinguisher 10.255.7.1:1
set routing-instances green vrf-target target:11111:1
set routing-instances green instance-type vpls interface ge-1/2/1.0
```

Step-by-Step Procedure

The following example requires you to navigate various levels in the configuration hierarchy. For instructions on how to do that, see *Using the CLI Editor in Configuration Mode*.

To configure a VPLS routing instance:

1. Configure the routing instance of type VPLS.

   ```
   [edit]
   user@host# edit routing-instances green
   ```

2. Enable the VPLS instance type.

   ```
   [edit routing-instances green]
   user@host# set instance-type vpls
   ```

3. Configure the VPLS site identifier and range for the VPLS routing instance.

   ```
   [edit routing-instances green protocols vpls]
   user@host# set site-range 10 site R3 site-identifier 2
   ```

4. Configure the no-tunnel-services option for the VPLS routing instance.

   ```
   [edit routing-instances green protocols vpls]
   user@host# set no-tunnel-services
   ```

5. Configure the route distinguisher.

   ```
   [edit routing-instances green]
   user@host# set route-distinguisher 10.255.7.1:1
   ```

6. Configure the route target.

   ```
   [edit routing-instances green]
   user@host# set vrf-target target:11111:1
   ```

7. Specify the VPLS interface to the CE router.

   ```
   [edit routing-instances green]
   user@host# set instance-type vpls interface ge-1/2/1.0
   ```

Results

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

```
[edit]
user@host# show routing-instances green
instance-type vpls;
interface ge-1/2/1.0;
route-distinguisher 10.255.7.1:1;
vrf-target target:11111:1;
protocols {
    vpls {
        site-range 10;
        no-tunnel-services;
        site R3 {
            site-identifier 2;
        }
    }
}
```

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

Verification

Confirm that the configuration is working properly.

- Verifying VPLS Routing Instance Is Configured on page 126
- Verifying VPLS Routing Attributes Are Configured on page 126

Verifying VPLS Routing Instance Is Configured

Purpose Verify that the VPLS routing instance is configured.

Action From operational mode, enter the **show routing-instances** command.

Verifying VPLS Routing Attributes Are Configured

Purpose Verify that attributes such as VPLS site identifier, site range, no tunnel services option, route distinguisher, and route target for the VPLS routing instance are configured.

Action From operational mode, enter the **show routing-instances green protocols vpls** command.

Related Documentation

- VPLS Configuration Overview on page 108
- Understanding VPLS Routing Instances on page 121

Example: Configuring Automatic Site Identifiers for VPLS

Supported Platforms SRX Series, vSRX

This example shows how to configure automatic site identifiers for VPLS sites.

Requirements

Before you begin, see information on selective stateless packet-based services in *Interfaces Feature Guide for Security Devices*.

Overview

When you enable automatic site identifiers, the Junos OS automatically assigns site identifiers to VPLS sites. In this example, you configure a routing instance called vpls instance and enable automatic site identifiers for VPLS.

NOTE: Site identifiers for VPLS sites can be different for different routing instances.

Configuration

Step-by-Step Procedure

The following example requires you to navigate various levels in the configuration hierarchy. For instructions on how to do that, see *Using the CLI Editor in Configuration Mode*.

To configure automatic site identifiers:

1. Configure the routing instance of type VPLS.

   ```
   [edit]
   user@host#set routing-instances vpls-instance
   ```

2. Enable automatic site identifiers.

   ```
   [edit routing-instances vpls-instance]
   user@host#set protocols vpls no-tunnel-services site site10 automatic-site-id
     collision-detect-time 10
   user@host#set protocols vpls no-tunnel-services site site10 automatic-site-id
     new-site-wait-time 20
   user@host#set protocols vpls no-tunnel-services site site10 automatic-site-id
     reclaim-wait-time minimum 5 maximum 20
   user@host#set protocols vpls no-tunnel-services site site10 automatic-site-id
     startup-wait-time 5
   ```

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

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

Verification

To verify the configuration is working properly, enter the **show vpls connections** command.

Related Documentation

- VPLS Configuration Overview on page 108
- VPLS Overview on page 103

CHAPTER 15

Configuring Routing and Signaling Protocols

Example: Configuring OSPF on the VPLS PE Router

Supported Platforms SRX1500, SRX300, SRX320, SRX340, SRX345, SRX550M, vSRX

This example shows how to configure OSPF on the VPLS PE router.

Requirements

Before you begin:

- Before you begin, see *Understanding Selective Stateless Packet-Based Services* .
- Configure the interfaces that will carry the VPLS traffic between the PE router and the CE devices. See “Example: Configuring Routing Interfaces on the VPLS PE Router” on page 113 and “Example: Configuring the Interface to the VPLS CE Device” on page 114.
- Create a VPLS routing instance on each PE router that is participating in the VPLS. See “Example: Configuring the VPLS Routing Instance” on page 124.

Overview

The PE routers exchange routing information using an IGP such as OSPF. In this example, you configure OSPF area 0.0.0.0 on the VPLS PE router and traffic engineering for OSPF.

Configuration

Step-by-Step Procedure

To configure OSPF on the VPLS PE router:

1. Configure the OSPF area on the VPLS PE router.

```
[edit]
user@host# set protocols ospf area 0.0.0.0 interface t1-1/0/1.0
user@host# set protocols ospf area 0.0.0.0 interface lo0.0
```

2. Configure traffic engineering for OSPF.

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

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

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

Verification

To verify the configuration is working properly, enter the **show protocols** command.

Related Documentation

- VPLS Configuration Overview on page 108
- VPLS Overview on page 103

Example: Configuring RSVP on the VPLS PE Router

Supported Platforms SRX1500, SRX300, SRX320, SRX340, SRX345, SRX550M, vSRX

This example shows how to configure RSVP on the VPLS PE router.

- Requirements on page 130
- Overview on page 131
- Configuration on page 131
- Verification on page 131

Requirements

Before you begin:

- Before you begin, see *Understanding Selective Stateless Packet-Based Services* .
- Configure the interfaces that will carry the VPLS traffic between the PE router and the CE devices. See“Example: Configuring Routing Interfaces on the VPLS PE Router” on page 113 and “Example: Configuring the Interface to the VPLS CE Device” on page 114.

- Create a VPLS routing instance on each PE router that is participating in the VPLS. See "Example: Configuring the VPLS Routing Instance" on page 124.
- Configure an IGP on the PE routers to exchange routing information. See "Example: Configuring OSPF on the VPLS PE Router" on page 129.

Overview

This example describes how to enable RSVP for all connections that participate in the LSP on the PE1 router.

Configuration

Step-by-Step Procedure

To configure RSVP on the VPLS PE router:

1. Configure the interface to the PE2 router for RSVP.

```
[edit ]
user@host# set protocols rsvp interface t1-1/0/1.0
```

2. Configure the loopback interface for RSVP.

```
[edit ]
user@host# set protocols rsvp interface lo0.0
```

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

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

Verification

To verify the configuration is working properly, enter the **show protocols** command.

Related Documentation

- VPLS Configuration Overview on page 108
- VPLS Overview on page 103

Example: Configuring MPLS on the VPLS PE Router

Supported Platforms SRX1500, SRX300, SRX320, SRX340, SRX345, SRX550M, vSRX

This example shows how to configure MPLS on the VPLS PE router.

- Requirements on page 132
- Overview on page 132
- Configuration on page 132
- Verification on page 133

Requirements

Before you begin:

- Before you begin, see *Understanding Selective Stateless Packet-Based Services* .
- Configure the interfaces that will carry the VPLS traffic between the PE router and the CE devices. See“Example: Configuring Routing Interfaces on the VPLS PE Router” on page 113 and “Example: Configuring the Interface to the VPLS CE Device” on page 114.
- Create a VPLS routing instance on each PE router that is participating in the VPLS. See “Example: Configuring the VPLS Routing Instance” on page 124.
- Configure an IGP on the PE routers to exchange routing information. See “Example: Configuring OSPF on the VPLS PE Router” on page 129.
- Configure RSVP-TE on the PE routers. See “Example: Configuring RSVP on the VPLS PE Router” on page 130.

Overview

This example shows you how to configure MPLS on the PE1 router to advertise the Layer 2 VPN interface that communicates with the PE2 router.

CAUTION: **MPLS is disabled by default on SRX Series devices. You must explicitly configure your router to allow MPLS traffic. However, when MPLS is enabled, all flow-based security features are deactivated and the router performs packet-based processing. Flow-based services such as security policies, zones, NAT, ALGs, chassis clustering, screens, firewall authentication, and IPsec VPNs are unavailable on the router.**

Configuration

Step-by-Step Procedure

To configure MPLS on the VPLS PE router:

1. Configure the interface to the PE2 router for MPLS.

   ```
   [edit ]
   user@host# set protocols mpls interface t1-1/0/1.0
   ```

2. Configure the loopback for MPLS.

   ```
   [edit ]
   user@host# set protocols mpls interface lo0.0
   ```

3. Configure the path to destination 10.255.7.164.

   ```
   [edit ]
   user@host# set protocols mpls label-switched-path chelsea-sagar to 10.255.7.164
   ```

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

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

Verification

To verify the configuration is working properly, enter the **show mpls** command.

Related Documentation

- VPLS Configuration Overview on page 108
- VPLS Overview on page 103

Example: Configuring LDP on the VPLS PE Router

Supported Platforms SRX1500, SRX300, SRX320, SRX340, SRX345, SRX550M, vSRX

This example shows how to configure LDP on the VPLS PE router.

- Requirements on page 133
- Overview on page 133
- Configuration on page 133
- Verification on page 134

Requirements

Before you begin:

- Before you begin, see *Understanding Selective Stateless Packet-Based Services* .
- Configure the interfaces that will carry the VPLS traffic between the PE router and the CE devices. See “Example: Configuring Routing Interfaces on the VPLS PE Router” on page 113 and “Example: Configuring the Interface to the VPLS CE Device” on page 114.
- Create a VPLS routing instance on each PE router that is participating in the VPLS. See “Example: Configuring the VPLS Routing Instance” on page 124.
- Configure an IGP on the PE routers to exchange routing information. See “Example: Configuring OSPF on the VPLS PE Router” on page 129.

Overview

This example describes how to enable LDP for all connections that participate in the LSP on the PE1 router.

Configuration

Step-by-Step Procedure

To configure LDP on the VPLS PE router:

1. Configure the interface to the PE2 router for LDP.

```
[edit ]
user@host# set protocols ldp interface ge-3/0/2
```

2. Configure the loopback interface for LDP.

```
[edit ]
user@host# set protocols ldp interface lo0
```

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

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

Verification

To verify the configuration is working properly, enter the **show protocols** command.

Related Documentation

- VPLS Configuration Overview on page 108
- VPLS Overview on page 103

Example: Configuring VPLS over GRE with IPsec VPNs

Supported Platforms SRX Series, vSRX

This example demonstrates a network scenario consisting of a central office and one branch office that will use VPLS, MPLS, GRE, and IPsec to create secure Ethernet connectivity over a Layer 3 network. This configuration can be expanded to add many other branch sites.

- Requirements on page 134
- Overview on page 134
- Configuration on page 139
- Verification on page 150

Requirements

Before you begin:

- Ensure that a layer 3 network is in place for all branch offices and that there is an ingress (head-end) device at the central office configured to terminate the VPNs from each branch office.
- Obtain IDP licenses for each SRX Series device. IDP is used to reassemble GRE packets that might become fragmented.

Overview

Junos OS can selectively choose whether traffic is processed by the flow engine or packet engine using the selective stateless packet-based feature. This feature allows you to combine flow and packet-based services in a single device. In this example, we describe a deployment scenario that uses this feature to deploy large-scale VPLS over GRE. This enables branch devices to securely transport Ethernet traffic over Layer 3 networks when used in conjunction with IPsec.

In this scenario you configure a central office ingress (head-end) using an SRX650 device and one branch office using an SRX240 device. This setup is accomplished by carrying MPLS pseudowires over GRE, which in turn, is encapsulated in IPsec in order to guarantee data integrity and confidentiality. By default, SRX Series devices use secure flow forwarding. Because VPLS services are provided in packet-mode only, the configuration requires the GRE tunnel to be terminated in a packet-mode routing instance (the default routing instance).

NOTE: You can also use an MX Series device as the ingress (head-end) device, which is mentioned later in this topic.

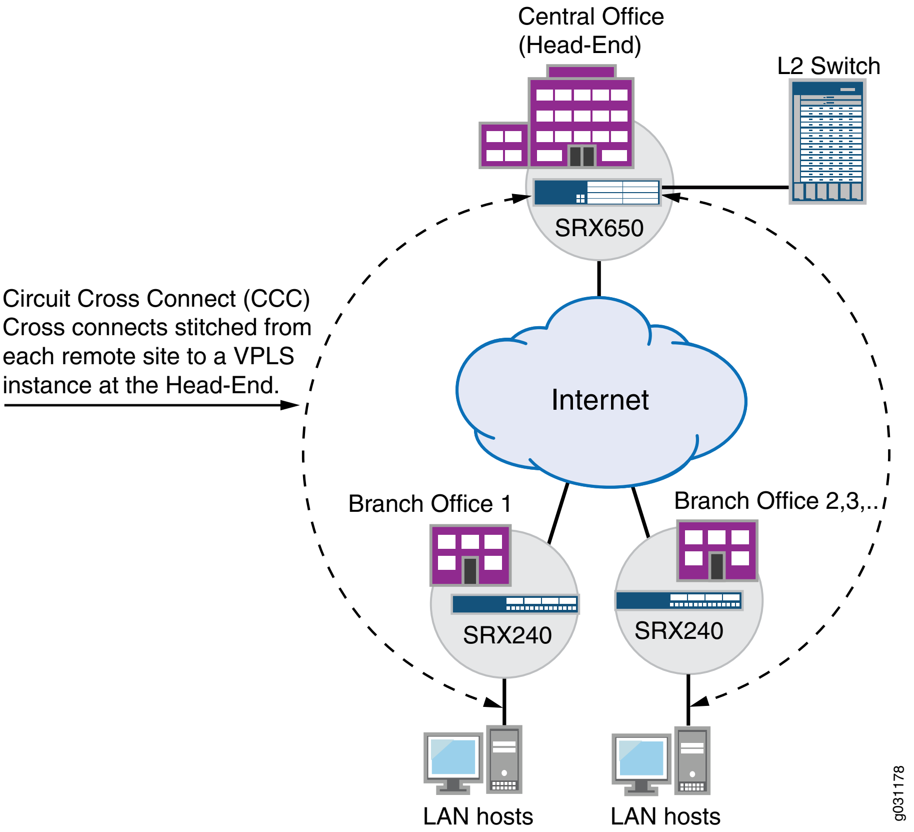
To better understand this configuration, we will discuss two scenarios. The first scenario uses pseudowires to allow the creation of point-to-point circuits between two endpoints carried over the MPLS network. If we leave the signaling protocols aside (that is, there are a few ways to provision the pseudowires), these connections are just point-to-point connections. Using this approach provides an end-to-end wire between sites. This is beneficial from a traffic processing point of view because the gateways do not need to do MAC address learning, they simply forward anything they receive to the pseudowire. Because of this, it may be difficult to deploy this setup when trying to provide connectivity to multiple branch offices.

The second scenario could use VPLS to provide a Layer 2 network abstraction. With VPLS, endpoints are expected to negotiate LSPs and pseudowires with every other endpoint (that is, they are fully meshed). When a node receives an Ethernet frame from one of its LAN interfaces the source MAC address is learned, if it’s not already known, and flooded using every pseudowire connecting to all other branch nodes. However, if the destination has been previously learned, then the frame is sent to the appropriate destination. When an Ethernet frame is received through one of the pseudowires (that is, from the MPLS network), source MAC address learning is performed. The next time a frame is sent to that MAC it does not need to be flooded and the frame is flooded to every single LAN interface in the node, but not over the pseudowires. In other words, the network acts as a distributed Layer 2 switch providing any-to-any Ethernet connectivity between the devices connected to the different nodes in the network.

While the advantages of this second scenario is evident (any-to-any connectivity, automated provisioning, and simple abstraction), it comes at the cost of complexity. Every PE node has to perform Layer 2 learning and flooding of traffic, which can cause problems when either multiple broadcast/multicast or frames to unknown MAC addresses are used. As an example, if you had a topology with a thousand branch offices, each office that receives a broadcast packet must replicate it 999 times, encapsulate each copy in GRE and IPsec and forward the resulting traffic. Additionally, because each node performs Layer 2 learning, there are limitations in the maximum number of MAC addresses that each node can learn, limiting the total number of nodes in the domain.

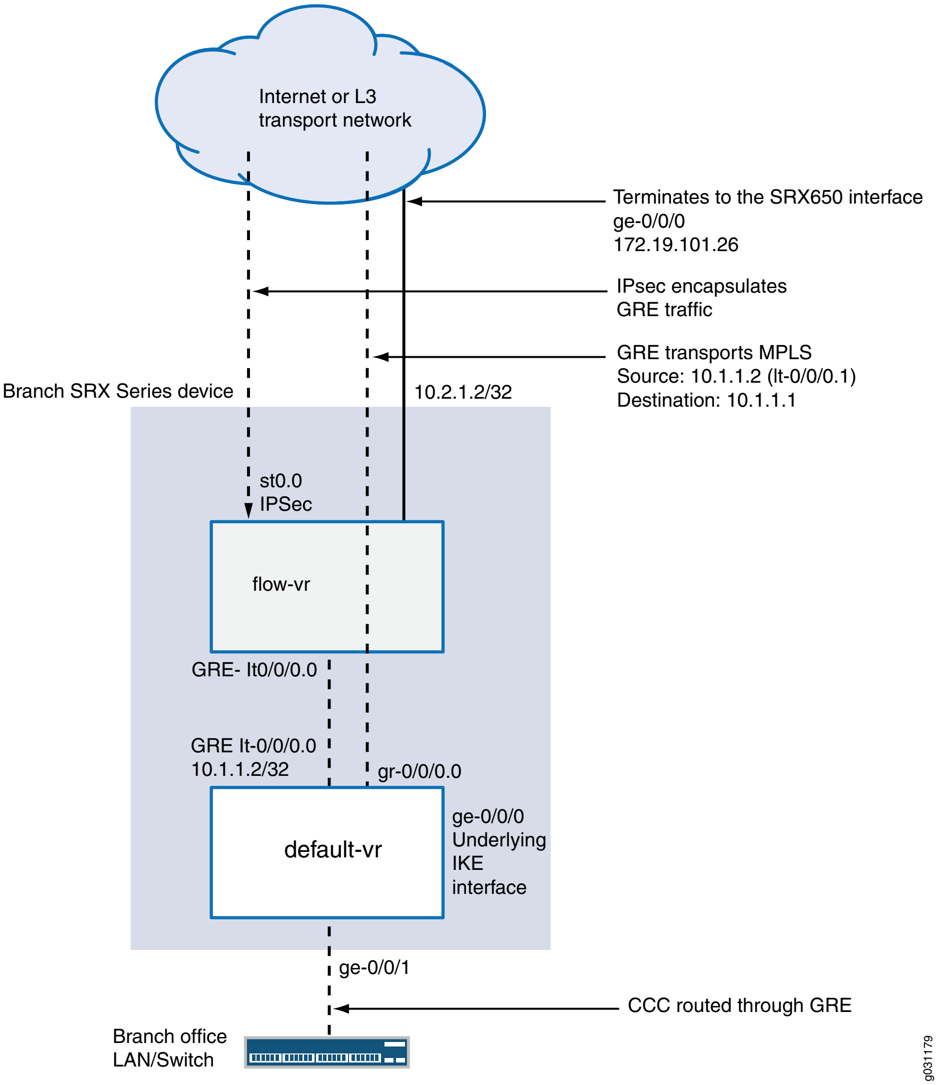
In this example, we use a hybrid approach to these two scenarios. We use a circuit cross connect (CCC) at each branch office stitched to a VPLS instance at central office (ingress). This solution makes sense if most of the traffic flows from the branch offices to central office, and the branch-to-branch office traffic is always forwarded through the hub. The use of CCCs at branch offices combined with VPLS stitching at the central office provides a scalable way to deploy large hub-and-spoke topologies where Ethernet must be transported over an IP network (with or without encryption). At the expense of configuration complexity, it is possible to use branch SRX Series devices to terminate such connections, providing a scalable and cost-effective way to deploy small-to-large networks where Ethernet traffic is carried transparently using lower cost IP connections. Figure 10 on page 136 shows this topology.

Figure 10: VPLS Deployment Scenario



In this deployment, VPLS services are provided only in packet mode and must be configured in the default routing instance. Unfortunately, IPsec is only provided in flow mode. Hence, a flow-mode routing-instance is used that provides both GRE reassembly and IPsec termination. While the GRE termination is done in the default routing instance, a flow-mode routing instance is connected between the default routing instance and the Internet (or whatever Layer 3 network is used as a transport), and it terminates the IPsec tunnel towards the ingress device. Because it is likely that a single public IP address is available, the Internet-facing Interface is connected to the default routing instance and is used to terminate IKE; however, the tunnel interface (st0) is bound to the flow-mode routing instance. See Figure 11 on page 137.

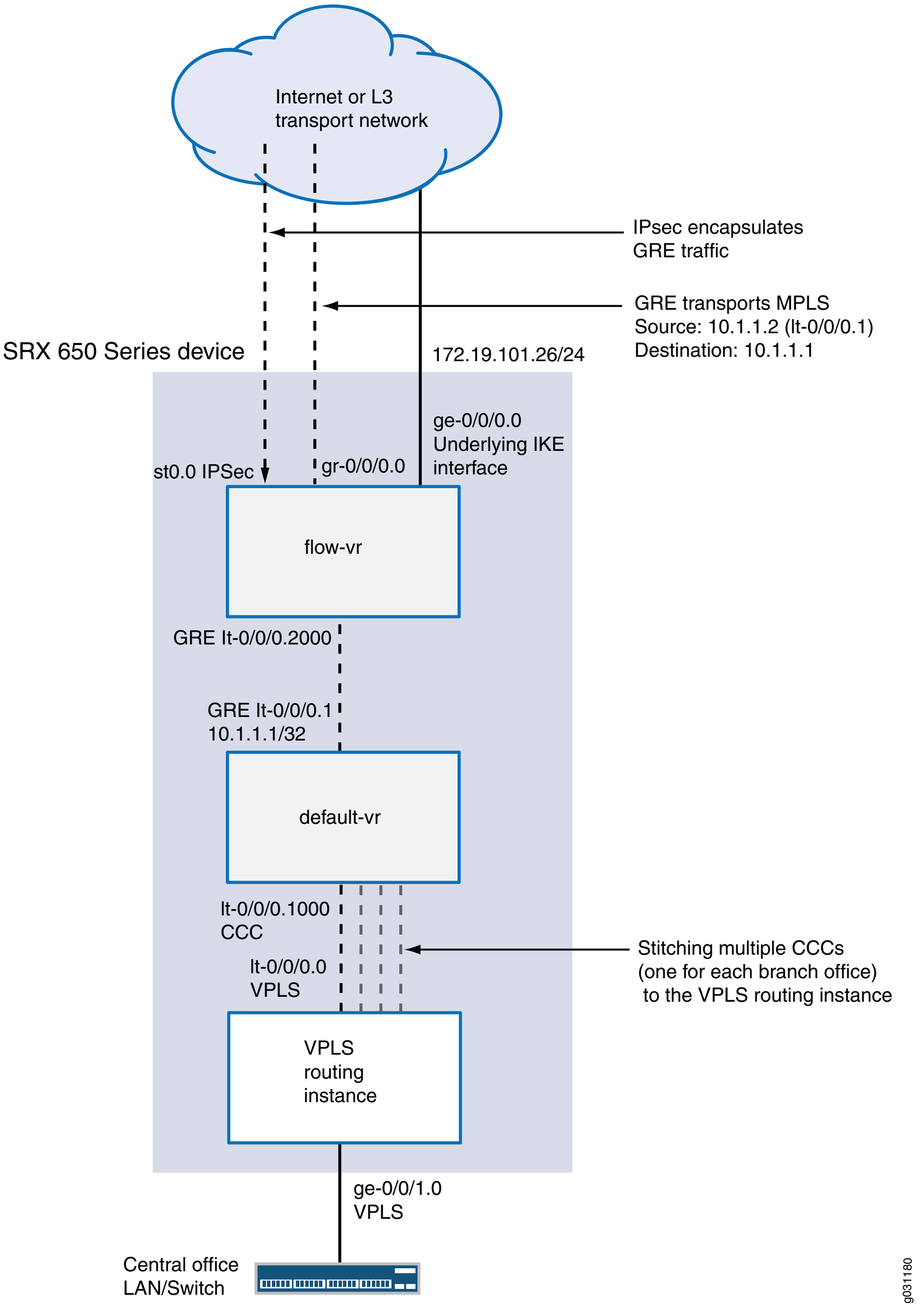
Figure 11: Branch Office Circuit Cross Connect Termination



When configuring the central office SRX650, the first thing you do is terminate the IPsec tunnels, GRE, and CCC connections. Because a branch SRX Series device is used as the ingress (head-end), the configuration to terminate the CCC circuits is identical to the one used at each branch office, with the exception that instead of one tunnel, multiple tunnels (and pseudowires) are terminated.

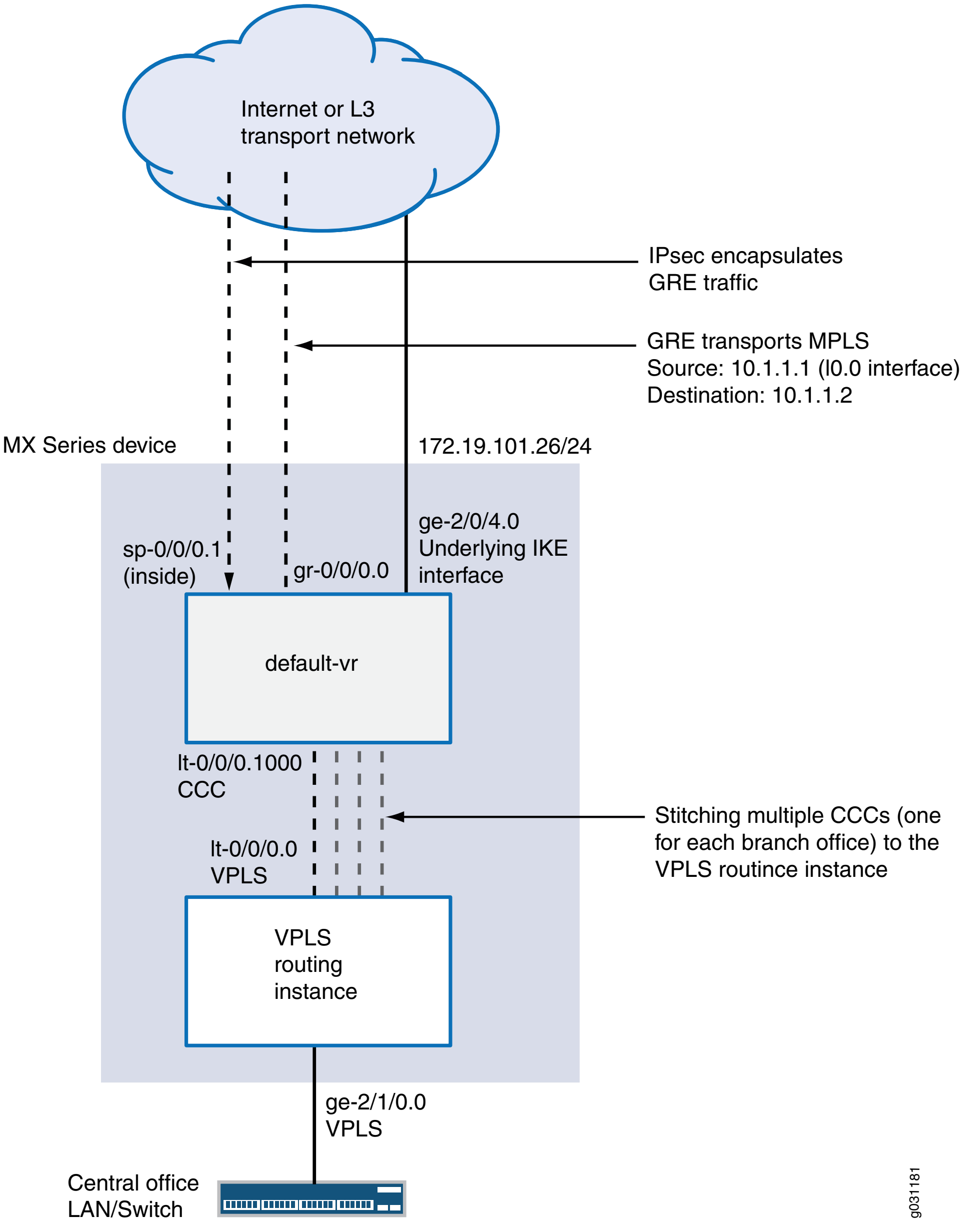
The pseudowires are stitched to a VPLS routing instance using logical tunnel (lt) interfaces. It is possible to use an lt interface unit to terminate a CCC connection and connect this unit to a different unit that is part of a VPLS routing instance. The overall result is as if the pseudowires were terminated directly in the VPLS routing instance. Figure 12 on page 138 illustrates this configuration.

Figure 12: Central Office Ingress (Head-End) Configuration with an SRX Series Device



You can also use an MX Series device as the central office ingress (head-end) to terminate all branch office connections. The differences in the configuration are due to the way IPsec is configured and the fact that on MX Series devices IDP is not required to reassemble the GRE packets; MX Series devices natively support GRE reassembly. With this configuration, you still use lt interfaces to stitch the CCCs between the remote branch offices and the VPLS routing instance as shown in Figure 13 on page 139.

Figure 13: Central Office Ingress (Head-End) Configuration with an MX Series Device



Configuration

In this example, we use SRX Series devices and the branch and ingress (head-end) sites will typically be connected to the Internet by Frame-Relay/T1-E1/xDSL/T3/E3 or even Ethernet. A provider MPLS network is not required.

- Configuring the SRX240 Device at the Branch Office on page 140
- Configuring the SRX650 Device at the Central Office on page 144

Configuring the SRX240 Device at the Branch Office

CLI Quick Configuration

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

```
set interfaces gr-0/0/0 description "GRE tunnel to SRX650”
set interfaces gr-0/0/0 unit 0 clear-dont-fragment-bit
set interfaces gr-0/0/0 unit 0 tunnel source 10.1.1.2
set interfaces gr-0/0/0 unit 0 tunnel destination 10.1.1.1
set interfaces gr-0/0/0 unit 0 tunnel allow-fragmentation
set interfaces gr-0/0/0 unit 0 family inet mtu 2000
set interfaces gr-0/0/0 unit 0 family inet filter input inet-packet-mode
set interfaces gr-0/0/0 unit 0 family mpls mtu 1900
set interfaces gr-0/0/0 unit 0 family mpls filter input mpls-packet-mode
set interfaces lt-0/0/0 unit 0 encapsulation frame-relay
set interfaces lt-0/0/0 unit 0 dlci 16
set interfaces lt-0/0/0 unit 0 peer-unit 1
set interfaces lt-0/0/0 unit 0 family inet
set interfaces lt-0/0/0 unit 0 description "Flow-vr Instance"
set interfaces lt-0/0/0 unit 1 encapsulation frame-relay
set interfaces lt-0/0/0 unit 1 dlci 16
set interfaces lt-0/0/0 unit 1 peer-unit 0
set interfaces lt-0/0/0 unit 1 family inet filter input inet-packet-mode
set interfaces lt-0/0/0 unit 1 family inet address 10.1.1.2/32
set interfaces ge-0/0/1 encapsulation ethernet-ccc
set interfaces ge-0/0/1 unit 0 description "CCC Interface to customer LAN"
set interfaces ge-0/0/1 unit 0 family ccc filter input ccc-packet-mode
set interfaces ge-0/0/0 unit 0 family inet address 172.19.101.45/24
set interfaces lo0 unit 0 family inet address 10.2.1.2/32
set interfaces st0 unit 0 family inet
set routing-options static route 0.0.0.0/0 next-hop 172.19.101.1
set routing-options static route 10.1.1.1/32 next-hop lt-0/0/0.1
set routing-options static route 10.2.1.1/32 next-hop gr-0/0/0.0
set routing-options router-id 10.2.1.2
set protocols mpls interface gr-0/0/0.0
set protocols ldp interface gr-0/0/0.0
set protocols ldp interface lo0.0
set protocols l2circuit neighbor 10.2.1.1 interface ge-0/0/1.0 virtual-circuit-id 1
set security ike policy SRX650 mode main
set security ike policy SRX650 proposal-set standard
set security ike policy SRX650 pre-shared-key ascii-text "$ABC123"
set security ike gateway SRX650 ike-policy SRX650
set security ike gateway SRX650 address 172.19.101.26
set security ike gateway SRX650 external-interface ge-0/0/0.0
set security ipsec policy SRX650 proposal-set standard
set security ipsec vpn SRX650 bind-interface st0.0
set security ipsec vpn SRX650 ike gateway SRX650
set security ipsec vpn SRX650 ike ipsec-policy SRX650
set security ipsec vpn SRX650 establish-tunnels immediately
set security zones security-zone untrust host-inbound-traffic system-services all
set security zones security-zone untrust host-inbound-traffic protocols all
set security zones security-zone untrust interfaces gr-0/0/0.0
set security zones security-zone untrust interfaces lo0.0
set security zones security-zone untrust interfaces lt-0/0/0.1
set security zones security-zone untrust interfaces ge-0/0/0.0
set security zones security-zone vpn host-inbound-traffic system-services all
set security zones security-zone vpn host-inbound-traffic protocols all
set security zones security-zone vpn interfaces st0.0
set security zones security-zone trust-flow host-inbound-traffic system-services all
set security zones security-zone trust-flow host-inbound-traffic protocols all
set security zones security-zone trust-flow interfaces lt-0/0/0.0
set security policies from-zone trust-flow to-zone vpn policy gre match source-address
  any
set security policies from-zone trust-flow to-zone vpn policy gre match destination-address
  any
set security policies from-zone trust-flow to-zone vpn policy gre match application
  junos-gre
set security policies from-zone trust-flow to-zone vpn policy gre then permit
  application-services idp
set security idp idp-policy gre-reassembly rulebase-ips rule match-all match application
  junos-gre
set security idp idp-policy gre-reassembly rulebase-ips rule match-all then action
  ignore-connection
set security idp active-policy gre-reassembly
set firewall family inet filter inet-packet-mode term control-traffic from protocol tcp
set firewall family inet filter inet-packet-mode term control-traffic from port 22
set firewall family inet filter inet-packet-mode term control-traffic from port 80
set firewall family inet filter inet-packet-mode term control-traffic from port 8080
set firewall family inet filter inet-packet-mode term control-traffic then accept
set firewall family inet filter inet-packet-mode term packet-mode then packet-mode
set firewall family inet filter inet-packet-mode term packet-mode then accept
set firewall family mpls filter mpls-packet-mode term packet-mode then packet-mode
set firewall family mpls filter mpls-packet-mode term packet-mode then accept
set firewall family ccc filter ccc-packet-mode term all then packet-mode
set firewall family ccc filter ccc-packet-mode term all then accept
set routing-instances flow-vr instance-type virtual-router
set routing-instances flow-vr interface lt-0/0/0.0
set routing-instances flow-vr interface st0.0
set routing-instances flow-vr routing-options static route 10.1.1.1/32 next-hop st0.0
set routing-instances flow-vr routing-options static route 10.1.1.2/32 next-hop lt-0/0/0.0
```

Step-by-Step Procedure

The following example requires you to navigate various levels in the configuration hierarchy. For instructions on how to do that, see *Using the CLI Editor in Configuration Mode*.

To configure the SRX240 at the branch office:

1. Configure a GRE tunnel to the central office.

```
[edit interfaces]
user@host# set gr-0/0/0 description "GRE tunnel to SRX650”
user@host# set gr-0/0/0 unit 0 clear-dont-fragment-bit
user@host# set gr-0/0/0 unit 0 tunnel source 10.1.1.2
user@host# set gr-0/0/0 unit 0 tunnel destination 10.1.1.1
user@host# set gr-0/0/0 unit 0 tunnel allow-fragmentation
user@host# set gr-0/0/0 unit 0 family inet mtu 2000
user@host# set gr-0/0/0 unit 0 family inet filter input inet-packet-mode
user@host# set gr-0/0/0 unit 0 family mpls mtu 1900
user@host# set gr-0/0/0 unit 0 family mpls filter input mpls-packet-mode
```

2. Create a logical interface that connects to the default routing instance.

   ```
   [edit interfaces]
   user@host# set lt-0/0/0 unit 0 encapsulation frame-relay
   user@host# set lt-0/0/0 unit 0 dlci 16
   user@host# set lt-0/0/0 unit 0 peer-unit 1
   user@host# set lt-0/0/0 unit 0 family inet
   user@host# set lt-0/0/0 unit 0 description "Flow-vr Instance"
   ```

3. Connect the logical tunnel interface to the flow mode virtual router.

   ```
   [edit interfaces]
   user@host# set lt-0/0/0 unit 1 encapsulation frame-relay
   user@host# set lt-0/0/0 unit 1 dlci 16
   user@host# set lt-0/0/0 unit 1 peer-unit 0
   user@host# set lt-0/0/0 unit 1 family inet filter input inet-packet-mode
   user@host# set lt-0/0/0 unit 1 family inet address 10.1.1.2/32
   ```

4. Connect the CCC interface to the branch LAN.

   ```
   [edit interfaces]
   user@host# set ge-0/0/1 encapsulation ethernet-ccc
   user@host# set ge-0/0/1 unit 0 description "CCC Interface to customer LAN"
   user@host# set ge-0/0/1 unit 0 family ccc filter input ccc-packet-mode
   ```

5. Configure the interface bound to the default virtual router.

   ```
   [edit interfaces]
   user@host# set ge-0/0/0 unit 0 family inet address 172.19.101.45/24
   ```

6. Set the loopback interface to terminate the CCC connection.

   ```
   [edit interfaces]
   user@host# set lo0 unit 0 family inet address 10.2.1.2/32
   ```

7. Bind the IPsec tunnel interface to the flow-mode virtual router.

   ```
   [edit interfaces]
   user@host# set st0 unit 0 family inet
   ```

8. Set a static route address, which will be the default gateway to the Internet.

   ```
   [edit routing-options]
   user@host# set static route 0.0.0.0/0 next-hop 172.19.101.1
   ```

9. Set a static route for the remote GRE tunnel endpoint.

   ```
   [edit routing-options]
   user@host# set static route 10.1.1.1/32 next-hop lt-0/0/0.1
   ```

10. Set a static route for the loopback interface of the SRX650 ingress (head-end) device.

    ```
    [edit routing-options]
    user@host# set static route 10.2.1.1/32 next-hop gr-0/0/0.0
    ```

11. Configure MPLS and the CCC using LDP as the label protocol.

    ```
    [edit]
    user@host# set routing-options router-id 10.2.1.2
    user@host# set protocols mpls interface gr-0/0/0.0
    user@host# set protocols ldp interface gr-0/0/0.0
    user@host# set protocols ldp interface lo0.0
    ```

```
user@host# set protocols l2circuit neighbor 10.2.1.1 interface ge-0/0/1.0
  virtual-circuit-id 1
```

12. Configure the IPsec tunnel.

NOTE: The underlying IKE interface is not in the same routing instance as the tunnel interface.

```
[edit security]
user@host# set ike policy SRX650 mode main
user@host# set ike policy SRX650 proposal-set standard
user@host# set ike policy SRX650 pre-shared-key ascii-text "$ABC123"
user@host# set ike gateway SRX650 ike-policy SRX650
user@host# set ike gateway SRX650 address 172.19.101.26
user@host# set ike gateway SRX650 external-interface ge-0/0/0.0
user@host# set ipsec policy SRX650 proposal-set standard
user@host# set ipsec vpn SRX650 bind-interface st0.0
user@host# set ipsec vpn SRX650 ike gateway SRX650
user@host# set ipsec vpn SRX650 ike ipsec-policy SRX650
user@host# set ipsec vpn SRX650 establish-tunnels immediately
```

13. Configure security zones.

NOTE: In a production environment, host-inbound traffic should be restricted to only allow the necessary protocols and services.

```
[edit security]
user@host# set zones security-zone untrust host-inbound-traffic system-services
  all
user@host# set zones security-zone untrust host-inbound-traffic protocols all
user@host# set zones security-zone untrust interfaces gr-0/0/0.0
user@host# set zones security-zone untrust interfaces lo0.0
user@host# set zones security-zone untrust interfaces lt-0/0/0.1
user@host# set zones security-zone untrust interfaces ge-0/0/0.0
user@host# set zones security-zone vpn host-inbound-traffic system-services all
user@host# set zones security-zone vpn host-inbound-traffic protocols all
user@host# set zones security-zone vpn interfaces st0.0
user@host# set zones security-zone trust-flow host-inbound-traffic system-services
  all
user@host# set zones security-zone trust-flow host-inbound-traffic protocols all
user@host# set zones security-zone trust-flow interfaces lt-0/0/0.0
```

14. Configure IDP.

```
[edit security]
user@host# set policies from-zone trust-flow to-zone vpn policy gre match
  source-address any
user@host# set policies from-zone trust-flow to-zone vpn policy gre match
  destination-address any
user@host# set policies from-zone trust-flow to-zone vpn policy gre match
  application junos-gre
```

```
user@host# set policies from-zone trust-flow to-zone vpn policy gre then permit
  application-services idp
user@host# set idp idp-policy gre-reassembly rulebase-ips rule match-all match
  application junos-gre
user@host# set idp idp-policy gre-reassembly rulebase-ips rule match-all then
  action ignore-connection
user@host# set idp active-policy gre-reassembly
```

15. Configure packet-mode filters.

```
[edit firewall]
user@host# set family inet filter inet-packet-mode term control-traffic from protocol
  tcp
user@host# set family inet filter inet-packet-mode term control-traffic from port
  22
user@host# set family inet filter inet-packet-mode term control-traffic from port
  80
user@host# set family inet filter inet-packet-mode term control-traffic from port
  8080
user@host# set family inet filter inet-packet-mode term control-traffic then accept
user@host# set family inet filter inet-packet-mode term packet-mode then
  packet-mode
user@host# set family inet filter inet-packet-mode term packet-mode then accept
user@host# set family mpls filter mpls-packet-mode term packet-mode then
  packet-mode
user@host# set family mpls filter mpls-packet-mode term packet-mode then accept
user@host# set family ccc filter ccc-packet-mode term all then packet-mode
user@host# set family ccc filter ccc-packet-mode term all then accept
```

16. Configure the flow-mode virtual router.

```
[edit routing-instances]]
user@host# set flow-vr instance-type virtual-router
user@host# set flow-vr interface lt-0/0/0.0
user@host# set flow-vr interface st0.0
user@host# set flow-vr routing-options static route 10.1.1.1/32 next-hop st0.0
user@host# set flow-vr routing-options static route 10.1.1.2/32 next-hop lt-0/0/0.0
```

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

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

Configuring the SRX650 Device at the Central Office

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

```
set interfaces ge-0/0/0 unit 0 family inet address 172.19.101.26/24
set interfaces gr-0/0/0 unit 0 clear-dont-fragment-bit
set interfaces gr-0/0/0 unit 0 tunnel source 10.1.1.1
set interfaces gr-0/0/0 unit 0 tunnel destination 10.1.1.2
set interfaces gr-0/0/0 unit 0 tunnel allow-fragmentation
set interfaces gr-0/0/0 unit 0 family inet mtu 1500
set interfaces gr-0/0/0 unit 0 family inet filter input inet-packet-mode
set interfaces gr-0/0/0 unit 0 family mpls filter input mpls-packet-mode
set interfaces lt-0/0/0 unit 0 description "VPLS hub port - Interconnect for CCC to
  SRX240"
set interfaces lt-0/0/0 unit 0 encapsulation ethernet-vpls
set interfaces lt-0/0/0 unit 0 peer-unit 1000
set interfaces lt-0/0/0 unit 1000 description "Stitch to VPLS for CCC to SRX240"
set interfaces lt-0/0/0 unit 1000 encapsulation ethernet-ccc
set interfaces lt-0/0/0 unit 1000 peer-unit 0
set interfaces lt-0/0/0 unit 1000 family ccc filter input ccc-packet-mode
set interfaces lt-0/0/0 unit 2000 encapsulation frame-relay
set interfaces lt-0/0/0 unit 2000 dlci 1
set interfaces lt-0/0/0 unit 2000 peer-unit 2001
set interfaces lt-0/0/0 unit 2000 family inet
set interfaces lt-0/0/0 unit 2001 encapsulation frame-relay
set interfaces lt-0/0/0 unit 2001 dlci 1
set interfaces lt-0/0/0 unit 2001 peer-unit 2000
set interfaces lt-0/0/0 unit 2001 family inet filter input inet-packet-mode
set interfaces lt-0/0/0 unit 2001 family inet address 10.1.1.1/32
set interfaces ge-0/0/1 unit 0
set interfaces ge-0/0/1 encapsulation ethernet-vpls
set interfaces lo0 unit 0 family inet address 10.2.1.1/32
set interfaces st0 unit 0 family inet
set routing-options static route 10.1.1.2/32 next-hop lt-0/0/0.2001
set routing-options static route 10.2.1.2/32 next-hop gr-0/0/0.0
set protocols mpls interface gr-0/0/0.0
set protocols ldp interface gr-0/0/0.0
set protocols ldp interface lo0.0
set protocols l2circuit neighbor 10.2.1.2 interface lt-0/0/0.1000 virtual-circuit-id 1
set security ike policy SRX mode main
set security ike policy SRX proposal-set standard
set security ike policy SRX pre-shared-key ascii-text "$ABC123"
set security ike gateway SRX240-1 ike-policy SRX
set security ike gateway SRX240-1 address 172.19.101.45
set security ike gateway SRX240-1 external-interface ge-0/0/0.0
set security ipsec policy SRX proposal-set standard
set security ipsec vpn SRX240-1 bind-interface st0.0
set security ipsec vpn SRX240-1 ike gateway SRX240-1
set security ipsec vpn SRX240-1 ike ipsec-policy SRX
set security ipsec vpn SRX240-1 establish-tunnels immediately
set security zones security-zone untrust host-inbound-traffic system-services all
set security zones security-zone untrust host-inbound-traffic protocols all
set security zones security-zone untrust interfaces lo0.0
set security zones security-zone untrust interfaces lt-0/0/0.2001
set security zones security-zone untrust interfaces gr-0/0/0.0
set security zones security-zone untrust interfaces ge-0/0/0.0
set security zones security-zone vpn host-inbound-traffic system-services all
set security zones security-zone vpn host-inbound-traffic protocols all
set security zones security-zone vpn interfaces st0.0
set security zones security-zone trust-flow host-inbound-traffic system-services all
set security zones security-zone trust-flow host-inbound-traffic protocols all
set security zones security-zone trust-flow interfaces lt-0/0/0.2000
set security policies from-zone trust-flow to-zone vpn policy gre match source-address
  any
```

```
set security policies from-zone trust-flow to-zone vpn policy gre match destination-address
  any
set security policies from-zone trust-flow to-zone vpn policy gre match application
  junos-gre
set security policies from-zone trust-flow to-zone vpn policy gre then permit
  application-services idp
set security policies from-zone vpn to-zone trust-flow policy gre match source-address
  any
set security policies from-zone vpn to-zone trust-flow policy gre match destination-address
  any
set security policies from-zone vpn to-zone trust-flow policy gre match application
  junos-gre
set security policies from-zone vpn to-zone trust-flow policy gre then permit
  application-services idp
set security idp idp-policy gre-reassembly rulebase-ips rule match-gre match application
  junos-gre
set security idp idp-policy gre-reassembly rulebase-ips rule match-gre then action
  ignore-connection
set security idp active-policy gre-reassembly
set firewall family inet filter inet-packet-mode term control-traffic from protocol tcp
set firewall family inet filter inet-packet-mode term control-traffic from port 22
set firewall family inet filter inet-packet-mode term control-traffic from port 80
set firewall family inet filter inet-packet-mode term control-traffic from port 8080
set firewall family inet filter inet-packet-mode term control-traffic then accept
set firewall family inet filter inet-packet-mode term packet-mode then packet-mode
set firewall family inet filter inet-packet-mode term packet-mode then accept
set firewall family mpls filter mpls-packet-mode term packet-mode then packet-mode
set firewall family mpls filter mpls-packet-mode term packet-mode then accept
set firewall family ccc filter ccc-packet-mode term all then packet-mode
set firewall family ccc filter ccc-packet-mode term all then accept
set routing-instances flow-vr instance-type virtual-router
set routing-instances flow-vr interface lt-0/0/0.2000
set routing-instances flow-vr interface st0.0
set routing-instances flow-vr routing-options static route 10.1.1.1/32 next-hop
  lt-0/0/0.2000
set routing-instances flow-vr routing-options static route 10.1.1.2/32 next-hop st0.0
set routing-instances vpls-hub instance-type vpls
set routing-instances vpls-hub interface lt-0/0/0.0
set routing-instances vpls-hub interface ge-0/0/1.0
```

Step-by-Step Procedure

The following example requires you to navigate various levels in the configuration hierarchy. For instructions on how to do that, see *Using the CLI Editor in Configuration Mode*.

To configure the ingress (head-end) SRX650 device at the central office:

1. Configure the interface bound to the default virtual router.

   ```
   [edit interfaces]
   user@host# set ge-0/0/0 unit 0 family inet address 172.19.101.26/24
   ```

2. Create the GRE tunnel from the SRX650 to the SRX240 device.

NOTE: As the network expands to include multiple branch offices, you will need to add a similar GRE tunnel configuration on the SRX650 device (head-end) along with a corresponding IPsec configuration to connect to each additional branch device (SRX240).

```
[edit interfaces]
user@host# set gr-0/0/0 unit 0 clear-dont-fragment-bit
user@host# set gr-0/0/0 unit 0 tunnel source 10.1.1.1
user@host# set gr-0/0/0 unit 0 tunnel destination 10.1.1.2
user@host# set gr-0/0/0 unit 0 tunnel allow-fragmentation
user@host# set gr-0/0/0 unit 0 family inet mtu 1500
user@host# set gr-0/0/0 unit 0 family inet filter input inet-packet-mode
user@host# set gr-0/0/0 unit 0 family mpls filter input mpls-packet-mode
```

3. Configure a logical tunnel interface to stitch the CCC connection to the VPLS instance.

```
[edit interfaces]
user@host# set lt-0/0/0 unit 0 description "VPLS hub port - Interconnect for CCC
  to SRX240"
user@host# set lt-0/0/0 unit 0 encapsulation ethernet-vpls
user@host# set lt-0/0/0 unit 0 peer-unit 1000
```

4. Set unit 1000 to terminate the CCC connection.

```
[edit interfaces]
user@host# set lt-0/0/0 unit 1000 description "Stitch to VPLS for CCC to SRX240"
user@host# set lt-0/0/0 unit 1000 encapsulation ethernet-ccc
user@host# set lt-0/0/0 unit 1000 peer-unit 0
user@host# set lt-0/0/0 unit 1000 family ccc filter input ccc-packet-mode
```

5. Configure the logical tunnel interface.

```
[edit interfaces]
user@host# set lt-0/0/0 unit 2000 encapsulation frame-relay
user@host# set lt-0/0/0 unit 2000 dlci 1
user@host# set lt-0/0/0 unit 2000 peer-unit 2001
user@host# set lt-0/0/0 unit 2000 family inet
```

6. Bind the logical tunnel interface to the default virtual router.

```
[edit interfaces]
user@host# set lt-0/0/0 unit 2001 encapsulation frame-relay
user@host# set lt-0/0/0 unit 2001 dlci 1
user@host# set lt-0/0/0 unit 2001 peer-unit 2000
user@host# set lt-0/0/0 unit 2001 family inet filter input inet-packet-mode
user@host# set lt-0/0/0 unit 2001 family inet address 10.1.1.1/32
```

7. Set the interface to the central office LAN network.

```
[edit interfaces]
user@host# set ge-0/0/1 unit 0
user@host# set ge-0/0/1 encapsulation ethernet-vpls
```

8. Set the loopback interface to terminate the CCC connections to each branch device.

```
[edit interfaces]
```

```
user@host# set lo0 unit 0 family inet address 10.2.1.1/32
```

9. Bind the IPsec interface to the flow-mode virtual router.

```
[edit interfaces]
user@host# set st0 unit 0 family inet
```

10. Set a static route for the remote GRE tunnel endpoint.

```
[edit routing-options]
user@host# set static route 10.1.1.2/32 next-hop lt-0/0/0.2001
```

11. Set a static route for the loopback interface of the branch device.

```
[edit]
user@host# set routing-options static route 10.2.1.2/32 next-hop gr-0/0/0.0
```

12. Configure MPLS and CCC using LDP as the label protocol.

```
[edit protocols]
user@host# set mpls interface gr-0/0/0.0
user@host# set ldp interface gr-0/0/0.0
user@host# set ldp interface lo0.0
user@host# set l2circuit neighbor 10.2.1.2 interface lt-0/0/0.1000 virtual-circuit-id 1
```

13. Configure the IPsec tunnel.

NOTE: The underlying IKE interface is not in the same routing instance as the tunnel interface.

```
[edit security]
user@host# set ike policy SRX mode main
user@host# set ike policy SRX proposal-set standard
user@host# set ike policy SRX pre-shared-key ascii-text "$ABC123"
user@host# set ike gateway SRX240-1 ike-policy SRX
user@host# set ike gateway SRX240-1 address 172.19.101.45
user@host# set ike gateway SRX240-1 external-interface ge-0/0/0.0
user@host# set ipsec policy SRX proposal-set standard
user@host# set ipsec vpn SRX240-1 bind-interface st0.0
user@host# set ipsec vpn SRX240-1 ike gateway SRX240-1
user@host# set ipsec vpn SRX240-1 ike ipsec-policy SRX
user@host# set ipsec vpn SRX240-1 establish-tunnels immediately
```

14. Configure security zones.

NOTE: In a production environment, restrict host-inbound traffic to only the necessary protocols and services.

```
[edit security]
user@host# set zones security-zone untrust host-inbound-traffic system-services all
user@host# set zones security-zone untrust host-inbound-traffic protocols all
user@host# set zones security-zone untrust interfaces lo0.0
```

```
user@host# set zones security-zone untrust interfaces lt-0/0/0.2001
user@host# set zones security-zone untrust interfaces gr-0/0/0.0
user@host# set zones security-zone untrust interfaces ge-0/0/0.0
user@host# set zones security-zone vpn host-inbound-traffic system-services all
user@host# set zones security-zone vpn host-inbound-traffic protocols all
user@host# set zones security-zone vpn interfaces st0.0
user@host# set zones security-zone trust-flow host-inbound-traffic system-services
  all
user@host# set zones security-zone trust-flow host-inbound-traffic protocols all
user@host# set zones security-zone trust-flow interfaces lt-0/0/0.2000
```

15. Configure IDP.

```
[edit security]
user@host# set policies from-zone trust-flow to-zone vpn policy GRE match
  source-address any
user@host# set policies from-zone trust-flow to-zone vpn policy GRE match
  destination-address any
user@host# set policies from-zone trust-flow to-zone vpn policy GRE match
  application junos-gre
user@host# set policies from-zone trust-flow to-zone vpn policy GRE then permit
  application-services idp
user@host# set policies from-zone vpn to-zone trust-flow policy GRE match
  source-address any
user@host# set policies from-zone vpn to-zone trust-flow policy GRE match
  destination-address any
user@host# set policies from-zone vpn to-zone trust-flow policy GRE match
  application junos-gre
user@host# set policies from-zone vpn to-zone trust-flow policy GRE then permit
  application-services idp
user@host# set idp idp-policy gre-reassembly rulebase-ips rule match-gre match
  application junos-gre
user@host# set idp idp-policy gre-reassembly rulebase-ips rule match-gre then
  action ignore-connection
user@host# set idp active-policy gre-reassembly
```

16. Configure packet-mode filters.

```
[edit firewall]
user@host# set family inet filter inet-packet-mode term control-traffic from protocol
  tcp
user@host# set family inet filter inet-packet-mode term control-traffic from port
  22
user@host# set family inet filter inet-packet-mode term control-traffic from port
  80
user@host# set family inet filter inet-packet-mode term control-traffic from port
  8080
user@host# set family inet filter inet-packet-mode term control-traffic then accept
user@host# set family inet filter inet-packet-mode term packet-mode then
  packet-mode
user@host# set family inet filter inet-packet-mode term packet-mode then accept
user@host# set family mpls filter mpls-packet-mode term packet-mode then
  packet-mode
user@host# set family mpls filter mpls-packet-mode term packet-mode then accept
user@host# set family ccc filter ccc-packet-mode term all then packet-mode
user@host#set family ccc filter ccc-packet-mode term all then accept
```

17. Configure the flow-mode virtual router.

```
[edit routing-instances]
user@host# set flow-vr instance-type virtual-router
user@host# set flow-vr interface lt-0/0/0.2000
user@host# set flow-vr interface st0.0
user@host# set flow-vr routing-options static route 10.1.1.1/32 next-hop
  lt-0/0/0.2000
user@host# set flow-vr routing-options static route 10.1.1.2/32 next-hop st0.0
```

18. Configure the VPLS instance.

```
[edit routing-instances]
user@host# set vpls-hub instance-type vpls
user@host# set vpls-hub interface lt-0/0/0.0
user@host# set vpls-hub interface ge-0/0/1.0
```

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

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

Verification

Confirm that the configuration is working properly.

- Verifying Interfaces on page 150
- Verifying an IPsec tunnel on page 150
- Verifying GRE on page 150
- Verifying the CCC/L2 circuit. on page 151
- Verifying that LDP sessions are working. on page 151

Verifying Interfaces

Purpose Verify that the interfaces are configured properly on each device in the VPLS network.

Action From configuration mode, enter **show interfaces** and verify that the IP addressing is correct for each interface, including logical tunnel (lt), loopback (lo), GRE (gr), IPsec tunnel st0, and GE interfaces.

Verifying an IPsec tunnel

Purpose Verify that an IPsec tunnel is working.

Action From operational mode, enter the **show security ipsec security associations** and the **show security ipsec statistics** command.

Verifying GRE

Purpose Verify that GRE is working.

Action From operational mode, enter the **show security flow session protocol gre** command. You can also do a ping between loopback addresses.

Verifying the CCC/L2 circuit.

Purpose Verify that the CCC/L2 circuit is working.

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

Verifying that LDP sessions are working.

Purpose Verify that LDP sessions are being created between devices.

Action From operational mode, enter the **show interfaces gr-0/0/0 detail** command.

Related Documentation

- VPLS Overview on page 103
- Understanding VPLS Interfaces on page 111
- *Understanding Selective Stateless Packet-Based Services*
- MPLS Overview on page 3

Example: Configuring VPLS with BGP Signaling

Supported Platforms SRX Series, vSRX

This example shows how to configure VPLS with BGP signaling between two devices.

- Requirements on page 151
- Overview on page 151
- Configuration on page 152
- Verification on page 162

Requirements

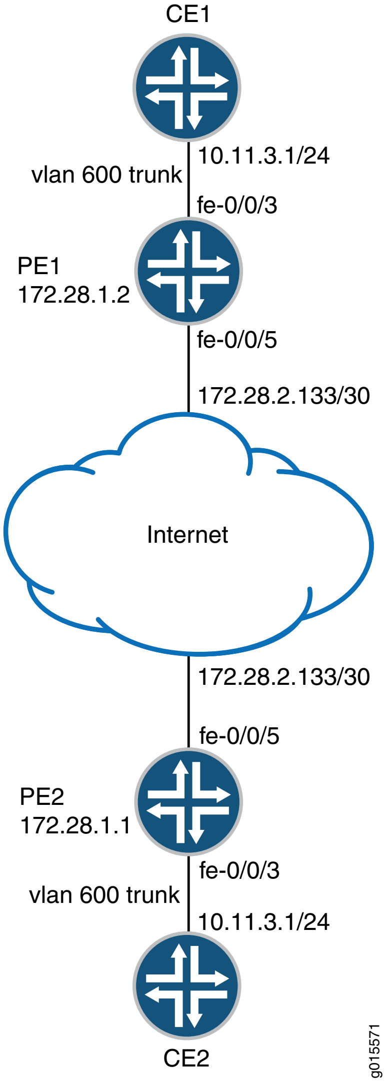
Before you begin, see *Understanding Selective Stateless Packet-Based Services* .

Overview

This example shows a minimum configuration for PE devices and CE devices to create a VPLS network with BGP signaling. The topology consists of two PE devices and two CE devices. In this example, you configure a VPLS routing instance vpls-instance between two PE devices, PE1 and PE2. You also configure the CE1 and CE2 devices that use Ethernet-based interfaces to connect VLAN 600 to their local PE devices. On the CE1 device, configure the Fast Ethernet interface that connects to the PE1 device. The VLAN identifier and IP address must match those of the CE2 device.

Figure 14 on page 152 shows the topology used in this example.

Figure 14: Configuring VPLS with BGP Signaling



Configuration

- Configuring the CE1 Device on page 152
- Configuring the PE1 Device on page 153
- Configuring the PE2 Device on page 157
- Configuring the CE2 Device on page 161

Configuring the CE1 Device

CLI Quick Configuration

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

```
set interfaces fe-0/0/3 vlan-tagging
set interfaces fe-0/0/3 unit 0 vlan-id 600
set interfaces fe-0/0/3 unit 0 family inet address 10.11.3.1/24
```

Step-by-Step Procedure

The following example requires you to navigate various levels in the configuration hierarchy. For instructions on how to do that, see *Using the CLI Editor in Configuration Mode*.

1. Enable VLAN tagging on the VPLS interface.

   ```
   [edit interfaces fe-0/0/3]
   user@host# set vlan-tagging
   ```

2. Configure the VLAN ID on the logical interface.

   ```
   [edit interfaces fe-0/0/3 unit 0]
   user@host# set vlan-id 600
   ```

3. Configure the VPLS family on the logical interface.

   ```
   [edit interfaces fe-0/0/3 unit 0]
   user@host# set family inet address 10.11.3.1/24
   ```

Results

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

```
fe-0/0/3 {
  vlan-tagging;
  unit 0 {
    vlan-id 600;
    family inet {
      address 10.11.3.1/24;
    }
  }
}
```

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

Configuring the PE1 Device

CLI Quick Configuration

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

```
set system host-name PE1
set interfaces fe-0/0/3 description "CE1 on PE1"
set interfaces fe-0/0/3 vlan-tagging
set interfaces fe-0/0/3 encapsulation vlan-vpls
set interfaces fe-0/0/3 unit 0 encapsulation vlan-vpls
set interfaces fe-0/0/3 unit 0 vlan-id 600
set interfaces fe-0/0/3 unit 0 family vpls
set interfaces fe-0/0/5 vlan-tagging
set interfaces fe-0/0/5 unit 37 vlan-id 37
set interfaces fe-0/0/5 unit 37 family inet address 172.28.2.133/30
set interfaces fe-0/0/5 unit 37 family mpls
```

```
set interfaces lo0 unit 0 family inet address 172.28.1.2/32
set routing-options router-id 172.28.1.2
set routing-options autonomous-system 65512
set protocols rsvp interface fe-0/0/5.37
set protocols mpls label-switched-path pe1-to-pe2 to 172.28.1.1
set protocols mpls interface fe-0/0/5.37
set protocols mpls interface lo0.0
set protocols bgp group vpls-peering type internal
set protocols bgp group vpls-peering local-address 172.28.1.2
set protocols bgp group vpls-peering family l2vpn signaling
set protocols bgp group vpls-peering neighbor 172.28.1.1
set protocols ospf area 0.0.0.0 interface lo0.0 passive
set protocols ospf area 0.0.0.0 interface fe-0/0/5.37
set routing-instances vpls-instance description "Routing instance from VPLS routing"
set routing-instances vpls-instance instance-type vpls
set routing-instances vpls-instance interface fe-0/0/3.0
set routing-instances vpls-instance route-distinguisher 172.28.1.2:1
set routing-instances vpls-instance vrf-target target:65512:1
set routing-instances vpls-instance protocols vpls site-range 10
set routing-instances vpls-instance protocols vpls no-tunnel-services site site10
  automatic-site-id
```

Step-by-Step Procedure

The following example requires you to navigate various levels in the configuration hierarchy. For instructions on how to do that, see *Using the CLI Editor in Configuration Mode*.

To configure PE1:

1. Configure the hostname for the PE1 device.

   ```
   [edit ]
   user@host# set system host-name PE1
   ```

2. Configure VPLS VLAN encapsulation on the VPLS PE1 device.

   ```
   [edit interfaces]
   user@host# set fe-0/0/3 description "CE1 on PE1"
   user@host# set fe-0/0/3 vlan-tagging
   user@host# set fe-0/0/3 encapsulation vlan-vpls
   user@host# set fe-0/0/3 unit 0 encapsulation vlan-vpls
   user@host# set fe-0/0/3 unit 0 vlan-id 600
   user@host# set fe-0/0/3 unit 0 family vpls
   ```

3. Configure the routing interface on the VPLS PE1 device.

   ```
   [edit interfaces]
   user@host# set fe-0/0/5 vlan-tagging
   user@host# set fe-0/0/5 unit 37 vlan-id 37
   user@host# set fe-0/0/5 unit 37 family inet address 172.28.2.133/30
   user@host# set fe-0/0/5 unit 37 family mpls
   user@host# set lo0 unit 0 family inet address 172.28.1.2/32
   ```

   NOTE: For this example, it is optional to configure VLAN tagging. Remove the VLAN tagging configuration on the physical interfaces if you do not plan to configure VLAN tagging.

4. Configure the routing options on the VPLS PE1 device.

```
[edit routing-options]
user@host# set router-id 172.28.1.2
user@host# set autonomous-system 65512
```

5. Configure RSVP on the VPLS PE1 device.

```
[edit protocols]
user@host# set rsvp interface fe-0/0/5.37
```

6. Configure MPLS on the VPLS PE1 device.

```
[edit protocols]
user@host# set mpls label-switched-path pe1-to-pe2 to 172.28.1.1
user@host# set mpls interface fe-0/0/5.37
user@host# set mpls interface lo0.0
```

7. Configure BGP on the VPLS PE1 device.

```
[edit protocols]
user@host# set bgp group vpls-peering type internal
user@host# set bgp group vpls-peering local-address 172.28.1.2
user@host# set bgp group vpls-peering family l2vpn signaling
user@host# set bgp group vpls-peering neighbor 172.28.1.1
```

8. (Optional) Configure OSPF on the VPLS PE1 device.

NOTE: For this example, it is optional to configure OFPF. You must configure OSPF only in cases where two PE devices are not connected directly.

```
[edit protocols]
user@host# set ospf area 0.0.0.0 interface lo0.0 passive
user@host# set ospf area 0.0.0.0 interface fe-0/0/5.37
```

9. Create a VPLS routing instance.

```
[edit ]
user@host# set routing-instances vpls-instance
```

10. Configure a VPLS routing instance.

```
[edit routing-instances vpls-instance]
user@host# set description "Routing instance from VPLS routing"
user@host# set instance-type vpls
user@host# set interface fe-0/0/3.0
user@host# set route-distinguisher 172.28.1.2:1
user@host# set vrf-target target:65512:1
user@host# set protocols vpls site-range 10
user@host# set protocols vpls no-tunnel-services site site10 automatic-site-id
```

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

```
[edit]
```

```
user@host# show system
host-name PE1;

[edit]
user@host# show interfaces
fe-0/0/5 {
   vlan-tagging;
   unit 37 {
      vlan-id 37;
      family inet {
         address 172.28.2.133/30;
      }
      family mpls;
   }
}
fe-0/0/3 {
   description "CE1 on PE1";
   vlan-tagging;
   encapsulation vlan-vpls;
   unit 0 {
      encapsulation vlan-vpls;
      vlan-id 600;
      family vpls;
   }
}
lo0 {
   unit 0 {
      family inet {
         address 172.28.1.2/32;
      }
   }
}

[edit]
user@host# show routing-options
router-id 172.28.1.2;
autonomous-system 65512;

[edit]
user@host# show protocols
rsvp {
   interface fe-0/0/5.37;
}
mpls {
   label-switched-path pe1-to-pe2 {
      to 172.28.1.1;
   }
   interface fe-0/0/5.37;
   interface lo0.0;
}
bgp {
   group vpls-peering {
      type internal;
      local-address 172.28.1.2;
      family l2vpn {
         signaling;
      }
```

```
        neighbor 172.28.1.1;
    }
}
ospf {
    area 0.0.0.0 {
        interface lo0.0 {
            passive;
        }
        interface fe-0/0/5.37;
    }
}

[edit]
user@host# show routing-instances
vpls-instance {
    description "Routing instance from VPLS routing";
    instance-type vpls;
    interface fe-0/0/3.0;
    route-distinguisher 172.28.1.2:1;
    vrf-target target:65512:1;
    protocols {
        vpls {
            site-range 10;
            no-tunnel-services;
            site site10 {
                automatic-site-id;
            }
        }
    }
}
```

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

Configuring the PE2 Device

CLI Quick Configuration

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

```
set system host-name PE2
set interfaces fe-0/0/3 description "CE2 on PE2"
set interfaces fe-0/0/3 vlan-tagging
set interfaces fe-0/0/3 encapsulation vlan-vpls
set interfaces fe-0/0/3 unit 0 encapsulation vlan-vpls
set interfaces fe-0/0/3 unit 0 vlan-id 600
set interfaces fe-0/0/3 unit 0 family vpls
set interfaces fe-0/0/5 vlan-tagging
set interfaces fe-0/0/5 unit 37 vlan-id 37
set interfaces fe-0/0/5 unit 37 family inet address 172.28.2.133/30
set interfaces fe-0/0/5 unit 37 family mpls
set interfaces lo0 unit 0 family inet address 172.28.1.1/32
set routing-options router-id 172.28.1.1
set routing-options autonomous-system 65512
set protocols rsvp interface fe-0/0/5.37
set protocols mpls label-switched-path pe2-to-pe1 to 172.28.1.2
set protocols mpls interface fe-0/0/5.37
set protocols mpls interface lo0.0
set protocols bgp group vpls-peering type internal
set protocols bgp group vpls-peering local-address 172.28.1.1
set protocols bgp group vpls-peering family l2vpn signaling
set protocols bgp group vpls-peering neighbor 172.28.1.2
set protocols ospf area 0.0.0.0 interface lo0.0 passive
set protocols ospf area 0.0.0.0 interface fe-0/0/5.37
set routing-instances vpls-instance description "Routing instance for VPLS routing"
set routing-instances vpls-instance instance-type vpls
set routing-instances vpls-instance interface fe-0/0/3.0
set routing-instances vpls-instance route-distinguisher 172.28.1.1:1
set routing-instances vpls-instance vrf-target target:65512:1
set routing-instances vpls-instance protocols vpls site-range 10
set routing-instances vpls-instance protocols vpls no-tunnel-services site site11
  automatic-site-id
```

Step-by-Step Procedure

The following example requires you to navigate various levels in the configuration hierarchy. For instructions on how to do that, see *Using the CLI Editor in Configuration Mode*.

To configure PE2:

1. Configure the hostname for the device.

   ```
   [edit ]
   user@host# set system host-name PE2
   ```

2. Configure VPLS VLAN encapsulation on the VPLS PE2 device.

   ```
   [edit interfaces]
   user@host# set fe-0/0/3 description "CE2 on PE2"
   user@host# set fe-0/0/3 vlan-tagging
   user@host# set fe-0/0/3 encapsulation vlan-vpls
   user@host# set fe-0/0/3 unit 0 encapsulation vlan-vpls
   user@host# set fe-0/0/3 unit 0 vlan-id 600
   user@host# set fe-0/0/3 unit 0 family vpls
   ```

3. Configure the routing interface on the VPLS PE2 device.

   ```
   [edit interfaces]
   user@host# set fe-0/0/5 vlan-tagging
   user@host# set fe-0/0/5 unit 37 vlan-id 37
   user@host# set fe-0/0/5 unit 37 family inet address 172.28.2.133/30
   user@host# set fe-0/0/5 unit 37 family mpls
   user@host# set lo0 unit 0 family inet address 172.28.1.1/32
   ```

   NOTE: For this example, it is optional to configure VLAN tagging. Remove the VLAN tagging configuration on the physical interfaces if you do not plan to configure VLAN tagging.

4. Configure the routing options on the VPLS PE2 device.

   ```
   [edit routing-options]
   user@host# set router-id 172.28.1.1
   ```

```
user@host# set autonomous-system 65512
```

5. Configure RSVP on the VPLS PE2 device.

```
[edit protocols]
user@host# set rsvp interface fe-0/0/5.37
```

6. Configure MPLS on the VPLS PE2 device.

```
[edit protocols]
user@host# set mpls label-switched-path pe2-to-pe1 to 172.28.1.2
user@host# set mpls interface fe-0/0/5.37
user@host# set mpls interface lo0.0
```

7. Configure BGP on the VPLS PE2 device.

```
[edit protocols]
user@host# set bgp group vpls-peering type internal
user@host# set bgp group vpls-peering local-address 172.28.1.1
user@host# set bgp group vpls-peering family l2vpn signaling
user@host# set bgp group vpls-peering neighbor 172.28.1.2
```

8. (Optional) Configure OSPF on the VPLS PE2 device.

NOTE: For this example, it is optional to configure OFPF. You must configure OSPF only in cases where two PE devices are not connected directly.

```
[edit protocols]
user@host# set ospf area 0.0.0.0 interface lo0.0 passive
user@host# set ospf area 0.0.0.0 interface fe-0/0/5.37
```

9. Create a VPLS routing instance.

```
[edit ]
user@host# set routing-instances vpls-instance
```

10. Configure a VPLS routing instance.

```
[edit routing-instances vpls-instance]
user@host# set description "Routing instance for VPLS routing"
user@host# set instance-type vpls
user@host# set interface fe-0/0/3.0
user@host# set route-distinguisher 172.28.1.1:1
user@host# set vrf-target target:65512:1
user@host# set protocols vpls site-range 10
user@host# set protocols vpls no-tunnel-services site site11 automatic-site-id
```

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

```
[edit]
user@host# show system
host-name PE2;

[edit]
```

```
user@host# show interfaces
fe-0/0/5 {
   vlan-tagging;
   unit 37 {
      vlan-id 37;
      family inet {
         address 172.28.2.133/30;
      }
      family mpls;
   }
}
fe-0/0/3 {
   description "CE2 on PE2";
   vlan-tagging;
   encapsulation vlan-vpls;
   unit 0 {
      encapsulation vlan-vpls;
      vlan-id 600;
      family vpls;
   }
}
lo0 {
   unit 0 {
      family inet {
         address 172.28.1.1/32;
      }
   }
}

[edit]
user@host# show routing-options
router-id 172.28.1.1;
autonomous-system 65512;

[edit]
user@host# show protocols
rsvp {
   interface fe-0/0/5.37;
}
mpls {
   label-switched-path pe2-to-pe1 {
      to 172.28.1.2;
   }
   interface fe-0/0/5.37;
   interface lo0.0;
}
bgp {
   group vpls-peering {
      type internal;
      local-address 172.28.1.1;
      family l2vpn {
         signaling;
      }
      neighbor 172.28.1.1;
   }
}
ospf {
        area 0.0.0.0 {
            interface lo0.0 {
                passive;
            }
            interface fe-0/0/5.37;
        }
    }
```

```
[edit]
user@host# show routing-instances
vpls-instance {
    description "Routing instance from VPLS routing";
    instance-type vpls;
    interface fe-0/0/3.0;
    route-distinguisher 172.28.1.1:1;
    vrf-target target:65512:1;
    protocols {
        vpls {
            site-range 10;
            no-tunnel-services;
            site site11 {
                automatic-site-id;
            }
        }
    }
}
```

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

Configuring the CE2 Device

CLI Quick Configuration

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

```
set interfaces fe-0/0/3 vlan-tagging
set interfaces fe-0/0/3 unit 0 vlan-id 600
set interfaces fe-0/0/3 unit 0 family inet address 10.11.3.2/24
```

Step-by-Step Procedure

The following example requires you to navigate various levels in the configuration hierarchy. For instructions on how to do that, see *Using the CLI Editor in Configuration Mode*.

1. Enable VLAN tagging on the VPLS interface.

   ```
   [edit interfaces fe-0/0/3]
   user@host# set vlan-tagging
   ```

2. Configure the VLAN ID on the logical interface.

   ```
   [edit interfaces fe-0/0/3 unit 0]
   user@host# set vlan-id 600
   ```

3. Configure the VPLS family on the logical interface.

   ```
   [edit interfaces fe-0/0/3 unit 0]
   ```

```
user@host# set family inet address 10.11.3.2/24
```

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

```
fe-0/0/3 {
   vlan-tagging;
   unit 0 {
      vlan-id 600;
      family inet {
         address 10.11.3.2/24;
      }
   }
}
```

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

NOTE: If VLAN trunking is not needed between the CE devices, remove the configuration on VLAN tagging on the interfaces connecting the CE and PE devices. Also, use ethernet-VPLS-encapsulation instead of vlan-vpls on the CE facing interfaces of the PE devices.

Verification

Confirm that the configuration is working properly.

- Verifying Interfaces on page 162
- Verifying Routing Information on page 162
- Verifying VPLS Information on page 163
- Verifying Automatic Site Identifier Generation on page 163

Verifying Interfaces

Purpose Verify that the interfaces are configured correctly.

Action From operational mode, enter the **show interfaces terse** command.

Verifying Routing Information

Purpose Verify that the routing information is configured correctly.

Action From operational mode, enter the following commands:

- **show route forwarding-table family mpls**
- **show route forwarding-table family vpls (destination | extensive | matching | table)**
- **show route instance (detail)**

Verifying VPLS Information

Purpose Verify that the VPLS is configured correctly.

Action From operational mode, enter the following commands:

- **show system statistics vpls**
- **show vpls connections**
- **show vpls statistics**

Verifying Automatic Site Identifier Generation

Purpose Verify that the automatic site identifier has been generated.

Action From operational mode, enter the **show vpls connections** command.

```
[edit]
user@host# show vpls connections
Layer-2 VPN connections:
Legend for connection status (St)
EI -- encapsulation invalid NC -- interface encapsulation not CCC/TCC/VPLS
EM -- encapsulation mismatch WE -- interface and instance encaps not same
VC-Dn -- Virtual circuit down NP -- interface hardware not present
CM -- control-word mismatch -> -- only outbound connection is up
CN -- circuit not provisioned <- -- only inbound connection is up
OR -- out of range Up -- operational
OL -- no outgoing label Dn -- down
LD -- local site signaled down CF -- call admission control failure
RD -- remote site signaled down SC -- local and remote site ID collision
LN -- local site not designated LM -- local site ID not minimum designated
RN -- remote site not designated RM -- remote site ID not minimum designated
XX -- unknown connection status IL -- no incoming label
MM -- MTU mismatch MI -- Mesh-Group ID not available
BK -- Backup connection ST -- Standby connection
PF -- Profile parse failure PB -- Profile busy
RS -- remote site standby SN -- Static Neighbor
VM -- VLAN ID mismatch
Legend for interface status
Up -- operational
Dn -- down
Instance: customer2
Local site: airwalk (2)
connection-site Type St Time last up # Up trans
4 rmt Up Mar 1 03:26:21 2012 1
Remote PE: 200.100.100.2, Negotiated control-word: No
Incoming label: 262148, Outgoing label: 262146
Local interface: lsi.1048838, Status: Up, Encapsulation: VPLS
Description: Intf - vpls customer2 local site 2 remote site 4
Instance: customer4
Local site: airwalk (6)
connection-site Type St Time last up # Up trans
8 rmt Up Feb 21 03:27:33 2012 1
```

```
Remote PE: 200.200.200.2, Negotiated control-word: No
Incoming label: 262160, Outgoing label: 262174
Local interface: lsi.1048836, Status: Up, Encapsulation: VPLS
Description: Intf - vpls customer4 local site 6 remote site 8
```

Related Documentation

- VPLS Overview on page 103
- Understanding VPLS Interfaces on page 111
- MPLS Overview on page 3

Example: Configuring BGP on the VPLS PE Router

Supported Platforms SRX1500, SRX300, SRX320, SRX340, SRX345, SRX550M, vSRX

This example shows how to configure BGP on the VPLS PE router.

- Requirements on page 164
- Overview on page 164
- Configuration on page 165
- Verification on page 165

Requirements

Before you begin:

- See *Understanding Selective Stateless Packet-Based Services* .
- Configure the interfaces that will carry the VPLS traffic between the PE router and the CE devices. See “Example: Configuring Routing Interfaces on the VPLS PE Router” on page 113 and “Example: Configuring the Interface to the VPLS CE Device” on page 114.
- Create a VPLS routing instance on each PE router that is participating in the VPLS. See “Example: Configuring the VPLS Routing Instance” on page 124.
- Configure an IGP on the PE routers to exchange routing information. See “Example: Configuring OSPF on the VPLS PE Router” on page 129.
- Configure RSVP-TE. See “Example: Configuring RSVP on the VPLS PE Router” on page 130. Then configure MPLS LSPs on the PE routers. See “Example: Configuring MPLS on the VPLS PE Router” on page 131. Alternatively, configure LDP on the PE routers. See “Example: Configuring LDP on the VPLS PE Router” on page 133.
- Configure routing options on the PE router. See “Example: Configuring Routing Options on the VPLS PE Router” on page 165.

Overview

In this example, you configure an internal BGP session between PE routers so that the routers can exchange information about routes originating and terminating in the VPLS. The PE routers use this information to determine which labels to use for traffic destined for remote sites.

NOTE: On all high-end SRX Series devices, BGP-based virtual private LAN service (VPLS) works on child ports and physical interfaces, but not over aggregated Ethernet (ae) interfaces.

Configuration

Step-by-Step Procedure

To configure BGP on the VPLS PE router:

1. Configure the BGP internal group on the VPLS PE router.

   ```
   [edit ]
   user@host# set protocols bgp group ibgp type internal local-address 10.255.7.168
     neighbor 10.255.7.164
   ```

2. Configure the BGP family L2vpn and specify NLRI signaling.

   ```
   [edit ]
   user@host# set protocols bgp family L2 VPN signaling
   ```

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

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

Verification

To verify the configuration is working properly, enter the **show protocols** command.

Related Documentation

- VPLS Configuration Overview on page 108
- VPLS Overview on page 103

Example: Configuring Routing Options on the VPLS PE Router

Supported Platforms SRX1500, SRX300, SRX320, SRX340, SRX345, SRX550M, vSRX

This example shows how to configure the routing options on the VPLS PE router.

- Requirements on page 165
- Overview on page 166
- Configuration on page 166
- Verification on page 166

Requirements

Before you begin:

- Before you begin, see *Understanding Selective Stateless Packet-Based Services* .
- Configure the interfaces that will carry the VPLS traffic between the PE router and the CE devices. See“Example: Configuring Routing Interfaces on the VPLS PE Router” on page 113 and “Example: Configuring the Interface to the VPLS CE Device” on page 114.

- Create a VPLS routing instance on each PE router that is participating in the VPLS. See "Example: Configuring the VPLS Routing Instance" on page 124.
- Configure an IGP on the PE routers to exchange routing information. See "Example: Configuring OSPF on the VPLS PE Router" on page 129
- Configure RSVP-TE, see "Example: Configuring RSVP on the VPLS PE Router" on page 130 and then MPLS LSPs on the PE routers, see "Example: Configuring MPLS on the VPLS PE Router" on page 131. Alternatively configure LDP on the PE routers, see "Example: Configuring LDP on the VPLS PE Router" on page 133.

Overview

This example describes how to specify the router ID and the AS number for each router involved in the VPLS . In this example, the routers PE1 and PE2 use the same AS number (100).

Configuration

Step-by-Step Procedure

To configure the routing options on the VPLS PE router:

1. Configure the router ID on the VPLS PE router.

```
[edit]
user@host# set routing-options router-id 10.255.7.168
```

2. Configure the AS number on the VPLS PE router.

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

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

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

Verification

To verify the configuration is working properly, enter the **show routing-options** command.

Related Documentation

- VPLS Configuration Overview on page 108
- VPLS Overview on page 103

CHAPTER 16

Configuring Encapsulation

Understanding VPLS VLAN Encapsulation

Supported Platforms SRX1500, SRX300, SRX320, SRX340, SRX345, SRX550M, vSRX

Gigabit Ethernet IQ, Gigabit Ethernet PIMs with small form-factor pluggable optics (SFPs), SRX Series devices with Gigabit Ethernet, Tri-Rate Ethernet copper, and 10-Gigabit Ethernet interfaces with VLAN tagging enabled can use flexible Ethernet services, VLAN virtual private LAN service (VPLS) encapsulation.

NOTE: VLAN encapsulation is not supported on SRX100 devices because there is no Gigabit Ethernet port.

Aggregated Ethernet interfaces configured for VPLS can use Ethernet VPLS or VLAN VPLS.

To configure the encapsulation on a Gigabit Ethernet IQ or Gigabit Ethernet physical interface, include the **encapsulation** statement at the **[edit interfaces interface-name]** hierarchy level, specifying **vlan-ccc** or **vlan-vpls**:

[edit interfaces *interface-name*] encapsulation (vlan-ccc | vlan-vpls);

To configure the encapsulation on an aggregated Ethernet interface, include the encapsulation statement at the **[edit interfaces *interface-name*]** hierarchy level, specifying **ethernet-vpls** or **vlan-vpls**:

[edit interfaces interface-name] encapsulation (ethernet-vpls | vlan-vpls);

Ethernet interfaces in VLAN mode can have multiple logical interfaces. In CCC and VPLS modes, VLAN IDs from 1 through 511 are reserved for normal VLANs, and VLAN IDs 512 through 4094 are reserved for CCC or VPLS VLANs. For 4-port Fast Ethernet interfaces, you can use VLAN IDs 512 through 1024 for CCC or VPLS VLANs.

Related Documentation

- Example: Configuring VPLS VLAN Encapsulation on Gigabit Ethernet Interfaces on page 171
- Understanding VPLS VLAN Encapsulation on a Logical Interface on page 168
- Example: Configuring Extended VLAN VPLS Encapsulation on page 172
- Example: Configuring VPLS VLAN Encapsulation on page 168
- VPLS Overview on page 103

Understanding VPLS VLAN Encapsulation on a Logical Interface

Supported Platforms SRX1500, SRX300, SRX320, SRX340, SRX345, SRX550M, vSRX

You cannot configure a logical interface with VLAN VLAN VPLS encapsulation unless you also configure the physical device with the same encapsulation or with flexible Ethernet services encapsulation. In general, the logical interface must have a VLAN ID of 512 or higher; if the VLAN ID is 511 or lower, it will be subject to the normal destination filter lookups in addition to source address filtering. However if you configure flexible Ethernet services encapsulation, this VLAN ID restriction is removed.

Ethernet interfaces in VLAN mode can have multiple logical interfaces. In VPLS mode, VLAN IDs from 1 through 511 are reserved for normal VLANs, and VLAN IDs 512 through 4094 are reserved for VPLS VLAN. For 4-port Fast Ethernet interfaces, you can use VLAN IDs 512 through 1024 for VPLS VLAN.

For encapsulation type **flexible-ethernet-services**, all VLAN IDs are valid.

Related Documentation

- VPLS Configuration Overview on page 108
- Understanding VPLS VLAN Encapsulation on page 167

Example: Configuring VPLS VLAN Encapsulation

Supported Platforms SRX1500, SRX300, SRX320, SRX340, SRX345, SRX550M, vSRX

This example shows how to configure VPLS VLAN encapsulation and enable it on the physical and the logical interfaces.

- Requirements on page 169
- Overview on page 169
- Configuration on page 169
- Verification on page 170

Requirements

Before you begin:

- Before you begin, see *Understanding Selective Stateless Packet-Based Services* .
- Configure the interfaces that will carry the VPLS traffic between the PE router and the CE devices. See "Example: Configuring Routing Interfaces on the VPLS PE Router" on page 113 and "Example: Configuring the Interface to the VPLS CE Device" on page 114.
- Create a VPLS routing instance on each PE router that is participating in the VPLS. See "Example: Configuring the VPLS Routing Instance" on page 124.
- Configure an IGP on the PE routers to exchange routing information. See "Example: Configuring OSPF on the VPLS PE Router" on page 129.
- Configure RSVP-TE, see "Example: Configuring RSVP on the VPLS PE Router" on page 130 and then MPLS LSPs on the PE routers, see "Example: Configuring MPLS on the VPLS PE Router" on page 131. Alternatively configure LDP on the PE routers, see "Example: Configuring LDP on the VPLS PE Router" on page 133.
- Configure routing options on the PE router. See "Example: Configuring Routing Options on the VPLS PE Router" on page 165.
- Configure an IBGP session between PE routers so that the routers can exchange information about routes originating and terminating in the VPLS. See "Example: Configuring BGP on the VPLS PE Router" on page 164.

Overview

This example describes how to enable VLAN tagging on VPLS interface ge-3/0/6, configure the encapsulation type on the physical and logical interfaces, and configure the VPLS family on the logical interface.

NOTE: **Perform the following CLI quick configuration and procedures on all of the PE interfaces (CE facing).**

Configuration

CLI Quick Configuration

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

```
set interfaces ge-3/0/6 vlan-tagging
set interfaces ge-3/0/6 encapsulation vlan-vpls
set interfaces ge-3/0/6 unit 0 encapsulation vlan-vpls
set interfaces ge-3/0/6 unit 0 vlan-id 512
set interfaces ge-3/0/6 unit 0 family vpls
```

Step-by-Step Procedure

The following example requires you to navigate various levels in the configuration hierarchy. For instructions on how to do that, see *Using the CLI Editor in Configuration Mode*

To configure VPLS VLAN encapsulation:

1. Enable VLAN tagging on the VPLS interface.

   ```
   [edit interfaces ge-3/0/6]
   user@host# set vlan-tagging
   ```

2. Configure the encapsulation type on the physical interface.

   ```
   [edit interfaces ge-3/0/6]
   user@host# set interfaces ge-3/0/6 encapsulation vlan-vpls
   ```

3. Configure the encapsulation type on the logical interface.

   ```
   [edit interfaces ge-3/0/6 unit 0]
   user@host# set encapsulation vlan-vpls
   ```

4. Configure the VLAN ID on the logical interface.

   ```
   [edit interfaces ge-3/0/6 unit 0]
   user@host# set vlan-id 512
   ```

5. Configure the family VPLS on the logical interface.

   ```
   [edit interfaces ge-3/0/6 unit 0]
   user@host# set family vpls
   ```

Results

From configuration mode, confirm your configuration by entering the **show interfaces ge-3/0/6** command. If the output does not display the intended configuration, repeat the configuration instructions in this example to correct it.

```
[edit]
user@host# show interfaces ge-3/0/6
vlan-tagging;
encapsulation vlan-vpls;
unit 0 {
  encapsulation vlan-vpls;
  vlan-id 512;
  family vpls;
}
```

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

Verification

Confirm that the configuration is working properly.

- Verifying VPLS VLAN Encapsulation on page 170
- Verifying VPLS VLAN Encapsulation for Logical Interfaces on page 171

Verifying VPLS VLAN Encapsulation

Purpose

Verify that the VPLS VLAN encapsulation is enabled at the interfaces.

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

Verifying VPLS VLAN Encapsulation for Logical Interfaces

Purpose Verify that the VPLS VLAN encapsulation is enabled at the logical interface.

Action From operational mode, enter the **show interfaces ge-3/0/6 unit 0** command.

Related Documentation

- VPLS Configuration Overview on page 108
- VPLS Overview on page 103

Example: Configuring VPLS VLAN Encapsulation on Gigabit Ethernet Interfaces

Supported Platforms SRX1500, SRX300, SRX320, SRX340, SRX345, SRX550M, vSRX

This example shows how to configure the VPLS VLAN encapsulation on either a Gigabit Ethernet IQ or Gigabit Ethernet physical interface.

- Requirements on page 171
- Overview on page 172
- Configuration on page 172
- Verification on page 172

Requirements

Before you begin:

- Before you begin, see *Understanding Selective Stateless Packet-Based Services* .
- Configure the interfaces that will carry the VPLS traffic between the PE router and the CE devices. See “Example: Configuring Routing Interfaces on the VPLS PE Router” on page 113 and “Example: Configuring the Interface to the VPLS CE Device” on page 114.
- Create a VPLS routing instance on each PE router that is participating in the VPLS. See “Example: Configuring the VPLS Routing Instance” on page 124.
- Configure an IGP on the PE routers to exchange routing information. See “Example: Configuring OSPF on the VPLS PE Router” on page 129.
- Configure RSVP-TE, see “Example: Configuring RSVP on the VPLS PE Router” on page 130 and then MPLS LSPs on the PE routers, see “Example: Configuring MPLS on the VPLS PE Router” on page 131. Alternatively configure LDP on the PE routers, see “Example: Configuring LDP on the VPLS PE Router” on page 133.
- Configure routing options on the PE router. See “Example: Configuring Routing Options on the VPLS PE Router” on page 165.
- Configure an IBGP session between PE routers so that the routers can exchange information about routes originating and terminating in the VPLS. See “Example: Configuring BGP on the VPLS PE Router” on page 164

Overview

This example describes how to configure Ethernet VPLS encapsulation on a Gigabit Ethernet IQ or Gigabit Ethernet physical interface and enable the VPLS family on the interface.

Configuration

Step-by-Step Procedure

To configure VPLS VLAN encapsulation on a Gigabit Ethernet IQ or Gigabit Ethernet physical interface:

1. Configure the ethernet-vpls encapsulation on the interface.

   ```
   [edit ]
   user@host# set interfaces ge-3/0/6 encapsulation ethernet-vpls
   ```

2. Enable the VPLS family on the interface.

   ```
   [edit ]
   user@host# set interfaces ge-3/0/6 unit 0 family vpls
   ```

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

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

Verification

To verify the configuration is working properly, enter the **show interfaces** command.

Related Documentation

- VPLS Configuration Overview on page 108
- VPLS Overview on page 103

Example: Configuring Extended VLAN VPLS Encapsulation

Supported Platforms SRX1500, SRX300, SRX320, SRX340, SRX345, SRX550M, vSRX

This example shows how to configure extended VLAN VPLS encapsulation and enable it on the physical and the logical interfaces.

Requirements

Before you begin:

- Before you begin, see *Understanding Selective Stateless Packet-Based Services* .
- Configure the interfaces that will carry the VPLS traffic between the PE router and the CE devices. See "Example: Configuring Routing Interfaces on the VPLS PE Router" on page 113 and "Example: Configuring the Interface to the VPLS CE Device" on page 114.
- Create a VPLS routing instance on each PE router that is participating in the VPLS. See "Example: Configuring the VPLS Routing Instance" on page 124.
- Configure an IGP on the PE routers to exchange routing information. See "Example: Configuring OSPF on the VPLS PE Router" on page 129.
- Configure RSVP-TE, see "Example: Configuring RSVP on the VPLS PE Router" on page 130 and then MPLS LSPs on the PE routers, see "Example: Configuring MPLS on the VPLS PE Router" on page 131. Alternatively configure LDP on the PE routers, see "Example: Configuring LDP on the VPLS PE Router" on page 133.
- Configure routing options on the PE router. See "Example: Configuring Routing Options on the VPLS PE Router" on page 165.
- Configure an IBGP session between PE routers so that the routers can exchange information about routes originating and terminating in the VPLS. See "Example: Configuring BGP on the VPLS PE Router" on page 164.

Overview

This example describes how to enable VLAN tagging on the VPLS interface ge-3/0/6, configure the extended-vlan-vpls type on the physical and logical interfaces, and configure the VPLS family on the logical interface.

NOTE: **Perform the following CLI quick configurations and procedures on all PE interfaces (CE facing).**

Configuration

CLI Quick Configuration

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

```
set interfaces ge-3/0/6 vlan-tagging
set interfaces ge-3/0/6 encapsulation extended-vlan-vpls
set interfaces ge-3/0/6 unit 0 vlan-id 100
set interfaces ge-3/0/6 unit 0 family vpls
```

Step-by-Step Procedure

The following example requires you to navigate various levels in the configuration hierarchy. For instructions on how to do that, see *Using the CLI Editor in Configuration Mode*.

To configure extended VPLS VLAN encapsulation:

1. Enable VLAN tagging on the VPLS interface as it will receive tagged packets from CE.

   ```
   [edit interfaces ge-3/0/6]
   user@host# set vlan-tagging
   ```

2. Configure the encapsulation type on the physical interface.

   ```
   [edit interfaces ge-3/0/6]
   user@host# set interfaces ge-3/0/6 encapsulation vlan-vpls
   ```

3. Configure the VLAN ID on the logical interface.

   ```
   [edit interfaces ge-3/0/6 unit 0]
   user@host# set encapsulation vlan-vpls vlan-id 100
   ```

4. Configure the VPLS family on the logical interface.

   ```
   [edit interfaces ge-3/0/6 unit 0]
   user@host# set family vpls
   ```

Results

From configuration mode, confirm your configuration by entering the **show interfaces ge-3/0/6** command. If the output does not display the intended configuration, repeat the configuration instructions in this example to correct it.

```
[edit]
user@host# show interfaces ge-3/0/6
vlan-tagging;
encapsulation extended-vlan-vpls;
unit 0 {
   encapsulation vlan-vpls;
   vlan-id 100;
   family vpls;
}
```

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

Verification

Confirm that the configuration is working properly.

- Verifying Extended VLAN VPLS Encapsulation on page 174
- Verifying Extended VLAN VPLS Encapsulation for Logical Interfaces on page 175

Verifying Extended VLAN VPLS Encapsulation

Purpose

Verify that the extended VLAN VPLS encapsulation is enabled at the interfaces.

Action

From operational mode, enter the **show interfaces** command.

Verifying Extended VLAN VPLS Encapsulation for Logical Interfaces

Purpose Verify that the extended VLAN VPLS encapsulation is enabled at the logical interface.

Action From operational mode, enter the **show interfaces ge-3/0/6 unit 0** command.

Related Documentation

- VPLS Configuration Overview on page 108
- VPLS Overview on page 103

PART 6

Configuration Statements and Operational Commands

CHAPTER 17

Configuration Statements

condition (Policy Options)

Syntax

```
condition condition-name {
  if-route-exists address table table-name ;
  route-active-on (node0 | node1);
}
```

Hierarchy Level [edit policy-options]

Release Information Statement introduced in Release 9.0 of Junos OS.

Description For chassis cluster configurations, specify the match condition for use in routing to a redundant Ethernet (**reth**) interface.

Options ***condition-name*** —Name of the routing policy match condition.

The remaining statement is explained separately.

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

Related Documentation

- *Junos OS Security Configuration Guide*

family (Security Forwarding Options)

Supported Platforms SRX Series, vSRX

Syntax

```
family {
   inet6 {
      mode (drop | flow-based | packet-based);
   }
   iso {
      mode packet-based;
   }
   mpls {
      mode packet-based;
   }
}
```

Hierarchy Level [edit security forwarding-options]

Release Information Statement introduced in Junos OS Release 8.5 .

Description Determine the protocol family to be used for packet forwarding.

NOTE: Packet-based processing is not supported on the following SRX Series devices: SRX5400, SRX5600, and SRX5800.

Options The remaining statements are explained separately. See CLI Explorer.

Required Privilege Level

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

Related Documentation

- MPLS Overview on page 3

flow-server (Forwarding Options)

Supported Platforms SRX Series, vSRX

Syntax

```
flow-server ip-address-or-host-name {
  aggregation {
    autonomous-system;
    destination-prefix;
    protocol-port;
    source-destination-prefix {
      caida-compliant;
    }
    source-prefix;
  }
  autonomous-system-type (origin | peer);
  (local-dump | no-local-dump);
  port port-number;
  source-address ip-address;
  version (5 | 500 |8);
  version9 {
    template template-name;
  }
}
```

Hierarchy Level

```
[edit forwarding-options sampling family inet output ]
[edit forwarding-options sampling family inet6 output ]
```

Release Information Statement introduced in Junos OS Release 10.4. Support for family inet6 added in Junos OS Release 12.1X45-D10.

Description Configure sending traffic aggregates in cflowd format.

Options

- **aggregation (version 8 only**—Aggregations to perform for exported flows.
- **autonomous-system-type**—Type of autonomous system number to export.
- **local-dump**—Dump cflowd records to log file before exporting.
- **no-local-dump**—Do not dump cflowd records to log file before exporting.
- **port**—UDP port number on host collecting cflowd packets.
- **source-address**—Source IPv4/IPv6 address for cflowd packets.

NOTE: The flow server may be IPv4 or IPv6 when you configure the collector IP address using the **set forwarding-options sampling family inet flow server** command, but only IPv4 sampling is achieved.

The flow server may be IPv4 or IPv6 when you configure the collector IP address using the **set forwarding-options sampling family inet6 flow server** command, but only IPv6 sampling is achieved.

- **version**—Format of exported cflowd aggregates.
- **version9**—Exported data in version 9 format.

Required Privilege Level

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

Related Documentation

- *Configuring the inet6 IPv6 Protocol Family*
- *Enabling Flow-Based Processing for IPv6 Traffic*

forwarding-options (Security)

Supported Platforms SRX Series, vSRX

Syntax

```
forwarding-options {
  family {
    inet6 {
      mode (drop | flow-based | packet-based);
    }
    iso {
      mode packet-based;
    }
    mpls {
      mode packet-based;
    }
  }
}
```

Hierarchy Level [edit security]

Release Information Statement introduced in Junos OS Release 8.5 .

Description Determine how the **inet6, iso**, and **mpls** protocol families manage security forwarding options.

NOTE:

- Packet-based processing is not supported on the following SRX Series devices: SRX5400, SRX5600, and SRX5800.
- On SRX Series devices, the default mode for processing traffic is flow mode. To configure an SRX Series device as a border router, you must change the mode from flow-based processing to packet-based processing. Use the **set security forwarding-options family mpls mode packet-based** statement to configure the SRX device to packet mode. You must reboot the device for the configuration to take effect.

Options The remaining statements are explained separately. See CLI Explorer.

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

Related Documentation

- MPLS Overview on page 3
- *Understanding Packet-Based Processing*
- *Juniper Networks Devices Processing Overview*

fragment

Supported Platforms	SRX Series, vSRX
Syntax	fragment;
Hierarchy Level	[edit security screen ids-option *screen-name* icmp]
Release Information	Statement introduced in Junos OS Release 8.5.
Description	Configure the device to detect and drop any ICMP frame with the More Fragments flag set or with an offset indicated in the **offset** field.
Required Privilege Level	security—To view this statement in the configuration. security-control—To add this statement to the configuration.
Related Documentation	• MPLS Overview on page 3

hash-key (Forwarding Options)

Supported Platforms SRX Series, vSRX

Syntax

```
hash-key {
   family inet {
      layer-3;
      layer-4;
      session-id;
   }
   family mpls {
      label-1;
      label 2;
      label-3;
      no-labels;
      payload {
         ip {
            layer-3-only;
            port-data {
               destination-lsb;
               destination-msb;
               source-lsb;
               source-msb;
            }
         }
      }
   }
   family multiservice {
      destination-mac;
      source-mac;
   }
}
```

Hierarchy Level **[edit forwarding-options]**

Release Information Statement modified in Junos OS Release 10.2.

Description Select which packet header data to use for per-flow load balancing.

Options

- **inet**—IPv4 protocol family.
- **mpls**—MPLS protocol family.
- **layer-3**—Incorporate Layer 3 data into the hash key.
- **layer-4**—Incorporate Layer 4 data into the hash key.
- **session-id**—Incorporate session ID data into the hash key (SRX3000 and SRX5000 lines only). The session ID data has higher precedence than the Layer 3 or 4 information.
- **label-1**—Incorporate the first MPLS label into the hash key.
- **label-2**—Incorporate the second MPLS label into the hash key.
- **label-3**—Incorporate the third MPLS label into the hash key.

- **no-labels**—Include no MPLS labels into the hash key.
- **payload**—Incorporate payload data into the hash key.
- **ip**—Include the IP address of the IPv4 or IPv6 payload into the hash key.
- **layer-3–only**—Include only Layer 3 IP information.
- **port-data**—Include the source and destination port field information.
- **source-msb**—Include the most significant byte of the source port.
- **source-lsb**—Include the least significant byte of the source port.
- **destination-msb**—Include the most significant byte of the destination port.
- **destinatione-lsb**—Include the least significant byte of the destination port.
- **source-mac**—Include source MAC address in hash key.
- **destinatione-mac**—Include destination MAC address in hash key.

Required Privilege Level

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

Related Documentation

- MPLS Overview on page 3

iso (Security Forwarding Options)

Supported Platforms SRX1500, SRX300, SRX320, SRX340, SRX345, SRX550M, vSRX

Syntax

```
iso {
  mode packet-based;
}
```

Hierarchy Level [edit security forwarding-options family]

Release Information Statement introduced in Junos OS Release 8.5.

Description Enable the forwarding of IS-IS traffic. By default, the device drops IS-IS traffic.

NOTE: Junos OS security processing is not applied to IS-IS packets forwarded by the device.

NOTE: Packet-based processing is not supported on the following SRX Series devices: SRX1500, SRX5600, and SRX5800.

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

Related Documentation

label-switched-path (Protocols MPLS)

Supported Platforms ACX Series, M Series, MX Series, PTX Series, SRX210, T1600, T640

Syntax

```
label-switched-path lsp-name {
  disable;
  adaptive;
  admin-down;
  admin-group {
    exclude [ group-names ];
    include-all [ group-names ];
    include-any [ group-names ];
  }
  auto-bandwidth {
    adjust-interval seconds;
    adjust-threshold percentage;
    maximum-bandwidth bps;
    minimum-bandwidth bps;
    monitor-bandwidth;
  }
  bandwidth bps {
    ct0 bps;
    ct1 bps;
    ct2 bps;
    ct3 bps;
  }
  class-of-service cos-value;
  description text;
  entropy-label;
  fast-reroute {
    (bandwidth bps | bandwidth-percent percentage);
    (exclude [ group-names ] | no-exclude);
    hop-limit number;
    (include-all [ group-names ] | no-include-all);
    (include-any [ group-names ] | no-include-any);
  }
  from address;
  install {
    destination-prefix/prefix-length <active>;
  }
  inter-domain;
  ldp-tunneling;
  link-protection;
  lsp-attributes {
    encoding-type (ethernet | packet | pdh | sonet-sdh);
    gpid (ethernet | hdlc | ipv4 | pos-scrambling-crc-16 | pos-no-scrambling-crc-16 |
      pos-scrambling-crc-32 | pos-no-scrambling-crc-32 | ppp);
    signal-bandwidth type;
    switching-type (fiber | lambda | psc-1 | tdm);
  }
  metric metric;
  no-cspf;
  no-decrement-ttl;
  node-link-protection;
  optimize-timer seconds;
```

```
p2mp lsp-name;
policing {
   filter filter-name;
   no-auto-policing;
}
preference preference;
primary path-name {
   adaptive;
   admin-group {
      exclude [ group-names ];
      include-all [ group-names ];
      include-any [ group-names ];
   }
   bandwidth bps {
      ct0 bps;
      ct1 bps;
      ct2 bps;
      ct3 bps;
   }
   class-of-service cos-value;
   hop-limit number;
   no-cspf;
   no-decrement-ttl;
   optimize-timer seconds;
   preference preference;
   priority setup-priority reservation-priority;
   (record | no-record);
   select (manual | unconditional);
   standby;
}
priority setup-priority reservation-priority;
(random | least-fill | most-fill);
(record | no-record);
retry-limit number;
retry-timer seconds;
revert-timer seconds;
secondary path-name {
   adaptive;
   admin-group {
      exclude[ group-names ];
      include-all [ group-names ];
      include-any [ group-names ];
   }
   bandwidth bps {
      ct0 bps;
      ct1 bps;
      ct2 bps;
      ct3 bps;
   }
   class-of-service cos-value;
   hop-limit number;
   no-cspf;
   no-decrement-ttl;
   optimize-timer seconds;
   preference preference;
   priority setup-priority reservation-priority;
```

```
        (record | no-record);
        select (manual | unconditional);
        standby;
    }
    soft-preemption;
    standby;
    to address;
    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 mpls],
[edit protocols mpls]

Release Information Statement introduced before Junos OS Release 7.4.

Description Configure an LSP to use in dynamic MPLS. When configuring an LSP, you must specify the address of the egress router in the **to** statement. All remaining statements are optional.

Options *lsp-name*—Name that identifies the LSP. The name can be up to 64 characters and can contain letters, digits, periods, and hyphens. To include other characters, enclose the name in quotation marks. The name must be unique within the ingress 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

- *Minimum MPLS Configuration*
- *Configuring the Ingress and Egress Router Addresses for LSPs*
- *Configuring Primary and Secondary LSPs*

mpls (Security Forwarding Options)

Supported Platforms SRX300, SRX320, SRX340, SRX345, SRX550M, vSRX

Syntax

```
mpls {
  mode packet-based;
}
```

Hierarchy Level [edit security forwarding-options family]

Release Information Statement introduced in Junos OS Release 9.0.

Description Enable the forwarding of MPLS traffic. By default, the device drops MPLS traffic.

CAUTION: Because MPLS operates in packet mode, security services are not available.

NOTE: Packet-based processing is not supported on the following SRX Series devices: SRX1500, SRX5600, and SRX5800.

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

Related Documentation

- MPLS Overview

multicast-scope

Supported Platforms	SRX Series, vSRX
Syntax	multicast-scope (*scope-value* | global | link-local | node-local | organization-local | orhigher | orlower | site-local);
Hierarchy Level	[edit policy-options policy-statement *policy-name* from]
Release Information	Statement introduced in Junos OS Release 9.5.
Description	Configure multicast scoping to match the routing policy.
Options	• ***scope-value*** **<orhigher | orlower>**—The multicast-scope value is a number from 0 through 15. • **global <orhigher | orlower>**—Global multicast scope • **link-local <orhigher | orlower>**—Link-local scope • **node-local <orhigher | orlower>**—Node-local scope • **organization-local <orhigher | orlower>**—Organizational-local scope • **orhigher**—Match on numerically higher scopes • **orlower**—Match on numerically lower scopes • **site-local <orhigher | orlower>**—Site-local values
Required Privilege Level	routing—To view this statement in the configuration. routing-control—To add this statement to the configuration.
Related Documentation	• MPLS Overview on page 3

policer (Firewall)

Supported Platforms SRX Series, vSRX

Syntax

```
policer policer-name {
   filter-specific;
   if-exceeding {
      (bandwidth-limit bps | bandwidth-percent percentage);
      burst-size-limit bytes;
   }
   logical-interface-policer;
   then {
      discard;
      forwarding-class forwarding-class-name;
      loss-priority (high | low |medium-high |medium-low);
      out-of-profile;
   }
}
```

Hierarchy Level [edit firewall]

Release Information Command introduced in Junos OS Release 9.5.

Description Configure policer rate limits and actions. To activate a policer, you must include the policer action modifier in the **then** statement in a firewall filter term or on an interface.

Options

- *policer-name*—Name of the policer to evaluate when packets are received on the interface
- **bandwidth-limit** *bps*—Specify the bandwidth limit as a number of bits per second
- **bandwidth-percent** *percentage*—Specify the bandwidth limit in percentage value
- **burst-size-limit** *bytes*—Specify the burst size limit as a number of bytes
- **filter-specific**— Specify that policer is filter-specific
- **logical-interface-policer**— Specify that policer is logical interface policer
- **discard**—Always discard non conforming red packets.
- **forwarding-class** *classname*—Specify the particular forwarding class
- **loss-priority**—Set the loss priority to high or low
- **out-of-profile**— Discard packets only if both congested and over threshold

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

Related Documentation

- simple-filter (Firewall) on page 195

simple-filter (Firewall)

Supported Platforms SRX5400, SRX5600, SRX5800

Syntax

```
simple-filter filter-name {
  term term-name {
    from {
      match-conditions;
    }
    then {
      (accept | discard);
      forwarding-class class-name;
      policer policer-name;
      three-color-policer policer-name {
        (single-rate single-rate-policer-name | two-rate two-rate-policer-name);
      }
    }
  }
}
```

Hierarchy Level [edit firewall family *family-name*]

Release Information Statement introduced in Junos OS Release 9.5.

Description Define a simple filter. Simple filters are recommended for metropolitan Ethernet applications.

Options

- **from**—Match packet fields to values. If the **from** option is not included, all packets are considered to match and the actions and action modifiers in the **then** statement are taken.
- **match-conditions**—One or more conditions to use to make a match.
- ***term-name***—Name that identifies the term. The name can contain letters, numbers, and hyphens (-), and can be up to 255 characters long. To include space in the name, enclose it in quotation marks (“ ”).
- **then**—Actions to take on matching packets. If the then option is not included and a packet matches all the conditions in the from statement, the packet is accepted.

NOTE: On SRX1400, SRX3400, and SRX3600 devices, the Forwarding class as match condition feature is not supported by a simple filter.

NOTE: SRX3400 and SRX3600 devices have the following limitations of a simple filter:

- The forwarding class is the match condition.

- In the packet processor on an IOC, up to 400 logical interfaces can be applied with simple filters.
- In the packet processor on an IOC, the maximum number of terms of all simple filters is 2000.
- In the packet processor on an IOC, the maximum number of policers is 2000
- In the packet processor on an IOC, the maximum number of three-color-policers is 2000
- The maximum burst size of a policer or three-color-policer is 16 MB.

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

Related Documentation

- policer (Firewall) on page 194

template (Flow Monitoring)

Supported Platforms SRX Series, vSRX

Syntax

```
template template-name {
  flow-active-timeout seconds;
  flow-inactive-timeout seconds;
  ipv4-template;
  ipv6-template;
  option-refresh-rate {
    packets packets;
    seconds seconds;
  }
  template-refresh-rate {
    packets packets;
    seconds seconds;
  }
}
```

Hierarchy Level [edit services flow-monitoring version9]

Release Information Statement introduced in Junos OS Release 10.4. Support for family inet6 added in Junos OS Release 12.1X45-D10.

Description Specify one or more version 9 templates.

Options

- **flow-active-timeout**—Interval after which active flow is exported. The range is from 10 through 600. The default value is 60.
- **flow-inactive-timeout**—Period of inactivity that marks a flow inactive. The range is from 10 through 600. The default value is 60.
- **ipv4-template**—IPv4 template configuration.
- **ipv6-template**—IPv6 template configuration.
- **option-refresh-rate**—Rate at which the device sends options. The range is from 1 through 480,000. The default value is 4800.
  - **packets**—Specify the number of packets. The range is from 1 through 480,000.
  - **seconds**—Specify the number of seconds. The range is from 10 through 600.
- **template-refresh-rate**—Rate at which the device sends template definitions. The range is from 1 through 480,000. The default value is 4800.
  - **packets**—Specify the number of packets. The range is from 1 through 480,000.
  - **seconds**—Specify the number of seconds. The range is from 10 through 600.

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

Related Documentation

- version9 (Flow Server) on page 203

traceoptions (Security Flow)

Supported Platforms SRX Series, vSRX

Syntax

```
traceoptions {
  file {
    filename;
    files number;
    match regular-expression;
    size maximum-file-size;
    (world-readable | no-world-readable);
  }
  flag flag;
  no-remote-trace;
  packet-filter filter-name {
    conn-tag session-conn
    destination-port port-identifier;
    destination-prefix address;
    interface interface-name;
    protocol protocol-identifier;
    source-port port-identifier;
    source-prefix address;
  }
  rate-limit messages-per-second;
  trace-level (brief | detail | error);
}
```

Hierarchy Level [edit security flow]

Release Information Statement introduced in Junos OS Release 8.5. Statement updated in Junos OS Release 12.1X46-D10 with the **trace-level** option and additional flags. The statement was updated in Junos OS Release 15.1X49-D70 with the addition of the conn-tag filter parameter.

Description Configure flow tracing options.

Options **file**—Configure the trace file options.

filename—Name of the file to receive the output of the tracing operation. Enclose the name within quotation marks. All files are placed in the directory **/var/log**. By default, the name of the file is the name of the process being traced.

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

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

Range: 2 through 1000 files

Default: 10 files

match ***regular-expression***—Refine the output to include lines that contain the regular expression.

size ***maximum-file-size***—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 and a filename.

Syntax: ***x*** **K** to specify KB, ***x*** **m** to specify MB, or ***x*** **g** to specify GB

Range: 0 KB through 1 GB

Default: 128 KB

world-readable | no-world-readable—By default, log files can be accessed only by the user who configures the tracing operation. The **world-readable** option enables any user to read the file. To explicitly set the default behavior, use the **no-world-readable** option.

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

all—Trace with all flags enabled

basic-datapath—Trace basic packet flow activity

fragmentation—Trace IP fragmentation and reassembly events

high-availability—Trace flow high-availability information

host-traffic—Trace flow host traffic information

multicast—Trace multicast flow information

route—Trace route lookup information

session—Trace session creation and deletion events

session-scan—Trace session scan information

tcp-basic—Trace TCP packet flow information

tunnel—Trace tunnel information

no-remote-trace—Set remote tracing as disabled.

packet-filter ***filter-name***—Packet filter to enable during the tracing operation. Configure the filtering options.

destination-port ***port-identifier***—Match TCP/UDP destination port

destination-prefix ***address***—Destination IP address prefix

interface ***interface-name***—Logical interface

protocol ***protocol-identifier***—Match IP protocol type

source-port ***port-identifier***—Match TCP/UDP source port

source-prefix ***address***—Source IP address prefix

rate-limit ***messages-per-second***—Limit the incoming rate of trace messages.

trace-level—Set the level for trace logging. This option is available only when the flag is set.

- **brief**—Trace key flow information, such as message types sent between SPU and central point, policy match, and packet drop reasons.
- **detail**—Trace extensive flow information, such as detailed information about sessions and fragments. Detail is the default level.
- **error**—Trace error information, such as system failure, unknown message type, and packet drop.

Required Privilege Level

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

Related Documentation

- *Juniper Networks Devices Processing Overview*

traffic-engineering (Protocols MPLS)

Supported Platforms	ACX Series, EX Series, M Series, MX Series, PTX Series, QFX Series standalone switches, SRX Series, T Series
Syntax	traffic-engineering (bgp | bgp-igp | bgp-igp-both-ribs | mpls-forwarding);
Hierarchy Level	[edit logical-systems *logical-system-name* protocols mpls], [edit protocols mpls]
Release Information	Statement introduced before Junos OS Release 7.4. Statement introduced in Junos OS Release 12.1 for EX Series switches. Statement introduced in Junos OS Release 12.3X50 for the QFX Series.
Description	Select whether MPLS performs traffic engineering on BGP destinations only or on both BGP and IGP destinations. Affects only LSPs originating from this routing device, not transit or egress LSPs.
Default	bgp
Options	**bgp**—On BGP destinations only. Ingress routes are installed in the inet.3 routing table. **bgp-igp**—On both BGP and IGP destinations. Ingress routes are installed in the inet.0 routing table. If IGP shortcuts are enabled, the shortcut routes are automatically installed in the inet.0 routing table. **bgp-igp-both-ribs**—On both BGP and IGP destinations. Ingress routes are installed in the inet.0 and inet.3 routing tables. This option is used to support VPNs. **mpls-forwarding**—On both BGP and IGP destinations. Use ingress routes for forwarding only, not for routing.
Required Privilege Level	routing—To view this statement in the configuration. routing-control—To add this statement to the configuration.
Related Documentation	• *Configuring Traffic Engineering for LSPs* • *Configuring MPLS on Provider Edge Switches Using IP Over MPLS (CLI Procedure)*

version9 (Flow Server)

Supported Platforms

SRX Series, vSRX

Syntax

```
version9 {
   template template-name;
}
```

Hierarchy Level

[edit forwarding-options sampling family inet output flow-server *ip-address*]
[edit forwarding-options sampling family inet6 output flow-server *ip-address*]

Release Information

Statement introduced in Junos OS Release 10.4. Support for family inet6 added in Junos OS Release 12.1X45-D10.

Description

Export data in version 9 format.

Options

- **apply-groups**—Groups from which to inherit configuration data.
- **apply-groups-except**—Do not inherit configuration data from these groups.
- **template**—Template configuration.

Required Privilege Level

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

Related Documentation

- forwarding-options (Security) on page 184

CHAPTER 18

Operational Commands

show bgp neighbor (View)

Supported Platforms SRX Series, vSRX

Syntax

show bgp neighbor
< ***neighbor-address*>**
<instance ***instance*** **>**

Release Information Command modified in Junos OS Release 8.5.

Description Display the state of the specified Border Gateway Protocol (BGP) neighbor. If a peer is forced to Idle state because of license check failure, the output displays the state and the reason—**LicenseCheckFailed**.

Options

- **instance** ***instance***—(Optional) Display peer information for a particular routing instance.
- ***neighbor-address***—(Optional) Display information for only the BGP peer at the specified IP address.

Required Privilege Level view

Related Documentation

- MPLS Overview on page 3

List of Sample Output

show bgp neighbor 5.5.5.2 on page 208
show bgp neighbor instance master on page 209

Output Fields Table 3 on page 206 lists the output fields for the **show bgp neighbor** command. Output fields are listed in the approximate order in which they appear.

Table 3: show bgp neighbor Output Fields

Field Name	Field Description
Peer	Address of the BGP neighbor. The address is followed by the neighbor's port number.
AS	AS number of the peer.
Local	Address of the local device. The address is followed by the peer's port number.
Type	Type of peer: Internal or External.

Table 3: show bgp neighbor Output Fields *(continued)*

Field Name	Field Description
State	Current state of the BGP session: • **Active**—BGP is initiating a transport protocol connection in an attempt to connect to a peer. If the connection is successful, BGP sends an open message. • **Connect**—BGP is waiting for the transport protocol connection to complete. • **Established**—The BGP session has been established, and the peers are exchanging update messages. • **Idle**—Either the BGP license check failed, or this is the first stage of a connection and BGP is waiting for a Start event. • **OpenConfirm**—BGP has acknowledged receipt of an open message from the peer and is waiting to receive a keepalive or notification message. • **OpenSent**—BGP has sent an open message and is waiting to receive an open message from the peer.
Flags	Internal BGP flags: • **Aggregate Label**—BGP has aggregated a set of incoming labels (labels received from the peer) into a single forwarding label. • **CleanUp**—The peer session is being shut down. • **Delete**—This peer has been deleted. • **Idled**—This peer has been permanently idled. • **Initializing**—The peer session is initializing. • **SendRtn**—Messages are being sent to the peer. • **Sync**—This peer is synchronized with the rest of the peer group. • **TryConnect**—Another attempt is being made to connect to the peer. • **Unconfigured**—This peer is not configured. • **WriteFailed**—An attempt to write to this peer failed. • **Last State**—Previous state of the BGP session.
Last State	Previous state of the BGP session.
Last Event	Last activity that occurred in the BGP session: • **Closed**—The BGP session closed. • **ConnectRetry**—The transport protocol connection failed, and BGP is trying again to connect. • **HoldTime**—The session ended because the hold timer expired. • **KeepAlive**—The local device sent a BGP keepalive message to the peer. • **Open**—The local device sent a BGP open message to the peer. • **OpenFail**—The local device did not receive an acknowledgment of a BGP open message from the peer. • **RecvKeepAlive**—The local device received a BGP keepalive message from the peer. • **RecvNotify**—The local device received a BGP notification message from the peer. • **RecvOpen**—The local device received a BGP open message from the peer. • **RecvUpdate**—The local device received a BGP update message from the peer. • **Start**—The peering session started. • **Stop**—The peering session stopped. • **TransportError**—A TCP error occurred.

Table 3: show bgp neighbor Output Fields *(continued)*

Field Name	Field Description
Last Error	Last error that occurred in the BGP session: • **Cease**—An error occurred, such as a version mismatch, that caused the session to close. • **Finite State Machine Error**—In setting up the session, BGP received a message that it did not understand. • **Hold Time Expired**—The session's hold time expired. • **Message Header Error**—The header of a BGP message was malformed. • **Open Message Error**—A BGP open message contained an error. • **None**—No errors occurred in the BGP session. • **Update Message Error**—A BGP update message contained an error.
Export	Name of the export policy that is configured on the peer.
Import	Name of the import policy that is configured on the peer.
Options	Configured BGP options: • **AddressFamily**—Configured address family: inet or inet-vpn. • **GracefulRestart**—Graceful restart is configured. • **HoldTime**—Hold time configured with the **hold-time** statement. The hold time is three times the interval at which keepalive messages are sent. • **Local Address**—Address configured with the **local-address** statement. • **NLRI**—Configured multicast BGP state for the BGP group: multicast, unicast, or both if you have configured **nlri any**. • **Peer AS**—Configured peer autonomous system (AS). • **Preference**—Preference value configured with the **preference** statement. • **Refresh**—Configured to refresh automatically when the policy changes. • **Rib-group**—Configured routing table group.
Address families configured	Names of configured address families for the VPN.
Local Address	Address of the local device.
Holdtime	Hold time configured with the **hold-time** statement. The hold time is three times the interval at which keepalive messages are sent.
Preference	Preference value configured with the **preference** statement.
Number of flaps	Number of times the BGP session has gone down and then come back up.
Trace file	Name of the file to receive the output of the tracing operation.

Sample Output

show bgp neighbor 5.5.5.2

```
user@host> show bgp neighbor 5.5.5.2
```

```
  Type: Internal    State: Idle (LicenseCheckFailed)   Flags: <ImportEval Sync>
Peer: 5.5.5.2 AS 200           Local: unspecified AS 200
  Type: Internal    State: Idle (LIcenseCheckFailed)       (route reflector
client)Flags: <ImportEval>
  Last State: Idle          Last Event: Start
  Last Error: None
  Options: <Preference LogUpDown Cluster AddressFamily PeerAS Rib-group Refresh>

  Address families configured: inet-unicast inet-vpn-unicast l2vpn-signaling
  Holdtime: 90 Preference: 170
  Number of flaps: 0
  Trace options:  all
  Trace file: /var/log/bgp size 131072 files 10
```

Sample Output

show bgp neighbor instance master

```
user@host> show bgp neighbor instance master
Peer: 5.5.5.1 AS 200           Local: 5.5.5.2 AS 200
  Type: Internal    State: Idle (LicenseCheckFailed)         Flags: <>
  Last State: Idle          Last Event: Start
  Last Error: Cease
  Export: [ static ]
  Options: <Preference LocalAddress LogUpDown AddressFamily PeerAS Rib-group
Refresh>
  Address families configured: inet-unicast inet-vpn-unicast
  Local Address: 5.5.5.2 Holdtime: 90 Preference: 170
  Number of flaps: 4
  Last flap event: RecvUpdate
  Error: 'Update Message Error' Sent: 3 Recv: 0
  Error: 'Cease' Sent: 2 Recv: 0
  Trace file: /var/log/bgp size 131072 files 10
```

show interfaces flow-statistics

Supported Platforms SRX Series, vSRX

Syntax **show interfaces flow-statistics** *<interface-name>*

Release Information Command introduced in Junos OS Release 9.2.

Description Display interfaces flow statistics.

Options *Interface-name* —(Optional) Display flow statistics about the specified interface. Following is a list of typical interface names. Replace *pim* with the PIM slot and *port* with the port number. For a complete list, see the *Interface Naming Conventions*.

- **at-*pim/0/port***—ATM-over-ADSL or ATM-over-SHDSL interface.
- **br-*pim/0/port***—Basic Rate Interface for establishing ISDN connections.
- **ce1-*pim/0/port***—Channelized E1 interface.
- **ct1-*pim/0/port***—Channelized T1 interface.
- **dl0**—Dialer Interface for initiating ISDN and USB modem connections.
- **e1-*pim/0/port***—E1 interface.
- **e3-*pim/0/port***—E3 interface.
- **fe-*pim/0/ port***—Fast Ethernet interface.
- **ge-*pim/0/port***—Gigabit Ethernet interface.
- **se-*pim/0/port***—Serial interface.
- **t1-*pim/0/port***—T1 (also called DS1) interface.
- **t3-*pim/0/ port***—T3 (also called DS3) interface.
- **wx-*slot*/0/0**—WAN acceleration interface, for the WXC Integrated Services Module (ISM 200).

Required Privilege Level view

Related Documentation

- *Juniper Networks Devices Processing Overview*
- *Understanding Interfaces*

List of Sample Output show interfaces flow-statistics (Gigabit Ethernet) on page 213

Output Fields Table 4 on page 211 lists the output fields for the **show interfaces flow-statistics** command. Output fields are listed in the approximate order in which they appear.

Table 4: show interfaces flow-statistics Output Fields

Field Name	Field Description
Traffic statistics	Number of packets and bytes transmitted and received on the physical interface.
Local statistics	Number of packets and bytes transmitted and received on the physical interface.
Transit statistics	Number of packets and bytes transiting the physical interface.
Flow input statistics	Statistics on packets received by flow module.
Flow output statistics	Statistics on packets sent by flow module.
Flow error statistics	Packet drop statistics for the flow module. For further details, see Table 5 on page 211.

Table 5: Flow Error Statistics (Packet Drop Statistics for the Flow Module)

Error	Error Description
Screen:	
Address spoofing	The packet was dropped when the screen module detected address spoofing.
Syn-attack protection	The packet was dropped because of SYN attack protection or SYN cookie protection.
VPN:	
Authentication failed	The packet was dropped because the IPsec Encapsulating Security Payload (ESP) or Authentication Header (AH) authentication failed.
No SA for incoming SPI	The packet was dropped because the incoming IPsec packet's security parameter index (SPI) does not match any known SPI.
Security association not active	The packet was dropped because an IPsec packet was received for an inactive SA.
NAT:	
Incoming NAT errors	The source NAT rule search failed, an invalid source NAT binding was found, or the NAT allocation failed.
Multiple incoming NAT	Sometimes packets are looped through the system more than once; if source NAT is specified more than once, the packet will be dropped.
Auth:	
Multiple user authentications	Sometimes packets are looped through the system more than once. Each time a packet passes through the system, that packet must be permitted by a policy. If the packet matches more than one policy that specifies user authentication, then it will be dropped.

Table 5: Flow Error Statistics (Packet Drop Statistics for the Flow Module) *(continued)*

User authentication errors	Packet was dropped because policy requires authentication; however: • Only Telnet, FTP, and HTTP traffic can be authenticated. • The corresponding authentication entry could not be found, if web-auth is specified. • The maximum number of authenticated sessions per user was exceeded.
Flow:	
No one interested in self packets	This counter is incremented for one of the following reasons: • The outbound interface is a self interface, but the packet is not marked as a to-self packet and the destination address is in a source NAT pool. • No service is interested in the to-self packet • When a zone has ident-reset service enabled, the TCP RST to IDENT request for port 113 is sent back and this counter is incremented.
No minor session	The packet was dropped because no minor sessions are available and a minor session was requested. Minor sessions are allocated for storing additional TCP state information.
No more sessions	The packet was dropped because there were no more free sessions available.
No route present	The packet was dropped because a valid route was not available to forward the packet. For new sessions, the counter is incremented for one of the following reasons: • No valid route was found to forward the packet. • A discard or reject route was found. • The route could not be added due to lack of memory. • The reverse path forwarding check failed for an incoming multicast packet. For existing sessions, the prior route was changed or deleted, or a more specific route was added. The session is rerouted, and this reroute could fail because: • A new route could not be found; either the previous route was removed, or the route was changed to discard or reject. • Multiple packets may concurrently force rerouting to occur, and only one packet can successfully complete the rerouting process. Other packets will be dropped. • The route table was locked for updates by the Routing Engine. Packets that match a new session are retried, whereas packets that match an existing session are not.
No tunnel found	The packet was dropped because a valid tunnel could not be found
No session for a gate	This counter is incremented when a packet is destined for an ALG, and the ALG decides to drop this packet.
No zone or NULL zone binding	The packet was dropped because its incoming interface was not bound to any zone.
Policy denied	The error counter is incremented for one of the following reasons: • Source and/or destination NAT has occurred and policy says to drop the packet. • Policy specifies user authentication, which failed. • Policy was configured to deny this packet.

Table 5: Flow Error Statistics (Packet Drop Statistics for the Flow Module) *(continued)*

TCP sequence number out of window	A TCP packet with a sequence number failed the TCP sequence number check that was received.
Counters Not Currently in Use	
No parent for a gate	-
Invalid zone received packet	-
No NAT gate	-

Sample Output

show interfaces flow-statistics (Gigabit Ethernet)

```
user@host> show interfaces flow-statistics ge-0/0/1.0
  Logical interface ge-0/0/1.0 (Index 70) (SNMP ifIndex 49)
    Flags: SNMP-Traps Encapsulation: ENET2
    Input packets : 5161
    Output packets: 83
    Security: Zone: zone2
    Allowed host-inbound traffic : bootp bfd bgp  dns dvmrp igmp ldp msdp nhrp
ospf pgm
    pim rip router-discovery rsvp sap vrrp dhcp finger ftp tftp ident-reset http
 https ike
   netconf ping rlogin rpm rsh snmp snmp-trap ssh telnet traceroute xnm-clear-text
 xnm-ssl
    lsping
    Flow Statistics :
    Flow Input statistics :
      Self packets :                     0
      ICMP packets :                     0
      VPN packets :                      2564
      Bytes permitted by policy :        3478
      Connections established :          1
    Flow Output statistics:
      Multicast packets :                0
      Bytes permitted by policy :        16994
    Flow error statistics (Packets dropped due to):
      Address spoofing:                  0
      Authentication failed:             0
      Incoming NAT errors:               0
      Invalid zone received packet:      0
      Multiple user authentications:     0
      Multiple incoming NAT:             0
      No parent for a gate:              0
      No one interested in self packets: 0
      No minor session:                  0
      No more sessions:                  0
      No NAT gate:                       0
      No route present:                  0
      No SA for incoming SPI:            0
      No tunnel found:                   0
      No session for a gate:             0
      No zone or NULL zone binding       0
      Policy denied:                     0
      Security association not active:   0
      TCP sequence number out of window: 0
      Syn-attack protection:            0
      User authentication errors:       0
    Protocol inet, MTU: 1500
      Flags: None
      Addresses, Flags: Is-Preferred Is-Primary
        Destination:  203.0.113.1/24, Local: 203.0.113.2, Broadcast: 2.2.2.255
```

show interfaces statistics (View)

Supported Platforms	SRX Series, vSRX
Syntax	show interfaces statistics *interface-name*
Release Information	Command introduced in Junos OS Release 10.1.
Description	Displays the interface input and output statistics for physical and logical interface.
Required Privilege Level	view
Related Documentation	- *Understanding Interfaces*
List of Sample Output	show interfaces statistics on page 215

Sample Output

show interfaces statistics

```
user@host> show interfaces statistics st0.1
Logical interface st0.1 (Index 91) (SNMP ifIndex 268)
     Flags: Point-To-Point SNMP-Traps Encapsulation: Secure-Tunnel
     Input packets : 2743333
     Output packets: 6790470992
     Security: Zone: untrust
     Allowed host-inbound traffic : bootp bfd bgp dns dvmrp igmp ldp msdp nhrp
ospf pgm pim rip router-discovery rsvp sap vrrp dhcp finger ftp tftp ident-reset
 http https ike netconf ping reverse-telnet
    reverse-ssh rlogin rpm rsh snmp snmp-trap ssh telnet traceroute xnm-clear-text
 xnm-ssl lsping ntp sip
     Protocol inet, MTU: 9192
       Addresses, Flags: Is-Preferred Is-Primary
         Destination: 192.167.1.0/30, Local: 192.167.1.1
```

show security flow status

Supported Platforms SRX Series, vSRX

Syntax **show security flow status**

Release Information Command introduced in Junos OS Release 10.2; session distribution mode option added in Junos OS Release 12.1X44-D10; enhanced route scaling mode option added in Junos OS Release 12.1X45-D10. GTP-U distribution option added in Junos OS Release 15.1X49-D40.

Starting with Junos OS Release 15.1X49-D10, SRX5K-MPC3-100G10G (IOC3) and SRX5K-MPC3-40G10G (IOC3) are introduced for SRX5400, SRX5600, and SRX5800 devices that perform hash-based data path packet forwarding to interconnect with all existing IOC and SPC cards using the XL chip (packet-processing chip).

The IOC3 XL chip uses a hash-based method to distribute ingress traffic to a pool of SPUs by default. Selection of hash keys depends on application protocols.

Description Display the flow processing modes and logging status.

Required Privilege Level view

Related Documentation

- *Juniper Networks Devices Processing Overview*

List of Sample Output

show security flow status on page 217
show security flow status (IPsec Performance Acceleration) on page 217
show security flow status (for hash-based datapath forwarding using SRX5K-MPC3-40G10G (IOC3) and SRX5K-MPC3-100G10G (IOC3) on page 218

Output Fields Table 6 on page 216 lists the output fields for the **show security flow status** command. Output fields are listed in the approximate order in which they appear.

Table 6: show security flow status Output Fields

Field Name	Field Description
Flow forwarding mode	Flow processing mode. • Inet forwarding mode • Inet6 forwarding mode • MPLS forwarding mode • ISO forwarding mode • Session distribution mode • Enhanced route scaling mode
Flow trace status	Flow logging status. • Flow tracing status • Flow tracing options

Table 6: show security flow status Output Fields *(continued)*

Field Name	Field Description
flow session distribution	SPU load distribution mode. • RR-based • Hash-based GTP-U distribution • Enabled
Flow packet ordering	packet-ordering mode. • Hardware • Software
Flow ipsec performance acceleration	IPsec VPN performance acceleration status.

Sample Output

show security flow status

```
root> show security flow status
Flow forwarding mode:
Inet forwarding mode: flow based
Inet6 forwarding mode: flow based
MPLS forwarding mode: drop
ISO forwarding mode: drop
Enhanced route scaling mode: Enabled (reboot needed to disable)
Flow trace status
Flow tracing status: on
Flow tracing options: all
Flow session distribution
Distribution mode: Hash-based
GTP-U distribution: Enabled
Flow packet ordering
Ordering mode: Software (reboot needed to change to software)
```

show security flow status (IPsec Performance Acceleration)

```
root> show security flow status
Flow forwarding mode:
    Inet forwarding mode: flow based
    Inet6 forwarding mode: drop
    MPLS forwarding mode: drop
    ISO forwarding mode: drop
Flow trace status
    Flow tracing status: off
Flow session distribution
    Distribution mode: RR-based
    GTP-U distribution: Enabled Flow packet ordering
Ordering mode: Software (reboot needed to change to software)
Flow ipsec performance acceleration: on
```

show security flow status (for hash-based datapath forwarding using SRX5K-MPC3-40G10G (IOC3) and SRX5K-MPC3-100G10G (IOC3)

```
root> show security flow status
node0:
--------------------------------------------------------------------------
  Flow forwarding mode:
    Inet forwarding mode: flow based
    Inet6 forwarding mode: drop
    MPLS forwarding mode: drop
    ISO forwarding mode: drop
  Flow trace status
    Flow tracing status: off
  Flow session distribution
    Distribution mode: Hash-based
    GTP-U distribution: Enabled
  Flow ipsec performance acceleration: off
  Flow packet ordering
    Ordering mode: Hardware

node1:
--------------------------------------------------------------------------
  Flow forwarding mode:
    Inet forwarding mode: flow based
    Inet6 forwarding mode: drop
    MPLS forwarding mode: drop
    ISO forwarding mode: drop
  Flow trace status
    Flow tracing status: off
  Flow session distribution
    Distribution mode: Hash-based
    GTP-U distribution: Enabled
  Flow ipsec performance acceleration: off
  Flow packet ordering
    Ordering mode: Hardware
```

show security ipsec security-associations

Supported Platforms SRX Series, vSRX

Syntax

```
show security ipsec security-associations
brief | detail
family (inet | inet6)
fpc slot-number
index SA-index-number
kmd-instance (all | kmd-instance-name)
pic slot-number>
sa-type shortcut
vpn-name vpn-name <traffic-selector traffic-selector-name>
```

Release Information Command introduced in Junos OS Release 8.5. Support for the **fpc**, **pic**, and **kmd-instance** options added in Junos OS Release 9.3. Support for the **family** option added in Junos OS Release 11.1. Support for the **vpn-name** option added in Junos OS Release 11.4R3. Support for the **traffic-selector** option and traffic selector field added in Junos OS Release 12.1X46-D10. Support for Auto Discovery VPN (ADVPN) added in Junos OS Release 12.3X48-D10. Support for IPsec datapath verification added in Junos OS Release 15.1X49-D70.

Description Display information about the IPsec security associations (SAs).

Options

- none—Display information about all SAs.
- **brief | detail**—(Optional) Display the specified level of output.
- **family**—(Optional) Display SAs by family. This option is used to filter the output.
  - **inet**—IPv4 address family.
  - **inet6**—IPv6 address family.
- **fpc *slot-number***—(Optional) Display information about existing IPsec SAs in this Flexible PIC Concentrator (FPC) slot. This option is used to filter the output.
- **index *SA-index-number***—(Optional) Display detailed information about the specified SA identified by this index number. To obtain a list of all SAs that includes their index numbers, use the command with no options.
- **kmd-instance**—(Optional) Display information about existing IPsec SAs in the key management process (in this case, it is KMD) identified by the FPC *slot-number* and PIC *slot-number*. This option is used to filter the output.
  - **all**—All KMD instances running on the Services Processing Unit (SPU).
  - ***kmd-instance-name***—Name of the KMD instance running on the SPU.
- **pic *slot-number***—(Optional) Display information about existing IPsec SAs in this PIC slot. This option is used to filter the output.
- **sa-type**—(Optional for ADVPN) Type of SA. **shortcut** is the only option for this release.

- **vpn-name** ***vpn-name***—Name of the VPN. If configured, **traffic-selector** ***traffic-selector-name*** can optionally be specified.

Required Privilege Level

view

Related Documentation

- *clear security ipsec security-associations*
- *Example: Configuring a Route-Based VPN Tunnel in a User Logical System*

List of Sample Output

show security ipsec security-associations (IPv4) on page 223
show security ipsec security-associations (IPv6) on page 223
show security ipsec security-associations index 131073 on page 223
show security ipsec security-associations brief on page 224
show security ipsec security-associations detail on page 224
show security ipsec security-associations family inet6 on page 225
show security ipsec security-associations fpc 6 pic 1 kmd-instance all (SRX Series Devices) on page 225
show security ipsec security-associations detail (ADVPN Suggester, Static Tunnel) on page 226
show security ike sa index 222075191 detail on page 226
show security ipsec security-associations detail (ADVPN Partner, Static Tunnel) on page 227
show security ike sa index 788674 detail on page 228
show security ipsec security-associations sa-type shortcut (ADVPN) on page 228
show security ipsec security-associations sa-type shortcut detail (ADVPN) on page 229
show security ipsec security-associations family inet detail on page 229

Output Fields

Table 7 on page 220 lists the output fields for the **show security ipsec security-associations** command. Output fields are listed in the approximate order in which they appear.

Table 7: show security ipsec security-associations

Field Name	Field Description
Total active tunnels	Total number of active IPsec tunnels.
ID	Index number of the SA. You can use this number to get additional information about the SA.
VPN name	IPsec name for VPN.
Gateway	IP address of the remote gateway.
Port	If Network Address Translation (NAT) is used, this value is 4500. Otherwise, it is the standard IKE port, 500.

Table 7: show security ipsec security-associations *(continued)*

Field Name	Field Description
Algorithm	Cryptography used to secure exchanges between peers during the IKE Phase 2 negotiations includes: • An authentication algorithm used to authenticate exchanges between the peers. Options are **hmac-md5-95**, **hmac-sha1-96,** or **ESP**. • An encryption algorithm used to encrypt data traffic. Options are **3des-cbc, aes-128-cbc, aes-192-cbc**, **aes-256-cbc**, or **des-cbc.**
SPI	Security parameter index (SPI) identifier. An SA is uniquely identified by an SPI. Each entry includes the name of the VPN, the remote gateway address, the SPIs for each direction, the encryption and authentication algorithms, and keys. The peer gateways each have two SAs, one resulting from each of the two phases of negotiation: Phase 1 and Phase 2.
Life: sec/kb	The lifetime of the SA, after which it expires, expressed either in seconds or kilobytes.
Sta	State has two options, **Installed** and **Not Installed**. • **Installed**—The SA is installed in the SA database. • **Not Installed**—The SA is not installed in the SA database. For transport mode, the value of State is always **Installed**.
Mon	The Mon field refers to VPN monitoring status. If VPN monitoring is enabled, then this field displays **U** (up) or **D** (down). A hyphen (-) means VPN monitoring is not enabled for this SA. A **V** means that IPsec datapath verification is in progress.
vsys or Virtual-system	The root system.
Tunnel index	Numeric identifier of the specific IPsec tunnel for the SA.
Local gateway	Gateway address of the local system.
Remote gateway	Gateway address of the remote system.
Traffic selector	Name of the traffic selector.
Local identity	Identity of the local peer so that its partner destination gateway can communicate with it. The value is specified as an IP address, fully qualified domain name, e-mail address, or distinguished name (DN).
Remote identity	IP address of the destination peer gateway.
DF-bit	State of the don't fragment bit: **set** or **cleared**.
Policy-name	Name of the applicable policy.

Table 7: show security ipsec security-associations *(continued)*

Field Name	Field Description
Location	**FPC**—Flexible PIC Concentrator (FPC) slot number. **PIC**—PIC slot number. **KMD-Instance**—The name of the KMD instance running on the SPU, identified by FPC *slot-number* and PIC *slot-number*. Currently, 4 KMD instances running on each SPU, and any particular IPsec negotiation is carried out by a single KMD instance.
Tunnel events	Tunnel event and the number of times the event has occurred. See *Tunnel Events* for descriptions of tunnel events and the action you can take.
Direction	Direction of the SA; it can be inbound or outbound.
AUX-SPI	Value of the auxiliary security parameter index(SPI). • When the value is **AH** or **ESP**, **AUX-SPI** is always 0. • When the value is **AH+ESP**, **AUX-SPI** is always a positive integer.
Mode	Mode of the SA: • **transport**—Protects host-to-host connections. • **tunnel—**Protects connections between security gateways.
Type	Type of the SA: • **manual**—Security parameters require no negotiation. They are static and are configured by the user. • **dynamic**—Security parameters are negotiated by the IKE protocol. Dynamic SAs are not supported in transport mode.
State	State of the SA: • **Installed**—The SA is installed in the SA database. • **Not Installed**—The SA is not installed in the SA database. For transport mode, the value of State is always **Installed**.
Protocol	Protocol supported. • Transport mode supports Encapsulation Security Protocol (ESP) and Authentication Header (AH). • Tunnel mode supports ESP and AH. ◦ **Authentication**—Type of authentication used. ◦ **Encryption**—Type of encryption used.
Soft lifetime	The soft lifetime informs the IPsec key management system that the SA is about to expire. Each lifetime of an SA has two display options, hard and soft, one of which must be present for a dynamic SA. This allows the key management system to negotiate a new SA before the hard lifetime expires. • **Expires in seconds**—Number of seconds left until the SA expires.

Table 7: show security ipsec security-associations *(continued)*

Field Name	Field Description
Hard lifetime	The hard lifetime specifies the lifetime of the SA. • **Expires in seconds**—Number of seconds left until the SA expires.
Lifesize Remaining	The lifesize remaining specifies the usage limits in kilobytes. If there is no lifesize specified, it shows unlimited. • **Expires in kilobytes**—Number of kilobytes left until the SA expires.
Anti-replay service	State of the service that prevents packets from being replayed. It can be **Enabled** or **Disabled**.
Replay window size	Configured size of the antireplay service window. It can be 32 or 64 packets. If the replay window size is 0, the antireplay service is disabled. The antireplay window size protects the receiver against replay attacks by rejecting old or duplicate packets.
Bind-interface	The tunnel interface to which the route-based VPN is bound.
Copy-Outer-DSCP	Indicates if copying outer IP header DSCP and ECN to inner IP header is enabled or disabled.

Sample Output

show security ipsec security-associations (IPv4)

```
user@host> show security ipsec security-associations
Total active tunnels: 1
ID    Gateway             Port  Algorithm       SPI      Life:sec/kb  Mon vsys
  131075 192.168.28.241    500   ESP:3des/sha1   86758ff0 6918/ unlim   -   0
  131075 192.168.28.241    500   ESP:3des/sha1   3183ff26 6918/ unlim   -   0
```

show security ipsec security-associations (IPv6)

```
user@host> show security ipsec security-associations
Total active tunnels: 1
ID     Algorithm     SPI      Life:sec/kb  Mon  vsys Port  Gateway
131074 ESP:3des/sha1 14caf1d9 3597/ unlim   -   root 500   2001:db8::1112
131074 ESP:3des/sha1 9a4db486 3597/ unlim   -   root 500   2001:db8::1112
```

show security ipsec security-associations index 131073

```
user@host> show security ipsec security-associations index 131073
  ID: 131073 Virtual-system: root, VPN Name: ike-vpn-chicago
  Local Gateway: 192.168.1.1, Remote Gateway: 192.168.1.2
  Local Identity: ipv4_subnet(any:0,[0..7]=0.0.0.0/0)
  Remote Identity: ipv4_subnet(any:0,[0..7]=0.0.0.0/0)
  Version: IKEv1
  DF-bit: clear
  , Copy-Outer-DSCP Enabled
  Bind-interface: st0.99

  Port: 500, Nego#: 116, Fail#: 0, Def-Del#: 0 Flag: 0x600a29
  Tunnel events:
  Fri Oct 30 2015 15:47:21 -0700: IPSec SA rekey successfully completed (115
times)
  Fri Oct 30 2015 11:38:35 -0700: IKE SA negotiation successfully completed (12
times)
  Mon Oct 26 2015 16:41:07 -0700: IPSec SA negotiation successfully completed (1
 times)
  Mon Oct 26 2015 16:40:56 -0700: Tunnel is ready. Waiting for trigger event or
peer to trigger negotiation (1 times)
  Mon Oct 26 2015 16:40:56 -0700: External interface's address received.
Information updated (1 times)
  Location: FPC 0, PIC 1, KMD-Instance 1
  Direction: inbound, SPI: 81b9fc17, AUX-SPI: 0
  Hard lifetime: Expires in 1774 seconds
  Lifesize Remaining:  Unlimited
  Soft lifetime: Expires in 1151 seconds
  Mode: Tunnel(0 0), Type: dynamic, State: installed
  Protocol: ESP, Authentication: hmac-sha1-96, Encryption: aes-cbc (128 bits)
  Anti-replay service: counter-based enabled

  , Replay window size: 64
  Location: FPC 0, PIC 1, KMD-Instance 1
  Direction: outbound, SPI: 727f629d, AUX-SPI: 0
  Hard lifetime: Expires in 1774 seconds
  Lifesize Remaining:  Unlimited
  Soft lifetime: Expires in 1151 seconds
  Mode: Tunnel(0 0), Type: dynamic, State: installed
  Protocol: ESP, Authentication: hmac-sha1-96, Encryption: aes-cbc (128 bits)
  Anti-replay service: counter-based enabled

  , Replay window size: 64
```

show security ipsec security-associations brief

```
user@host> show security ipsec security-associations brief
Total active tunnels: 2
ID     Gateway     Port Algorithm     SPI      Life:sec/kb Mon vsys
<16384 192.168.1.1 500  ESP:3des/sha1 af88baa  28795/unlim D   0
>16384 192.168.1.1 500  ESP:3des/sha1 f4e3e5f4 28795/unlim D   0
```

show security ipsec security-associations detail

```
user@host> show security ipsec security-associations detail
ID: 131073 Virtual-system: root, VPN Name: ike-vpn-chicago
Local Gateway: 192.168.1.1, Remote Gateway: 192.168.1.2
Local Identity: ipv4_subnet(any:0,[0..7]=0.0.0.0/0)
Remote Identity: ipv4_subnet(any:0,[0..7]=0.0.0.0/0)
Version: IKEv1
DF-bit: clear
, Copy-Outer-DSCP Enabled
Bind-interface: st0.99

Port: 500, Nego#: 8, Fail#: 0, Def-Del#: 0 Flag: 0x600a29
Tunnel events:
Mon Oct 26 2015 22:27:50 -0700: IPSec SA rekey successfully completed (7 times)
Mon Oct 26 2015 16:41:07 -0700: IPSec SA negotiation successfully completed (1
times)
Mon Oct 26 2015 16:41:07 -0700: IKE SA negotiation successfully completed (1
times)
Mon Oct 26 2015 16:40:56 -0700: Tunnel is ready. Waiting for trigger event or
peer to trigger negotiation (1 times)
```

```
Mon Oct 26 2015 16:40:56 -0700: External interface's address received. Information
 updated (1 times)
Location: FPC 0, PIC 1, KMD-Instance 1
Direction: inbound, SPI: 81ed9998, AUX-SPI: 0
Hard lifetime: Expires in 2296 seconds
Lifesize Remaining:  Unlimited
Soft lifetime: Expires in 1688 seconds
Mode: Tunnel(0 0), Type: dynamic, State: installed
Protocol: ESP, Authentication: hmac-sha1-96, Encryption: aes-cbc (128 bits)
Anti-replay service: counter-based enabled

, Replay window size: 64
Location: FPC 0, PIC 1, KMD-Instance 1
Direction: outbound, SPI: 80565248, AUX-SPI: 0
Hard lifetime: Expires in 2296 seconds
Lifesize Remaining:  Unlimited
Soft lifetime: Expires in 1688 seconds
Mode: Tunnel(0 0), Type: dynamic, State: installed
Protocol: ESP, Authentication: hmac-sha1-96, Encryption: aes-cbc (128 bits)
Anti-replay service: counter-based enabled

, Replay window size: 64
```

show security ipsec security-associations family inet6

```
user@host> show security ipsec security-associations family inet6
  Virtual-system: root
  Local Gateway: 2001:db8:1212::1111, Remote Gateway: 2001:db8:1212::1112
  Local Identity: ipv4_subnet(any:0,[0..7]=0.0.0.0/0)
  Remote Identity: ipv4_subnet(any:0,[0..7]=0.0.0.0/0)
    DF-bit: clear
    Direction: inbound, SPI: 14caf1d9, AUX-SPI: 0
                              , VPN Monitoring: -
    Hard lifetime: Expires in 3440 seconds
    Lifesize Remaining:  Unlimited
    Soft lifetime: Expires in 2813 seconds
    Mode: tunnel, Type: dynamic, State: installed
    Protocol: ESP, Authentication: hmac-sha1-96, Encryption: 3des-cbc
    Anti-replay service: counter-based enabled, Replay window size: 64

    Direction: outbound, SPI: 9a4db486, AUX-SPI: 0
                              , VPN Monitoring: -
    Hard lifetime: Expires in 3440 seconds
    Lifesize Remaining:  Unlimited
    Soft lifetime: Expires in 2813 seconds
    Mode: tunnel, Type: dynamic, State: installed
    Protocol: ESP, Authentication: hmac-sha1-96, Encryption: 3des-cbc
    Anti-replay service: counter-based enabled, Replay window size: 64
```

show security ipsec security-associations fpc 6 pic 1 kmd-instance all (SRX Series Devices)

```
user@host> show security ipsec security-associations fpc 6 pic 1 kmd-instance all
  Total active tunnels: 1

ID    Gateway          Port  Algorithm       SPI      Life:sec/kb  Mon vsys

<2    192.168.1.2      500   ESP:3des/sha1   67a7d25d 28280/unlim   -   0

>2    192.168.1.2      500   ESP:3des/sha1   a23cbcdc 28280/unlim   -   0
```

show security ipsec security-associations detail (ADVPN Suggester, Static Tunnel)

```
user@host> show security ipsec security-associations detail
ID: 70516737 Virtual-system: root, VPN Name: ZTH_HUB_VPN
  Local Gateway: 192.168.1.1, Remote Gateway: 192.168.1.2
  Local Identity: ipv4_subnet(any:0,[0..7]=0.0.0.0/0)
  Remote Identity: ipv4_subnet(any:0,[0..7]=0.0.0.0/0)
  Version: IKEv2
  DF-bit: clear
  Bind-interface: st0.1

  Port: 500, Nego#: 5, Fail#: 0, Def-Del#: 0 Flag: 0x608a29
  Tunnel events:
  Tue Nov 03 2015 01:24:27 -0800: IPSec SA negotiation successfully completed (1
 times)
  Tue Nov 03 2015 01:24:27 -0800: IKE SA negotiation successfully completed (4
times)
  Tue Nov 03 2015 01:23:38 -0800: User cleared IPSec SA from CLI (1 times)
  Tue Nov 03 2015 01:21:32 -0800: IPSec SA negotiation successfully completed (1
 times)
  Tue Nov 03 2015 01:21:31 -0800: IPSec SA delete payload received from peer,
corresponding IPSec SAs cleared (1 times)
  Tue Nov 03 2015 01:21:27 -0800: IPSec SA negotiation successfully completed (1
 times)
  Tue Nov 03 2015 01:21:13 -0800: Tunnel configuration changed. Corresponding
IKE/IPSec SAs are deleted (1 times)
  Tue Nov 03 2015 01:19:27 -0800: IPSec SA negotiation successfully completed (1
 times)
  Tue Nov 03 2015 01:19:27 -0800: Tunnel is ready. Waiting for trigger event or
peer to trigger negotiation (1 times)
  Location: FPC 0, PIC 3, KMD-Instance 2
  Direction: inbound, SPI: 43de5d65, AUX-SPI: 0
  Hard lifetime: Expires in 1335 seconds
  Lifesize Remaining:  Unlimited
  Soft lifetime: Expires in 996 seconds
  Mode: Tunnel(0 0), Type: dynamic, State: installed
  Protocol: ESP, Authentication: hmac-sha1-96, Encryption: aes-cbc (192 bits)
  Anti-replay service: counter-based enabled

  , Replay window size: 64
  Location: FPC 0, PIC 3, KMD-Instance 2
  Direction: outbound, SPI: 5b6e157c, AUX-SPI: 0
  Hard lifetime: Expires in 1335 seconds
  Lifesize Remaining:  Unlimited
  Soft lifetime: Expires in 996 seconds
  Mode: Tunnel(0 0), Type: dynamic, State: installed
  Protocol: ESP, Authentication: hmac-sha1-96, Encryption: aes-cbc (192 bits)
  Anti-replay service: counter-based enabled

  , Replay window size: 64
```

show security ike sa index 222075191 detail

```
user@host> show security ike sa index 222075191 detail
node0:
--------------------------------------------------------------------------
IKE peer 192.168.1.2, Index 222075191, Gateway Name: ZTH_HUB_GW
  Location: FPC 0, PIC 3, KMD-Instance 2
  Auto Discovery VPN:
   Type: Static, Local Capability: Suggester, Peer Capability: Partner
   Suggester Shortcut Suggestions Statistics:
     Suggestions sent    :    2
     Suggestions accepted:    4
     Suggestions declined:    1
  Role: Responder, State: UP
  Initiator cookie: 7b996b4c310d2424, Responder cookie: 5724c5882a212157
  Exchange type: IKEv2, Authentication method: RSA-signatures
  Local: 192.168.1.1:500, Remote: 192.168.1.2:500
  Lifetime: Expires in 828 seconds
  Peer ike-id: C=US, DC=example, ST=CA, L=Sunnyvale, O=example, OU=engineering,
CN=cssvk36-d
  Xauth user-name: not available
  Xauth assigned IP: 0.0.0.0
  Algorithms:
   Authentication        : hmac-sha1-96
   Encryption            : aes256-cbc
   Pseudo random function: hmac-sha1
   Diffie-Hellman group  : DH-group-5
  Traffic statistics:
   Input  bytes  :                20474
   Output bytes  :                21091
   Input  packets:                  237
   Output packets:                  237
  IPSec security associations: 2 created, 0 deleted
  Phase 2 negotiations in progress: 1

    Negotiation type: Quick mode, Role: Responder, Message ID: 0
    Local: 192.168.1.1:500, Remote: 192.168.1.2:500
    Local identity: C=US, DC=example, ST=CA, L=Sunnyvale, O=example,
OU=engineering, CN=host1
    Remote identity: C=US, DC=example, ST=CA, L=Sunnyvale, O=example,
OU=engineering, CN=host2
    Flags: IKE SA is created
```

show security ipsec security-associations detail (ADVPN Partner, Static Tunnel)

```
user@host> show security ipsec security-associations detail
ID: 67108872 Virtual-system: root, VPN Name: ZTH_SPOKE_VPN
  Local Gateway: 192.168.1.2, Remote Gateway: 192.168.1.1
  Local Identity: ipv4_subnet(any:0,[0..7]=0.0.0.0/0)
  Remote Identity: ipv4_subnet(any:0,[0..7]=0.0.0.0/0)
  Version: IKEv2
  DF-bit: clear, Bind-interface: st0.1
  Port: 500, Nego#: 0, Fail#: 0, Def-Del#: 0 Flag: 0x8608a29
  Tunnel events:
  Tue Nov 03 2015 01:24:26 -0800: IPSec SA negotiation successfully completed (1
 times)
  Tue Nov 03 2015 01:24:26 -0800: IKE SA negotiation successfully completed (4
times)
  Tue Nov 03 2015 01:23:37 -0800: IPSec SA delete payload received from peer,
corresponding IPSec SAs cleared (1 times)
  Tue Nov 03 2015 01:21:31 -0800: IPSec SA negotiation successfully completed (1
 times)
  Tue Nov 03 2015 01:21:31 -0800: Tunnel is ready. Waiting for trigger event or
peer to trigger negotiation (1 times)
  Tue Nov 03 2015 01:18:26 -0800: Key pair not found for configured local
certificate. Negotiation failed (1 times)
  Tue Nov 03 2015 01:18:13 -0800: CA certificate for configured local certificate
 not found. Negotiation not initiated/successful (1 times)
  Direction: inbound, SPI: 5b6e157c, AUX-SPI: 0
  Hard lifetime: Expires in 941 seconds
  Lifesize Remaining:  Unlimited
  Soft lifetime: Expires in 556 seconds
  Mode: Tunnel(0 0), Type: dynamic, State: installed
  Protocol: ESP, Authentication: hmac-sha1-96, Encryption: aes-cbc (192 bits)
  Anti-replay service: counter-based enabled, Replay window size: 64
  Direction: outbound, SPI: 43de5d65, AUX-SPI: 0
  Hard lifetime: Expires in 941 seconds
  Lifesize Remaining:  Unlimited
  Soft lifetime: Expires in 556 seconds
  Mode: Tunnel(0 0), Type: dynamic, State: installed
  Protocol: ESP, Authentication: hmac-sha1-96, Encryption: aes-cbc (192 bits)
  Anti-replay service: counter-based enabled, Replay window size: 64
```

show security ike sa index 788674 detail

```
user@host> show security ike sa index 788674 detail
IKE peer 192.168.1.1, Index 788674, Gateway Name: ZTH_SPOKE_GW
  Auto Discovery VPN:
   Type: Static, Local Capability: Partner, Peer Capability: Suggester
   Partner Shortcut Suggestions Statistics:
     Suggestions received:    2
     Suggestions accepted:    2
     Suggestions declined:    0
  Role: Initiator, State: UP
  Initiator cookie: 7b996b4c310d2424, Responder cookie: 5724c5882a212157
  Exchange type: IKEv2, Authentication method: RSA-signatures
  Local: 192.168.1.2:500, Remote: 192.168.1.1:500
  Lifetime: Expires in 734 seconds
  Peer ike-id: C=US, DC=example, ST=CA, L=Sunnyvale, O=example, OU=engineering,
CN=test
  Xauth user-name: not available
  Xauth assigned IP: 0.0.0.0
  Algorithms:
   Authentication        : hmac-sha1-96
   Encryption            : aes256-cbc
   Pseudo random function: hmac-sha1
   Diffie-Hellman group  : DH-group-5
  Traffic statistics:
   Input  bytes  :                22535
   Output bytes  :                21918
   Input  packets:                  256
   Output packets:                  256
  IPSec security associations: 2 created, 0 deleted
  Phase 2 negotiations in progress: 1

    Negotiation type: Quick mode, Role: Initiator, Message ID: 0
    Local: 192.168.1.2:500, Remote: 192.168.1.1:500
    Local identity: C=US, DC=example, ST=CA, L=Sunnyvale, O=example,
OU=engineering, CN=host1
    Remote identity: C=US, DC=example, ST=CA, L=Sunnyvale, O=example,
OU=engineering, CN=host2
    Flags: IKE SA is created
```

show security ipsec security-associations sa-type shortcut (ADVPN)

```
user@host> show security ipsec security-associations sa-type shortcut
Total active tunnels: 1
ID    Algorithm       SPI      Life:sec/kb  Mon lsys Port  Gateway
<268173318 ESP:aes-cbc-256/sha1 6f164ee0 3580/ unlim - root 500 192.168.0.111

>268173318 ESP:aes-cbc-256/sha1 e6f29cb0 3580/ unlim - root 500 192.168.0.111
```

show security ipsec security-associations sa-type shortcut detail (ADVPN)

```
user@host> show security ipsec security-associations sa-type shortcut detail
node0:
--------------------------------------------------------------------------

ID: 67108874 Virtual-system: root, VPN Name: ZTH_SPOKE_VPN
  Local Gateway: 192.168.1.2, Remote Gateway: 192.168.1.2
  Local Identity: ipv4_subnet(any:0,[0..7]=0.0.0.0/0)
  Remote Identity: ipv4_subnet(any:0,[0..7]=0.0.0.0/0)
  Auto Discovery VPN:
    Type: Shortcut, Shortcut Role: Initiator
  Version: IKEv2
  DF-bit: clear, Bind-interface: st0.1
  Port: 4500, Nego#: 0, Fail#: 0, Def-Del#: 0 Flag: 0x40608a29
  Tunnel events:
    Tue Nov 03 2015 01:47:26 -0800: IPSec SA negotiation successfully completed
(1 times)
    Tue Nov 03 2015 01:47:26 -0800: Tunnel is ready. Waiting for trigger event
or peer to trigger negotiation (1 times)
    Tue Nov 03 2015 01:47:26 -0800: IKE SA negotiation successfully completed (1
 times)
  Direction: inbound, SPI: b7a5518, AUX-SPI: 0
    Hard lifetime: Expires in 1766 seconds
    Lifesize Remaining:  Unlimited
    Soft lifetime: Expires in 1381 seconds
    Mode: Tunnel(0 0), Type: dynamic, State: installed
    Protocol: ESP, Authentication: hmac-sha1-96, Encryption: aes-cbc (192 bits)
    Anti-replay service: counter-based enabled, Replay window size: 64
  Direction: outbound, SPI: b7e0268, AUX-SPI: 0
    Hard lifetime: Expires in 1766 seconds
    Lifesize Remaining:  Unlimited
    Soft lifetime: Expires in 1381 seconds
    Mode: Tunnel(0 0), Type: dynamic, State: installed
    Protocol: ESP, Authentication: hmac-sha1-96, Encryption: aes-cbc (192 bits)
    Anti-replay service: counter-based enabled, Replay window size: 64
```

show security ipsec security-associations family inet detail

```
user@host> show security ipsec security-associations family inet detail
ID: 131073 Virtual-system: root, VPN Name: ike-vpn-chicago
  Local Gateway: 192.168.1.1, Remote Gateway: 192.168.1.2
  Local Identity: ipv4_subnet(any:0,[0..7]=0.0.0.0/0)
  Remote Identity: ipv4_subnet(any:0,[0..7]=0.0.0.0/0)
  Version: IKEv1
  DF-bit: clear
  , Copy-Outer-DSCP Enabled
  Bind-interface: st0.99

  Port: 500, Nego#: 116, Fail#: 0, Def-Del#: 0 Flag: 0x600a29
  Tunnel events:
  Fri Oct 30 2015 15:47:21 -0700: IPSec SA rekey successfully completed (115
times)
  Fri Oct 30 2015 11:38:35 -0700: IKE SA negotiation successfully completed (12
times)
  Mon Oct 26 2015 16:41:07 -0700: IPSec SA negotiation successfully completed (1
 times)
  Mon Oct 26 2015 16:40:56 -0700: Tunnel is ready. Waiting for trigger event or
peer to trigger negotiation (1 times)
  Mon Oct 26 2015 16:40:56 -0700: External interface's address received.
Information updated (1 times)
```

```
Location: FPC 0, PIC 1, KMD-Instance 1
Direction: inbound, SPI: 81b9fc17, AUX-SPI: 0
Hard lifetime: Expires in 1713 seconds
Lifesize Remaining:  Unlimited
Soft lifetime: Expires in 1090 seconds
Mode: Tunnel(0 0), Type: dynamic, State: installed
Protocol: ESP, Authentication: hmac-sha1-96, Encryption: aes-cbc (128 bits)
Anti-replay service: counter-based enabled

, Replay window size: 64
Location: FPC 0, PIC 1, KMD-Instance 1
Direction: outbound, SPI: 727f629d, AUX-SPI: 0
Hard lifetime: Expires in 1713 seconds
Lifesize Remaining:  Unlimited
Soft lifetime: Expires in 1090 seconds
Mode: Tunnel(0 0), Type: dynamic, State: installed
Protocol: ESP, Authentication: hmac-sha1-96, Encryption: aes-cbc (128 bits)
Anti-replay service: counter-based enabled

, Replay window size: 64
```

show security ipsec statistics

Supported Platforms SRX Series, vSRX

Syntax

```
show security ipsec statistics
<fpc  slot-number >
<index  SA-index-number  >
<kmd-instance  kmd-instance-name  >
pic  slot-number
```

Release Information Command introduced in Junos OS Release 8.5. **fpc** and **pic** options added in Junos OS Release 9.3. **kmd-instance** option added in Junos OS Release 10.4.

Description Display standard IPsec statistics.

Options

- none—Display statistics about all IPsec security associations (SAs).
- **fpc** ***slot-number*** —Specific to SRX Series devices. Display statistics about existing IPsec SAs in this Flexible PIC Concentrator (FPC) slot. This option is used to filter the output.
- **index** ***SA-index-number*** —(Optional) Display statistics for the SA with this index number.
- **kmd-instance** ***kmd-instance-name*** —Specific to SRX Series devices. Display information about existing IKE SAs in the key management process (the daemon, which in this case is KMD) identified by FPC *slot-number* and PIC *slot-number*. This option is used to filter the output.
  - **all**—All KMD instances running on the Services Processing Unit (SPU).
  - ***kmd-instance-name***—Name of the KMD instance running on the SPU.
- **pic** ***slot-number*** —Specific to SRX Series devices. Display statistics about existing IPsec SAs in this PIC slot. This option is used to filter the output.

Required Privilege Level view

Related Documentation

- *clear security ipsec statistics*

List of Sample Output

show security ipsec statistics on page 232
show security ipsec statistics index 5 on page 233
show security ipsec statistics fpc 6 pic 1 (SRX Series devices) on page 233

Output Fields Table 8 on page 231 lists the output fields for the **show security ipsec statistics** command. Output fields are listed in the approximate order in which they appear.

Table 8: show security ipsec statistics Output Fields

Field Name	Field Description
Virtual-system	The root system.

Table 8: show security ipsec statistics Output Fields *(continued)*

Field Name	Field Description
ESP Statistics	- **Encrypted bytes**—Total number of bytes encrypted by the local system across the IPsec tunnel. - **Decrypted bytes**—Total number of bytes decrypted by the local system across the IPsec tunnel. - **Encrypted packets**—Total number of packets encrypted by the local system across the IPsec tunnel. - **Decrypted packets**—Total number of packets decrypted by the local system across the IPsec tunnel.
AH Statistics	- **Input bytes**—Total number of bytes received by the local system across the IPsec tunnel. - **Output bytes**—Total number of bytes transmitted by the local system across the IPsec tunnel. - **Input packets**—Total number of packets received by the local system across the IPsec tunnel. - **Output packets**—Total number of packets transmitted by the local system across the IPsec tunnel.
Errors	- **AH authentication failures**—Total number of authentication header (AH) failures. An AH failure occurs when there is a mismatch of the authentication header in a packet transmitted across an IPsec tunnel. - **Replay errors**—Total number of replay errors. A replay error is generated when a duplicate packet is received within the replay window. - **ESP authentication failures**—Total number of Encapsulation Security Payload (ESP) failures. An ESP failure occurs when there is an authentication mismatch in ESP packets. - **ESP decryption failures**—total number of ESP decryption errors. - **Bad headers**—Total number of invalid headers detected. - **Bad trailers**—Total number of invalid trailers detected.

Sample Output

show security ipsec statistics

```
user@host> show security ipsec statistics
Virtual-system: Root
ESP Statistics:
  Encrypted bytes:              0
  Decrypted bytes:              0
  Encrypted packets:            0
  Decrypted packets:            0
AH Statistics:
  Input bytes:                  0
  Output bytes:                 0
  Input packets:                0
  Output packets:               0
Errors:
  AH authentication failures: 0, Replay errors: 0
  ESP authentication failures: 0, ESP decryption failures: 0
  Bad headers: 0, Bad trailers: 0
```

Sample Output

show security ipsec statistics index 5

```
user@host> show security ipsec statistics index 5
Virtual-system: Root
SA index: 5
ESP Statistics:
  Encrypted bytes:              0
  Decrypted bytes:              0
  Encrypted packets:            0
  Decrypted packets:            0
AH Statistics:
  Input bytes:                  0
  Output bytes:                 0
  Input packets:                0
  Output packets:               0
Errors:
  AH authentication failures: 0, Replay errors: 0
  ESP authentication failures: 0, ESP decryption failures: 0
  Bad headers: 0, Bad trailers: 0
```

Sample Output

show security ipsec statistics fpc 6 pic 1 (SRX Series devices)

```
user@host> show security ipsec statistics fpc 6 pic 1
ESP Statistics:
Encrypted bytes:          536408
Decrypted bytes:          696696
Encrypted packets:          1246
Decrypted packets:           888
AH Statistics:
Input bytes:                   0
Output bytes:                  0
Input packets:                 0
Output packets:                0
Errors:
AH authentication failures: 0, Replay errors: 0
ESP authentication failures: 0, ESP decryption failures: 0
Bad headers: 0, Bad trailers: 0
```