JUNIPER NETWORKS

Junos® OS

Subscriber Interfaces Over MPLS Pseudowires Feature Guide

Release
13.2

Published: 2013-07-31

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

Junos® OS Subscriber Interfaces Over MPLS Pseudowires Feature Guide
13.2

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

YEAR 2000 NOTICE

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

END USER LICENSE AGREEMENT

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

Table of Contents

List of Figures

List of Tables

About the Documentation

Documentation and Release Notes

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

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

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

Supported Platforms

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

- MX Series

Using the Examples in This Manual

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

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

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

Merging a Full Example

To merge a full example, follow these steps:

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

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

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

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

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

Merging a Snippet

To merge a snippet, follow these steps:

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

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

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

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

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

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

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

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

Documentation Conventions

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

Table 1: Notice Icons

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

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

Table 2: Text and Syntax Conventions

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

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

Convention	Description	Examples
Italic text like this	• Introduces or emphasizes important new terms. • Identifies book names. • Identifies RFC and Internet draft titles.	• A policy *term* is a named structure that defines match conditions and actions. • *Junos OS System Basics Configuration Guide* • RFC 1997, *BGP Communities Attribute*
Italic text like this	Represents variables (options for which you substitute a value) in commands or configuration statements.	Configure the machine's domain name: [edit] root@# **set system domain-name** *domain-name*
Text like this	Represents names of configuration statements, commands, files, and directories; configuration hierarchy levels; or labels on routing platform components.	• To configure a stub area, include the **stub** statement at the **[edit protocols ospf area area-id]** hierarchy level. • The console port is labeled **CONSOLE**.
< > (angle brackets)	Enclose optional keywords or variables.	**stub <default-metric** ***metric*****>;**
| (pipe symbol)	Indicates a choice between the mutually exclusive keywords or variables on either side of the symbol. The set of choices is often enclosed in parentheses for clarity.	**broadcast | multicast** **(*string1* | *string2* | *string3*)**
# (pound sign)	Indicates a comment specified on the same line as the configuration statement to which it applies.	**rsvp { # Required for dynamic MPLS only**
[ ] (square brackets)	Enclose a variable for which you can substitute one or more values.	**community name members [** ***community-ids*** **]**

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

; (semicolon)	Identifies a leaf statement at a configuration hierarchy level.	
GUI Conventions		
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 send your comments to techpubs-comments@juniper.net, or fill out the documentation feedback form at https://www.juniper.net/cgi-bin/docbugreport/ . If you are using e-mail, be sure to include the following information with your comments:

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

Requesting Technical Support

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

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

Self-Help Online Tools and Resources

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

- Find CSC offerings: http://www.juniper.net/customers/support/
- Search for known bugs: http://www2.juniper.net/kb/
- Find product documentation: http://www.juniper.net/techpubs/
- Find solutions and answer questions using our Knowledge Base: http://kb.juniper.net/
- Download the latest versions of software and review release notes: http://www.juniper.net/customers/csc/software/
- Search technical bulletins for relevant hardware and software notifications: https://www.juniper.net/alerts/

- Join and participate in the Juniper Networks Community Forum: http://www.juniper.net/company/communities/
- Open a case online in the CSC Case Management tool: http://www.juniper.net/cm/

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

Opening a Case with JTAC

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

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

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

PART 1

Overview

CHAPTER 1

Subscriber Interfaces Over MPLS Pseudowires in Subscriber Access Networks

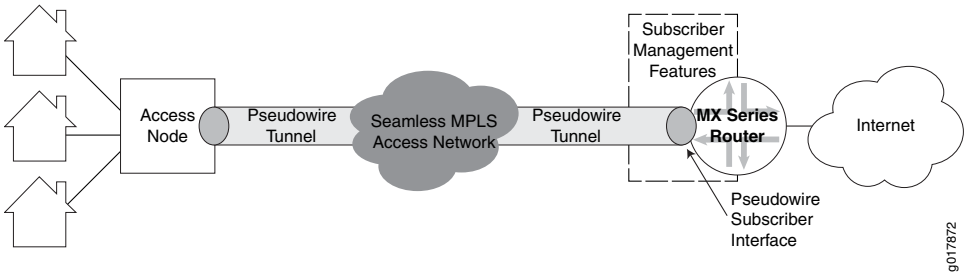
Pseudowire Subscriber Logical Interfaces Overview

Subscriber management supports the creation of subscriber interfaces over point-to-point MPLS pseudowires. The pseudowire subscriber interface capability enables service providers to extend an MPLS domain from the access-aggregation network to the service edge, where subscriber management is performed. Service providers can take advantage of MPLS capabilities such as failover, rerouting, and uniform MPLS label provisioning, while using a single pseudowire to service a large number of DHCP and PPPoE subscribers in the service network.

The pseudowire is a tunnel that is either an MPLS-based Layer 2 VPN or Layer 2 circuit. The pseudowire tunnel transports Ethernet encapsulated traffic from an access node (for example, a DSLAM or other aggregation device) to the MX Series router that hosts the subscriber management services. The termination of the pseudowire tunnel on the MX Series router is similar to a physical Ethernet termination, and is the point at which subscriber management functions are performed. A service provider can configure multiple pseudowires on a per-DSLAM basis and then provision support for a large number of subscribers on a specific pseudowire. Figure 1 on page 4 shows an MPLS network that provides subscriber management support.

At the access node end of the pseudowire, the subscriber traffic can be groomed into the pseudowire in a variety of ways, limited only by the number and types of interfaces that can be stacked on the pseudowire. You specify an anchor point, which identifies the logical tunnel interface that terminates the pseudowire tunnel at the access node.

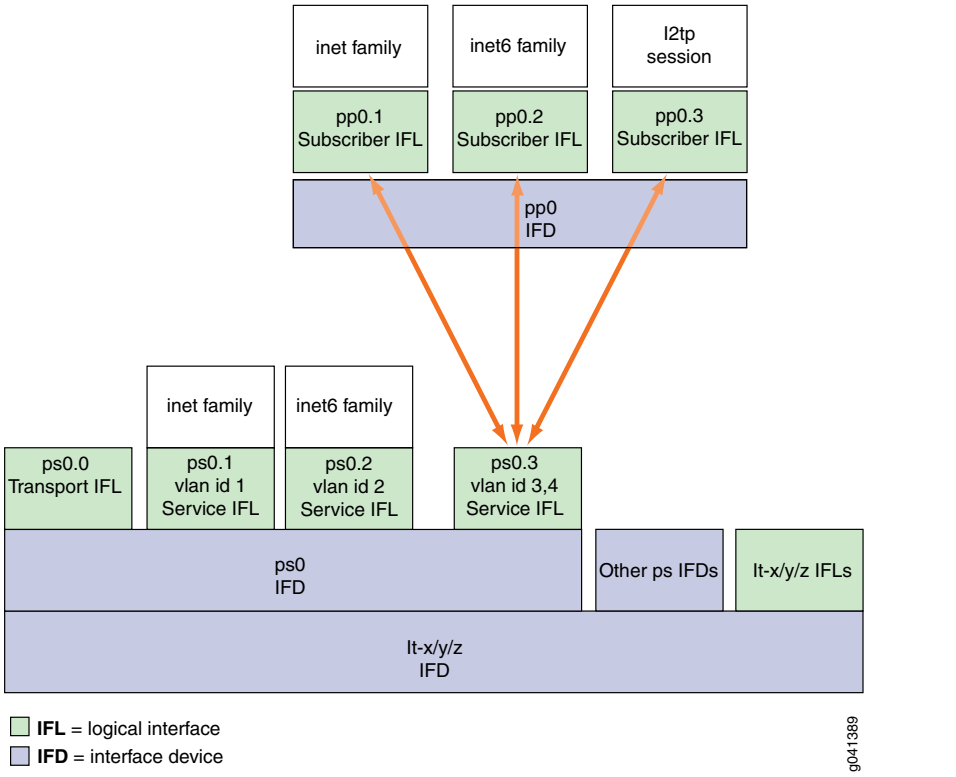
Figure 1: MPLS Access Network with Subscriber Management Support



NOTE: Subscriber interfaces over MPLS pseudowires are supported on MX Series routers with MPCs.

Figure 2 on page 4 shows the protocol stack for a pseudowire subscriber logical interface. The pseudowire is a virtual device that is stacked above the logical tunnel anchor point on the physical interface (the IFD), and supports a circuit-oriented Layer 2 protocol (either Layer 2 VPN or Layer 2 circuit). The Layer 2 protocol provides the transport and service logical interfaces, and supports the protocol family (IPv4, IPv6, or PPPoE).

Figure 2: Pseudowire Subscriber Interface Protocol Stack



The pseudowire configuration is transparent to the subscriber management applications and has no impact on the packet payloads that are used for subscriber management.

Subscriber applications such as DHCP and PPPoE can be stacked over Layer 2 similar to the way in which they are stacked over a physical interface.

Related Documentation

- Configuring a Pseudowire Subscriber Logical Interface on page 13
- Hierarchical CoS on MPLS Pseudowire Subscriber Interfaces Overview on page 5

Hierarchical CoS on MPLS Pseudowire Subscriber Interfaces Overview

Junos OS supports two aspects of CoS for MPLS pseudowire subscriber interfaces. You can apply CoS rewrite rules and behavior aggregate (BA) classifiers to MPLS pseudowire subscriber interfaces. In addition, CoS performs egress hierarchical shaping towards the subscriber on MPLS pseudowire subscriber interfaces.

Hierarchical CoS enables you to apply traffic scheduling and queuing parameters and packet transmission scheduling parameters to an individual subscriber interface rather than to all interfaces configured on the port. Hierarchical CoS is supported on MX Series routers with either EQ DPCs or MPC/MICs installed.

On Juniper Networks MX Series routers, MPC/MIC and EQ DPC interfaces support a four-level CoS scheduling hierarchy that, when fully configured, consists of the physical interface (level 1), the interface set or the underlying interface (level 2), one or more logical interfaces (level 3), and one or more queues (level 4). Although all CoS scheduling hierarchies are four-level, level 1 is always the physical interface and level 4 is always the queue. Hierarchical scheduling configurations consist of the type of interfaces you configure; for example, a logical interface or an interface set and where those interfaces reside in the scheduling hierarchy, either level 2 or level 3. Because many hierarchical scheduling configurations are possible, we use the terms *two-level hierarchical scheduling* and *three-level hierarchical scheduling* in this discussion.

Related Documentation

- Pseudowire Subscriber Logical Interfaces Overview on page 3
- Configuring a Pseudowire Subscriber Logical Interface on page 13
- *Understanding Two-Level and Three-level Hierarchical CoS for Subscriber Interfaces on MX Series Routers*
- CoS Two-Level Hierarchical Scheduling on MPLS Pseudowire Subscriber Interfaces on page 5
- CoS Three-Level Hierarchical Scheduling on MPLS Pseudowire Subscriber Interfaces on page 7
- CoS Configuration Overview for MPLS Pseudowire Subscriber Interfaces on page 21
- **hierarchical-scheduler (Subscriber Interfaces on MX Series Routers) on page 36**

CoS Two-Level Hierarchical Scheduling on MPLS Pseudowire Subscriber Interfaces

Two-level hierarchical scheduling limits the number of hierarchical levels in the scheduling hierarchy to two. In a two-level scheduling hierarchy, all logical interfaces and interface

sets share a single level 2 node.Table 3 on page 6 summarizes the interface hierarchy and the CoS scheduler node levels for two-level hierarchical scheduling.

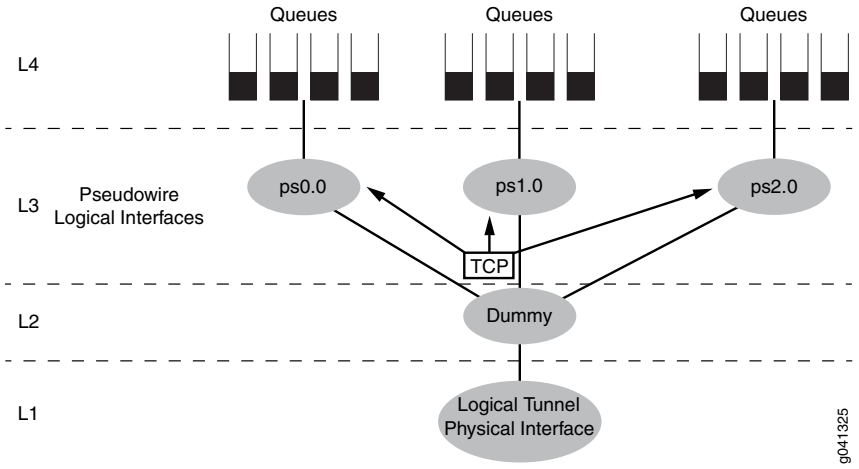
Table 3: Two-Level Hierarchical Scheduling-Interface Hierarchy Versus Scheduling Nodes

Level 1	Level 2	Level 3	Level 4
Physical interface	–	Pseudowire transport logical interface	One or more queues
Physical interface	–	Interface set	One or more queues
Physical interface	–	Pseudowire service logical interface	One or more queues

You use the two-level hierarchical scheduling when you have many pseudowires but you do not require shaping specific to the subscriber logical interface. For example, when your configuration is one subscriber per pseudowire interface.

Figure 3 on page 6 shows a two-level hierarchical scheduling configuration for the MPLS pseudowires. In this configuration, level 1 is the physical interface used for the logical tunnel anchor node. All of the pseudowire transport interfaces share a single level 2 node. The level 3 nodes are the pseudowire transport logical interfaces (ps0.0, ps1.0, and ps2.0). In this configuration, interface sets are not configured and only the logical interfaces have traffic control profiles.

Figure 3: MPLS Pseudowire Subscriber Interface Two-Level Scheduler Configuration



Two-level hierarchical scheduling has up to eight class of service queues. For this configuration, include the **maximum-hierarchy-levels 2** option under the **[edit interfaces *interface-name* hierarchical-scheduler]** statement at the physical interface for the anchor logical tunnel.

NOTE: You cannot configure shaping policies on both the pseudowire logical interfaces and the subscriber logical interfaces over the same pseudowire. If a traffic-control profile is configured on a pseudowire logical interface, and CoS policies are configured on the subscriber logical interface over another pseudowire, all of the logical interfaces are at level 3 and act as peers.

Related Documentation

- Pseudowire Subscriber Logical Interfaces Overview on page 3
- Configuring a Pseudowire Subscriber Logical Interface on page 13
- *Understanding Two-Level and Three-level Hierarchical CoS for Subscriber Interfaces on MX Series Routers*
- Hierarchical CoS on MPLS Pseudowire Subscriber Interfaces Overview on page 5
- CoS Three-Level Hierarchical Scheduling on MPLS Pseudowire Subscriber Interfaces on page 7
- CoS Configuration Overview for MPLS Pseudowire Subscriber Interfaces on page 21
- Configuring CoS Two-Level Hierarchical Scheduling for MPLS Pseudowire Subscriber Interfaces on page 22
- **hierarchical-scheduler (Subscriber Interfaces on MX Series Routers) on page 36**

CoS Three-Level Hierarchical Scheduling on MPLS Pseudowire Subscriber Interfaces

In three-level hierarchical scheduling, the CoS scheduler nodes at level 1, level 2, and level 3 form a scheduling hierarchy. You can configure many different three-level scheduling hierarchies, depending on the location of the interface set and the use of underlying interfaces. In all variations, the physical interface on which the logical tunnel resides is a level 1 CoS scheduler node and the queues reside at level 4. Three-level scheduling hierarchies can have up to eight class of service queues.
Table 4 on page 7summarizes the most common three-level hierarchical scheduling configurations and shows the interface hierarchy and CoS scheduler nodes.

Table 4: Three-Level Hierarchical Scheduling-Interface Hierarchy Versus CoS Scheduling Node Levels

Level 1	Level 2	Level 3	Level 4
Physical interface	Pseudowire interface set	Pseudowire service logical interfaces	One or more queues
Physical interface	Pseudowire transport logical interface	Pseudowire interface set	One or more queues
Physical interface	Pseudowire transport logical interface	Pseudowire service logical interfaces	One or more queues

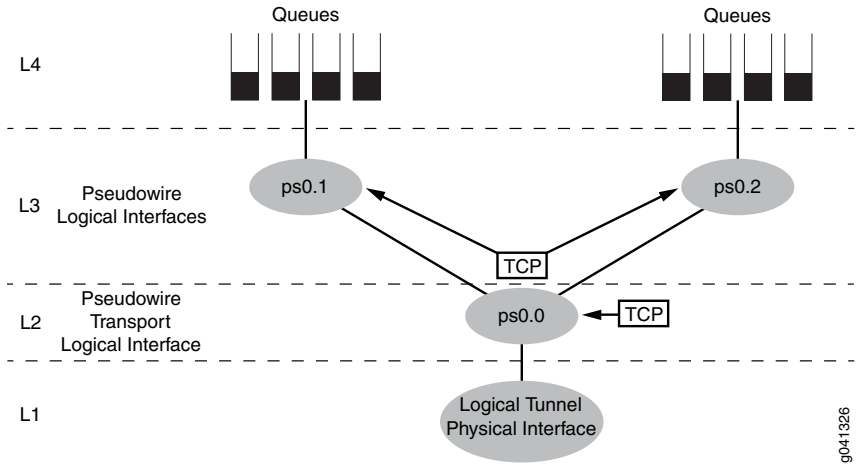
Three-Level Scheduling Hierarchy: Pseudowire Logical Interfaces over a Transport Logical Interface

Figure 4 on page 8 shows an MPLS pseudowire three-level scheduling hierarchy that includes two pseudowire service logical interfaces over a pseudowire transport logical interface. This variation uses the following scheduler nodes:

- Level 4—Forwarding class-based queues
- Level 3—Pseudowire service logical interfaces (ps0.1 and ps0.2) for subscriber sessions
- Level 2—Pseudowire transport logical interface (ps0.0)
- Level 1—Common/shared physical interface of the logical tunnel anchor point

You apply the traffic-control profiles at the pseudowire transport logical interfaces (level 2) and the pseudowire service logical interfaces (level 3).

Figure 4: Three-Level Scheduling Hierarchy Case 1: Pseudowire Service Logical Interfaces over a Transport Logical Interface



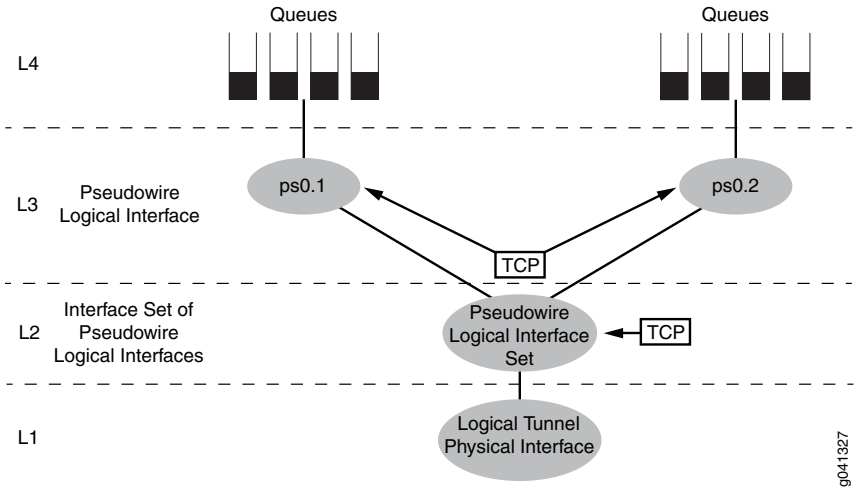
Three-Level Scheduling Hierarchy : Pseudowire Service Logical Interfaces over a Pseudowire Service Interface Set

Figure 5 on page 9 shows another variation of MPLS pseudowire three-level hierarchical scheduling that includes two pseudowire service logical interfaces over a pseudowire service interface set. This variation uses the following CoS scheduler nodes:

- Level 4—Forwarding class-based queues
- Level 3—Pseudowire service logical interfaces (ps0.1 and ps0.2)
- Level 2—Pseudowire service interface set
- Level 1—Common/shared physical interface of the logical tunnel anchor point

You apply the traffic-control profile at the pseudowire service interfaces (level 3) and at the interface set (level 2). This variation is most useful for subscriber edge deployments.

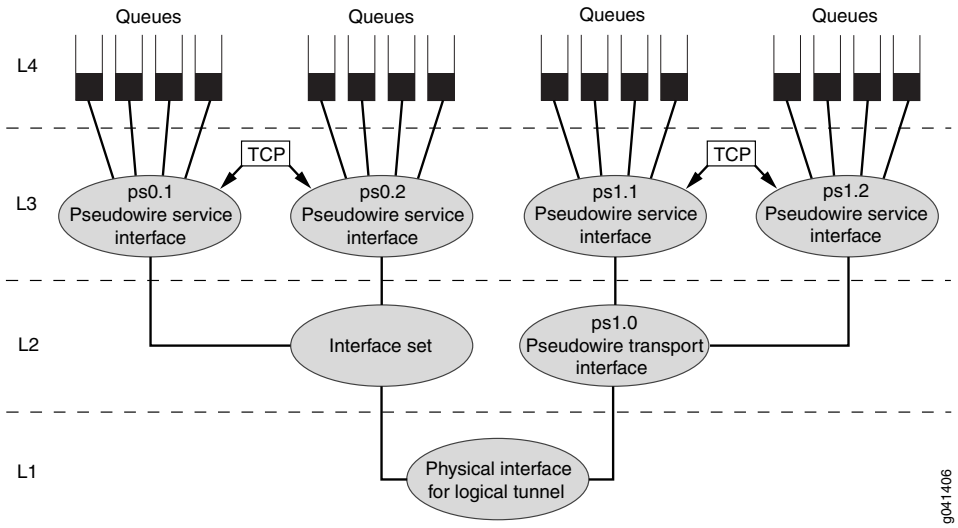
Figure 5: Three-Level Scheduling Hierarchy Case 2: Pseudowire Service Logical Interfaces over a Pseudowire Service Interface Set



Three-Level Scheduling Hierarchy Combined Deployment Scenario

Figure 6 on page 9 shows a deployment scenario that combines the three-level hierarchical scheduling scenarios in Figure 4 on page 8 and Figure 5 on page 9.

Figure 6: Three-Level Hierarchical Scheduling for MPLS Pseudowire Subscriber Interfaces-Deployment Scenario



This variation uses the following CoS scheduler nodes:

- Level 4—Forwarding class-based queues
- Level 3—Pseudowire service logical interfaces (ps0.1, ps0.2, ps1.1, and ps1.2)

- Level 2—Service interface set for pseudowire service interfaces (ps0.1 and ps0.2) and transport logical interface (ps1.0) for the pseudowire service logical interfaces (ps1.1 and ps1.2)
- Level 1—Common/shared physical interface of the logical tunnel anchor point

You apply the traffic-control profiles to the interfaces at both level 2 and level 3, as well as the interface set at level 2.

Related Documentation

- Pseudowire Subscriber Logical Interfaces Overview on page 3
- Configuring a Pseudowire Subscriber Logical Interface on page 13
- *Understanding Two-Level and Three-level Hierarchical CoS for Subscriber Interfaces on MX Series Routers*
- Hierarchical CoS on MPLS Pseudowire Subscriber Interfaces Overview on page 5
- CoS Configuration Overview for MPLS Pseudowire Subscriber Interfaces on page 21
- Configuring CoS Three-Level Hierarchical Scheduling for MPLS Pseudowire Subscriber Interfaces (Logical Interfaces over a Transport Logical Interface) on page 24
- Configuring CoS Three-Level Hierarchical Scheduling for MPLS Pseudowire Subscriber Interfaces (Logical Interfaces over a Pseudowire Interface Set) on page 26
- **hierarchical-scheduler (Subscriber Interfaces on MX Series Routers) on page 36**

PART 2

Configuration

CHAPTER 2

Configuration Tasks for Subscriber Interfaces over MPLS Pseudowires

Configuring a Pseudowire Subscriber Logical Interface

A pseudowire subscriber logical interface terminates an MPLS pseudowire tunnel from an access node to the MX Series router that hosts subscriber management, and enables you to perform subscriber management services at the interface.

To create a pseudowire subscriber logical interface:

1. Specify the number of pseudowire logical interfaces that the router can support.

   See "Configuring the Maximum Number of Pseudowire Logical Interface Devices Supported on the Router" on page 15.

2. Configure the pseudowire subscriber logical interface device.

   See "Configuring a Pseudowire Subscriber Logical Interface Device" on page 15.

3. Configure the transport logical interface.

   See "Configuring the Transport Logical Interface for a Pseudowire Subscriber Logical Interface" on page 17.

4. Configure the signaling for the pseudowire subscriber interface. You can use either Layer 2 circuit signaling or Layer 2 VPN signaling. The two signaling types are mutually exclusive for a given pseudowire.

   - To configure Layer 2 circuit signaling, see "Configuring Layer 2 Circuit Signaling for Pseudowire Subscriber Logical Interfaces" on page 17.
   - To configure Layer 2 VPN signaling, see "Configuring Layer 2 VPN Signaling for Pseudowire Subscriber Logical Interfaces" on page 18.

5. Configure the service logical interface.

   See "Configuring the Service Logical Interface for a Pseudowire Subscriber Logical Interface" on page 19.

6. Configure the underlying interface device.

   See *Configuring an Underlying Interface for Dynamic PPPoE Subscriber Interfaces*.

7. Configure CoS parameters and BA classification.

   See "CoS Configuration Overview for MPLS Pseudowire Subscriber Interfaces" on page 21.

8. (Optional) Associate a dynamic profile with the pseudowire subscriber logical interface.

   You can associate DHCP, PPPoE, IP demux, and VLAN dynamic profiles with pseudowire subscriber logical interfaces. The support is similar to the typical Ethernet interface support.

   NOTE: **When using a PPPoE dynamic profile to create a pseudowire subscriber logical interface over a demux interface device, the dynamic profile must explicitly specify the correct pseudowire interface device over which the interface is created. The dynamic profile does not automatically create the interface over the demux0 interface device, as is the case with a VLAN demux interface.**

   For additional information about associating dynamic profiles to interfaces, see *Dynamic Profile Attachment to DHCP Subscriber Interfaces Overview* and *Configuring VLAN Interfaces to Use Dynamic Profiles*.

9. (Optional) Configure interface set support for pseudowire subscriber logical interfaces.

   See *Configuring Interface Sets* and *Applying Interface Sets*.

10. (Optional) Stack PPPoE logical interfaces over a pseudowire logical device.

Related Documentation

- Pseudowire Subscriber Logical Interfaces Overview on page 3

Configuring the Maximum Number of Pseudowire Logical Interface Devices Supported on the Router

You must set the maximum number of pseudowire logical interface devices (pseudowire tunnels) that the router can use for subscriber logical interfaces. You can specify a maximum of 2048 pseudowire logical interface devices for an MX Series router. Each pseudowire device supports a maximum of 4000 subscriber logical interfaces.

A PFE can host a maximum of 2048 pseudowire logical interface devices, which is the chassis maximum. This PFE hosting support provides the configuration flexibility needed for special cases that might occur for business edge scenarios. However, you can exceed the available PFE resources as you configure additional services on the pseudowire logical interface devices ports. To support a scaled configuration, ensure that you populate the appropriate number of PFEs for the chassis, and that you distribute the pseudowire logical interface devices across the PFEs in such a way that ensures that no PFE is overwhelmed by the anticipated peak load. As part of the network planning for your particular deployment, you must consider the exact mix of the distribution of the pseudowire logical interface devices and the services associated with the devices.

BEST PRACTICE: **A configured pseudowire logical interface device consumes resources from shared pools even when the device has no active subscriber logical interfaces. To conserve resources, do not deploy an excessive number of pseudowire devices that you do not intend to use.**

To configure the number of pseudowire logical interface devices that you want the router to support:

1. Specify that you want to configure the pseudowire service.

   ```
   [edit chassis]
   user@host# edit pseudowire-service
   ```

2. Set the maximum number of pseudowire logical interface devices.

   ```
   [edit chassis pseudowire-service]
   user@host# set device-count 500
   ```

Related Documentation

- Pseudowire Subscriber Logical Interfaces Overview on page 3
- Configuring a Pseudowire Subscriber Logical Interface on page 13

Configuring a Pseudowire Subscriber Logical Interface Device

To configure a pseudowire logical interface device that the router uses for subscriber logical interfaces, you specify the logical tunnel that processes the pseudowire termination. You can also configure additional optional parameters for the interface device, such as VLAN tagging method, MTU, and gratuitous ARP support.

To configure the pseudowire subscriber interface device:

1. Specify that you want to configure the pseudowire subscriber logical interface device.

   user@host# **edit interfaces** ps0

2. Specify the logical tunnel interface that is the anchor point for the pseudowire logical interface device. The anchor point must be an **lt** device in the format **lt-*fpc/pic/port***.

   [edit interfaces ps0]
   user@host# **set** anchor-point **lt-1/0/10**

3. (Optional) Specify the VLAN tagging method used for the pseudowire logical interface device. You can specify single tagging, dual (stacked) tagging, mixed (flexible) tagging, or no tagging.

   [edit interfaces ps0]
   user@host# **set** flexible-vlan-tagging

   See *Enabling VLAN Tagging* for additional information about VLAN tagging.

4. (Optional) Specify the MTU for the pseudowire logical interface device. If you do not explicitly configure the MTU, the router uses the default value of 1500.

   [edit interfaces ps0]
   user@host# **set** mtu **2500**

   See *Setting the Protocol MTU* for additional information.

5. (Optional) Specify that the pseudowire logical interface device does not respond to gratuitous ARP requests.

   [edit interfaces ps0]
   user@host# **set** no-gratuitous-arp-request

   See *Configuring Gratuitous ARP* for additional information.

6. Configure additional optional parameters for the pseudowire logical interface device, such as *description*, *apply-groups*, *apply-groups-except*, and traceoptions.

Related Documentation

- Pseudowire Subscriber Logical Interfaces Overview on page 3
- Configuring a Pseudowire Subscriber Logical Interface on page 13

Configuring the Transport Logical Interface for a Pseudowire Subscriber Logical Interface

This topic describes how to configure a pseudowire transport logical interface. A pseudowire device can have only one transport logical interface.

A pseudowire logical device and its related pseudowire logical interfaces are dependent on the state of the underlying logical transport interface device, which is either the Layer 2 VPN or Layer 2 circuit.

NOTE: We recommend that you use unit 0 to represent the transport logical interface for the pseudowire device. Non-zero unit numbers represent *service* logical interfaces used for pseudowire subscriber interfaces.

To configure a pseudowire transport logical interface:

1. Specify that you want to configure the pseudowire subscriber logical interface device.

   ```
   [edit]
   user@host# edit interfaces ps0
   ```

2. Specify that you want to configure unit 0, which represents the transport logical interface.

   ```
   [edit interfaces ps0]
   user@host# edit unit 0
   ```

3. Specify the **ethernet-ccc** encapsulation method for the transport logical interface.

   ```
   [edit interfaces ps0 unit 0]
   user@host# set encapsulation ethernet-ccc
   ```

Related Documentation

- Pseudowire Subscriber Logical Interfaces Overview on page 3
- Configuring a Pseudowire Subscriber Logical Interface on page 13

Configuring Layer 2 Circuit Signaling for Pseudowire Subscriber Logical Interfaces

This topic describes the steps for configuring Layer 2 circuit signaling used for the pseudowire subscriber logical interface support. You can also use Layer 2 VPN signaling for pseudowire subscriber logical interfaces. The two methods are mutually exclusive; you can use only one method for a particular pseudowire.

To configure Layer 2 circuit signaling for pseudowire interfaces:

1. Specify that you want to configure Layer 2 circuit parameters at the protocols hierarchy level.

   ```
   [edit protocols]
   user@host# edit l2circuit
   ```

2. Specify the IP address of the neighbor, to identify the PE router used for the Layer 2 circuit.

   ```
   [edit protocols l2circuit]
   user@host# edit neighbor 192.168.102.15
   ```

3. Specify the interface used by the Layer 2 circuit traffic.

   ```
   [edit protocols l2circuit neighbor 192.168.102.15]
   user@host# edit interface ps1.0
   ```

4. Configure the virtual circuit ID that identifies the Layer 2 circuit for the pseudowire.

   ```
   [edit protocols l2circuit neighbor 192.168.102.15 interface ps1.0]
   user@host# set virtual-circuit-id 5
   ```

5. (Optional) If multiple VLAN interfaces are carried over the pseudowire Layer 2 payload, configure the **no-vlan-id-validate** statement. This statement prevents VLAN validation in the signaling.

   ```
   [edit protocols l2circuit neighbor 192.168.102.15 interface ps1.0]
   user@host# set no-vlan-id-validate
   ```

For more information about Layer 2 circuits, see *Configuring Interfaces for Layer 2 Circuits*.

Related Documentation

- Pseudowire Subscriber Logical Interfaces Overview on page 3
- Configuring a Pseudowire Subscriber Logical Interface on page 13
- *Configuring Interfaces for Layer 2 Circuits*

Configuring Layer 2 VPN Signaling for Pseudowire Subscriber Logical Interfaces

This topic describes the steps for configuring Layer 2 VPN signaling used for the pseudowire subscriber logical interface support. You can also use Layer 2 circuit signaling for pseudowire subscriber logical interfaces. The two methods are mutually exclusive; you can use only one method on a particular pseudowire.

To configure Layer 2 VPN signaling for pseudowire interfaces:

1. Specify the name of the routing instance you want to configure.

   ```
   [edit]
   user@host# edit routing-instances l2vpn0
   ```

2. Configure the Layer 2 VPN routing instance type.

   ```
   [edit routing-instances l2vpn0]
   user@host# set instance-type l2vpn
   ```

3. Associate the pseudowire logical interface for the Layer 2 VPN.

   ```
   [edit routing-instances l2vpn0]
   user@host# set interface ps1.0
   ```

4. Configure the unique identifier for the routes that belong to the Layer 2 VPN.

   ```
   [edit routing-instances l2vpn0]
   user@host# set route-distinguisher 111.1.1.1:100
   ```

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

   ```
   [edit routing-instances l2vpn0]
   user@host# set vrf-target target:10:100
   ```

6. Specify that you want to configure the Layer 2 VPN protocol for the routing instance.

   ```
   [edit routing-instances l2vpn0]
   user@host# edit protocols l2vpn
   ```

7. Configure the encapsulation type for the routing instance.

   ```
   [edit routing-instances l2vpn0 protocols l2vpn]
   user@host# set encapsulation-type ethernet
   ```

8. Specify the site name and site identifier for the Layer 2 VPN.

   ```
   [edit routing-instances l2vpn0 protocols l2vpn]
   user@host# set site PE1 site-identifier 1
   ```

9. Specify the interface that connects to the site, and the remote interface to which you want the specified interface to connect.

   ```
   [edit routing-instances l2vpn0 protocols l2vpn]
   user@host# set interface ps1.0 remote-site-id 2
   ```

10. Configure the tracing options for traffic that uses the Layer 2 VPN.

    ```
    [edit routing-instances l2vpn0 protocols l2vpn]
    user@host# set traceoptions file l2vpn flag all
    ```

Related Documentation

- Pseudowire Subscriber Logical Interfaces Overview on page 3
- Configuring a Pseudowire Subscriber Logical Interface on page 13

Configuring the Service Logical Interface for a Pseudowire Subscriber Logical Interface

This topic describes how to configure a pseudowire service logical interface. Service logical interfaces represent the attachment circuits for pseudowire logical interfaces.

As described in the "Pseudowire Subscriber Logical Interfaces Overview" on page 3, you can choose whether to configure a service logical interface together with a higher subscriber logical interface, depending upon the business need. In a broadband edge configuration, the higher subscriber logical interface is the demarcation point for subscribers. However, in a business edge configuration, the service logical interface is the demarcation point for the business subscribers, and also serves as the subscriber logical interface, so no subscriber logical interfaces are explicitly configured.

NOTE: Non-zero unit numbers represent *service* logical interfaces used for pseudowire subscriber interfaces. Use unit 0 to represent the *transport* logical interface for the pseudowire device.

To configure a pseudowire service logical interface:

1. Specify that you want to configure the pseudowire subscriber logical interface device.

```
[edit]
user@host# edit interfaces ps0
```

2. Configure the unit for the service logical interface. Use a non-zero unit number.

```
[edit interfaces ps0]
user@host# edit unit 1
```

3. Configure the VLAN tag IDs.

```
[edit interfaces ps0 unit 1]
user@host# set vlan-tags outer 1 inner 1
```

4. Configure the interface to respond to ARP requests when the device has an active route to the ARP request target address.

```
[edit interfaces ps0 unit 1]
user@host# set proxy-arp
```

5. Specify that you want to configure the protocol family information. Pseudowire service logical interfaces support IPv4 (inet), IPv6 (inet6), and PPPoE (pppoe) protocol families.

   For example, to configure the IPv4 family:

   a. Specify that you want to configure IPv4.

```
[edit interfaces ps0 unit 1]
user@host# edit family inet
```

   b. Configure the parameters for the family.

```
[edit interfaces ps0 unit 1 family inet]
user@host# set filter input filter 1 output filter 4
user@host# set mac-validate loose
user@host# set input-hierarchical-policer policer-1
user@host# set unnumbered-address lo0.0 preferred-source-address 100.0.0.1
```

Related Documentation

- Pseudowire Subscriber Logical Interfaces Overview on page 3
- Configuring a Pseudowire Subscriber Logical Interface on page 13

CHAPTER 3

CoS Configuration Tasks for Subscriber Interfaces over MPLS Pseudowires

CoS Configuration Overview for MPLS Pseudowire Subscriber Interfaces

CoS supports two-level and three-level hierarchies for MPLS pseudowire subscriber interfaces.

To configure two-level scheduling, include the **maximum-hierarchy-levels 2** option under the **[edit interfaces *interface-name* hierarchical-scheduler]** statement on the physical interface of the logical tunnel anchor point.

To configure three-level hierarchical scheduling, include the **implicit-hierarchy** option under the **[edit interfaces *interface-name* hierarchical-scheduler]** statement on the physical interface of the logical tunnel anchor point. Use the following guidelines for configuring the **implicit-hierarchy** option:

- If an output traffic-control profile is configured on the pseudowire transport interface and on a pseudowire service interface, the two interfaces form a scheduling hierarchy. The pseudowire transport interface resides in a level 2 scheduler node and the pseudowire service interface resides in a level 3 scheduler node.
- If an output traffic-control profile is configured on the pseudowire services interface but not on a pseudowire transport interface, the pseudowire services interface resides in a level 3 scheduler node.
- If an output traffic-control profile is only configured on the pseudowire transport interface and not on the pseudowire services interface, the pseudowire transport interface resides in a level 3 scheduler node and all pseudowire traffic uses this node.

If the **implicit-hierarchy** option is not set on the logical tunnel anchor point, logical interfaces behave normally with the hierarchical-scheduler mode configured with or without the **hierarchical-scheduler maximum-hierarchy-levels** option under the **[edit interfaces *interface-name* hierarchical-scheduler]** statement. In this case, when you apply a traffic-control profile to the pseudowire and service logical interfaces, they both reside in level 3 scheduler nodes and do not form a scheduling hierarchy, which might not be the desirable behavior. In business edge, where only the pseudowire logical interfaces need to be shaped, applying the traffic-control profile at just the transport logical interface may be sufficient.

When configuring the logical tunnel physical interface for the maximum hierarchy level, all pseudowire logical interfaces operating on the physical interface use the same hierarchy model. If you want to mix two-level and three-level scheduling hierarchies, you can group the pseudowires together by hierarchy levels and share the same logical tunnel anchor point or you can use three-level scheduling for all pseudowires over the anchor point.

To specify rewrite rules and classifiers on pseudowire interfaces, reference the pseudowire device under the **[edit class-of-service interfaces]** hierarchy level and specify the rewrite rules and classifiers for the pseudowire interfaces.

To control all pseudowire traffic using the same logical tunnel interface, apply CoS policies at the physical interface for the anchor logical tunnel.

Related Documentation

- Pseudowire Subscriber Logical Interfaces Overview on page 3
- Configuring a Pseudowire Subscriber Logical Interface on page 13
- *Understanding Two-Level and Three-level Hierarchical CoS for Subscriber Interfaces on MX Series Routers*
- Hierarchical CoS on MPLS Pseudowire Subscriber Interfaces Overview on page 5
- Configuring CoS Two-Level Hierarchical Scheduling for MPLS Pseudowire Subscriber Interfaces on page 22
- Configuring CoS Three-Level Hierarchical Scheduling for MPLS Pseudowire Subscriber Interfaces (Logical Interfaces over a Transport Logical Interface) on page 24
- Configuring CoS Three-Level Hierarchical Scheduling for MPLS Pseudowire Subscriber Interfaces (Logical Interfaces over a Pseudowire Interface Set) on page 26
- **hierarchical-scheduler (Subscriber Interfaces on MX Series Routers) on page 36**

Configuring CoS Two-Level Hierarchical Scheduling for MPLS Pseudowire Subscriber Interfaces

Before configuring CoS parameters for MPLS pseudowire subscriber interfaces, you must first complete these tasks:

1. Configure the pseudowire logical interfaces. See "Configuring a Pseudowire Subscriber Logical Interface" on page 13.
2. Configure the pseudowire device count. See "Configuring the Maximum Number of Pseudowire Logical Interface Devices Supported on the Router" on page 15.

3. Configure the pseudowire device including the logical tunnel anchor point. See "Configuring a Pseudowire Subscriber Logical Interface Device" on page 15.
4. Configure the pseudowire transport logical interface. See "Configuring the Transport Logical Interface for a Pseudowire Subscriber Logical Interface" on page 17.
5. Configure the pseudowire signaling (either Layer 2 circuit signaling or Layer 2 VPN signaling). See "Configuring Layer 2 Circuit Signaling for Pseudowire Subscriber Logical Interfaces" on page 17 or "Configuring Layer 2 VPN Signaling for Pseudowire Subscriber Logical Interfaces" on page 18.
6. Configure the pseudowire logical interfaces. See "Configuring the Service Logical Interface for a Pseudowire Subscriber Logical Interface" on page 19.

To configure CoS policies on MPLS pseudowire subscriber interfaces using two-level scheduling:

1. Configure the hierarchical scheduler for the physical interface used for the logical tunnel (anchor point). For two-level scheduling the hierarchical scheduler must be set to **maximum-scheduler levels 2**.

   ```
   [edit]
   user@host#edit interfaces ps ps-anchor-device-name
   user@host#set hierarchical-scheduler maximum-hierarchy-levels 2
   ```

2. Specify the traffic-control profile to use on the pseudowire logical interface.

   ```
   [edit class-of-service]
   user@host#edit interfaces ps ps-device-name
   user@host#edit unit logical-unit-number
   user@host#set output-traffic-control-profile profile-name
   ```

3. Configure the rewrite rule.

   The available rewrite rule types for pseudowire interfaces are **dscp** and **inet-precedence**.

   ```
   [edit class-of-service]
   user@host#edit interfaces ps ps-device-name
   user@host#edit unit logical-unit-number
   user@host#edit rewrite-rules (dscp | inet-precedence) rewrite-name
   user@host#edit forwarding-class class-name
   user@host#set loss-priority class-name code-point (alias | bits)
   ```

4. Configure the classifier.

   The available classifier types for pseudowire interfaces are **dscp** and **inet-precedence**.

   ```
   [edit class-of-service]
   user@host#edit interfaces ps ps-device-name
   user@host#edit unit logical-unit-number
   user@host#edit classifiers (dscp | inet-precedence) classifier-name
   user@host#edit forwarding-class class-name
   user@host#set loss-priority class-name code-points [aliases] [bit-patterns]
   ```

5. Apply the rewrite rule and classifier to the pseudowire interface.

   For the *interface_name* parameter, specify the pseudowire device name.

   ```
   [edit class-of-service interfaces interface_name unit logical-unit-number]
   ```

user@host#set rewrite-rule (dscp | inet-precedence) (*rewrite-name* | default) protocol *protocol-types*
user@host#set classifiers (dscp | inet-precedence) (*classifier-name* | default)

Related Documentation

- *CoS on Ethernet Pseudowires in Universal Edge Networks Overview*
- For more information about rewrite rules and classifiers, see the *Junos OS Class of Service Library for Routing Devices*
- *Understanding Two-Level and Three-level Hierarchical CoS for Subscriber Interfaces on MX Series Routers*
- Hierarchical CoS on MPLS Pseudowire Subscriber Interfaces Overview on page 5
- CoS Two-Level Hierarchical Scheduling on MPLS Pseudowire Subscriber Interfaces on page 5
- CoS Three-Level Hierarchical Scheduling on MPLS Pseudowire Subscriber Interfaces on page 7
- CoS Configuration Overview for MPLS Pseudowire Subscriber Interfaces on page 21
- **hierarchical-scheduler (Subscriber Interfaces on MX Series Routers) on page 36**

Configuring CoS Three-Level Hierarchical Scheduling for MPLS Pseudowire Subscriber Interfaces (Logical Interfaces over a Transport Logical Interface)

Before configuring CoS three-level scheduling on pseudowire logical interfaces over a transport logical interface, you must first complete these tasks:

1. Configure the pseudowire logical interfaces. See "Configuring a Pseudowire Subscriber Logical Interface" on page 13.
2. Configure the pseudowire device count. See "Configuring the Maximum Number of Pseudowire Logical Interface Devices Supported on the Router" on page 15.
3. Configure the pseudowire device including the logical tunnel anchor point. See "Configuring a Pseudowire Subscriber Logical Interface Device" on page 15.
4. Configure the pseudowire transport logical interface. See "Configuring the Transport Logical Interface for a Pseudowire Subscriber Logical Interface" on page 17.
5. Configure the pseudowire signaling (either Layer 2 circuit signaling or Layer 2 VPN signaling). See "Configuring Layer 2 Circuit Signaling for Pseudowire Subscriber Logical Interfaces" on page 17 or "Configuring Layer 2 VPN Signaling for Pseudowire Subscriber Logical Interfaces" on page 18.
6. Configure the pseudowire logical interfaces. See "Configuring the Service Logical Interface for a Pseudowire Subscriber Logical Interface" on page 19.

Three-level scheduling on pseudowire logical interfaces over a transport logical interface requires you to apply the traffic-control profiles at both the pseudowire logical interface and the pseudowire transport logical interface. To configure CoS policies on three-level scheduling on pseudowire logical interfaces over a transport logical interface:

1. Configure the hierarchical scheduler for the physical interface used for the logical tunnel (anchor point). For three-level scheduling the hierarchical scheduler must be set to **implicit-hierarchy**.

   [edit]
   user@host#edit interfaces *ps-anchor-device-name*
   user@host#set hierarchical-scheduler implicit-hierarchy

2. Specify the traffic-control profile to use on the pseudowire logical interface.

   [edit class-of-service]
   user@host#edit interfaces ps *ps-device-name*
   user@host#edit unit *logical-unit-number*
   user@host#set output-traffic-control-profile *profile-name*

3. Specify the traffic-control profile to use on the pseudowire transport logical interface.

   [edit class-of-service]
   user@host#edit interfaces ps *ps-device-name*
   user@host#edit unit *logical-unit-number*
   user@host#set output-traffic-control-profile *profile-name*

4. Configure the rewrite rule.

   The available rewrite rule types for pseudowire interfaces are **dscp** and **inet-precedence**.

   [edit class-of-service]
   user@host#edit interfaces ps *ps-device-name*
   user@host#edit unit *logical-unit-number*
   user@host#edit rewrite-rules (dscp | inet-precedence) *rewrite-name*
   user@host#edit forwarding-class *class-name*
   user@host#set loss-priority *class-name* code-point (*alias* | *bits*)

5. Configure the classifier.

   The available classifier types for pseudowire interfaces are **dscp** and **inet-precedence**.

   [edit class-of-service]
   user@host#edit interfaces ps *ps-device-name*
   user@host#edit unit *logical-unit-number*
   user@host#edit classifiers (dscp | inet-precedence) *classifier-name*
   user@host#edit forwarding-class *class-name*
   user@host#set loss-priority *class-name* code-points [*aliases*] [*bit-patterns*]

6. Apply the rewrite rule and classifier to the pseudowire interfaces.

   For the *interface_name* parameter, specify the pseudowire device name.

   [edit class-of-service interfaces *interface_name* unit *logical-unit-number*]
   user@host#set rewrite-rule (dscp | inet-precedence) (*rewrite-name* | default) protocol *protocol-types*
   user@host#set classifiers (dscp | inet-precedence) (*classifier-name* | default)

Related Documentation

- *CoS on Ethernet Pseudowires in Universal Edge Networks Overview*
- For more information about rewrite rules and classifiers, see the *Junos OS Class of Service Library for Routing Devices*
- *Understanding Two-Level and Three-level Hierarchical CoS for Subscriber Interfaces on MX Series Routers*
- Hierarchical CoS on MPLS Pseudowire Subscriber Interfaces Overview on page 5
- CoS Three-Level Hierarchical Scheduling on MPLS Pseudowire Subscriber Interfaces on page 7
- CoS Configuration Overview for MPLS Pseudowire Subscriber Interfaces on page 21
- Configuring CoS Three-Level Hierarchical Scheduling for MPLS Pseudowire Subscriber Interfaces (Logical Interfaces over a Pseudowire Interface Set) on page 26
- **hierarchical-scheduler (Subscriber Interfaces on MX Series Routers) on page 36**

Configuring CoS Three-Level Hierarchical Scheduling for MPLS Pseudowire Subscriber Interfaces (Logical Interfaces over a Pseudowire Interface Set)

Before configuring three-level scheduling on pseudowire logical interfaces over a pseudowire logical interface set, you must first complete the following tasks:

1. Configure the pseudowire logical interfaces. See "Configuring a Pseudowire Subscriber Logical Interface" on page 13.
2. Configure the pseudowire device count. See "Configuring the Maximum Number of Pseudowire Logical Interface Devices Supported on the Router" on page 15.
3. Configure the pseudowire device including the logical tunnel anchor point. See "Configuring a Pseudowire Subscriber Logical Interface Device" on page 15.
4. Configure the pseudowire transport logical interface. See "Configuring the Transport Logical Interface for a Pseudowire Subscriber Logical Interface" on page 17.
5. Configure the pseudowire signaling (either Layer 2 circuit signaling or Layer 2 VPN signaling). See "Configuring Layer 2 Circuit Signaling for Pseudowire Subscriber Logical Interfaces" on page 17 or "Configuring Layer 2 VPN Signaling for Pseudowire Subscriber Logical Interfaces" on page 18.
6. Configure the pseudowire logical interfaces. See "Configuring the Service Logical Interface for a Pseudowire Subscriber Logical Interface" on page 19.

Three-level scheduling on pseudowire logical interfaces over a pseudowire logical interface set requires you to apply the traffic-control profiles at both the pseudowire logical interface and the pseudowire logical interface-set. To configure CoS policies on MPLS pseudowire subscriber interfaces using three-level implicit hierarchical scheduling:

1. Configure the hierarchical scheduler for the physical interface used for the logical tunnel (anchor point). For three-level scheduling the hierarchical scheduler must be set to **implicit-hierarchy**.

```
[edit]
user@host#edit interfaces ps-anchor-device-name
user@host#set hierarchical-scheduler implicit-hierarchy
```

2. Specify the traffic-control profile to use on the pseudowire logical interfaces.

```
[edit class-of-service]
user@host#edit interfaces ps ps-device-name
user@host#edit unit logical-unit-number
user@host#set output-traffic-control-profile profile-name
```

3. Define a pseudowire logical interface set and configure the traffic-control profile used for the interface set.

```
[edit class-of-service]
user@host#edit interfaces
user@host#edit interface-set interface-set-name
user@host#edit output-traffic-control-profile profile-name
```

4. Group the pseudowire logical interfaces in the pseudowire logical interface set.

```
[edit ]
user@host#edit interfaces
user@host#edit interface-set interface-set-name
user@host#edit interface ps ps-device-name
user@host#edit unit logical-unit-number
```

5. Configure the rewrite rule.

   The available rewrite rule types for pseudowire interfaces are **dscp** and **inet-precedence**.

```
[edit class-of-service]
user@host#edit interfaces ps ps-device-name
user@host#edit unit logical-unit-number
user@host#edit rewrite-rules (dscp | inet-precedence) rewrite-name
user@host#edit forwarding-class class-name
user@host#set loss-priority class-name code-point (alias | bits)
```

6. Configure the classifier.

   The available classifier types for pseudowire interfaces are **dscp** and **inet-precedence**.

```
[edit class-of-service]
user@host#edit interfaces ps ps-device-name
user@host#edit unit logical-unit-number
user@host#edit classifiers (dscp | inet-precedence) classifier-name
user@host#edit forwarding-class class-name
user@host#set loss-priority class-name code-points [aliases] [bit-patterns]
```

7. Apply the rewrite rule and classifier to the pseudowire interfaces.

   For the *interface_name* parameter, specify the ps device name.

```
[edit class-of-service interfaces interface_name unit logical-unit-number]
user@host#set rewrite-rule (dscp | inet-precedence) (rewrite-name | default) protocol
  protocol-types
user@host#set classifiers (dscp | inet-precedence) (classifier-name | default)
```

Related Documentation

- *CoS on Ethernet Pseudowires in Universal Edge Networks Overview*

- For more information about rewrite rules and classifiers, see the *Junos OS Class of Service Library for Routing Devices*
- *Understanding Two-Level and Three-level Hierarchical CoS for Subscriber Interfaces on MX Series Routers*
- Hierarchical CoS on MPLS Pseudowire Subscriber Interfaces Overview on page 5
- CoS Three-Level Hierarchical Scheduling on MPLS Pseudowire Subscriber Interfaces on page 7
- CoS Configuration Overview for MPLS Pseudowire Subscriber Interfaces on page 21
- Configuring CoS Three-Level Hierarchical Scheduling for MPLS Pseudowire Subscriber Interfaces (Logical Interfaces over a Transport Logical Interface) on page 24
- **hierarchical-scheduler (Subscriber Interfaces on MX Series Routers) on page 36**

CHAPTER 4

Configuration Statements

anchor-point (Pseudowire Subscriber Interfaces)

Syntax anchor-point *lt-device*;

Hierarchy Level [edit interfaces ps0]

Release Information Statement introduced in Junos OS Release 13.1.

Description Specify the logical tunnel (lt) interface that identifies the Packet Forwarding Engine that processes the pseudowire termination.

Options ***lt-device***—An lt device in the format **lt-*fpc/pic/port***

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

Related Documentation

- Pseudowire Subscriber Logical Interfaces Overview on page 3
- Configuring a Pseudowire Subscriber Logical Interface on page 13
- Configuring a Pseudowire Subscriber Logical Interface Device on page 15

device-count (Pseudowire Subscriber Interfaces)

Syntax device-count *number*;

Hierarchy Level [edit chassis pseudowire-service]

Release Information Statement introduced in Junos OS Release 13.1.

Description Configure the number of pseudowire logical devices available to the router.

Options ***number***—Number of devices.
Range: 1 through 2048

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

Related Documentation

- Pseudowire Subscriber Logical Interfaces Overview on page 3
- Configuring a Pseudowire Subscriber Logical Interface on page 13
- Configuring the Maximum Number of Pseudowire Logical Interface Devices Supported on the Router on page 15

encapsulation (Logical Interface)

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

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

Release Information Statement introduced before Junos OS Release 7.4.
Statement introduced in Junos OS Release 12.1X48 for PTX Series Packet Transport Routers (**vlan-ccc** and **vlan-tcc** options only).
Statement introduced in Junos OS Release 12.2 for the ACX Series Universal Access Routers. Only the **atm-ccc-cell-relay** and **atm-ccc-vc-mux** options are supported on ACX Series routers.

Description Configure a logical link-layer encapsulation type.

Options

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

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

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

atm-mlppp-llc—For ATM2 IQ interfaces only, use Multilink Point-to-Point (MLPPP) over AAL5 LLC. For this encapsulation type, your router must be equipped with a Link Services or Voice Services PIC. MLPPP over ATM encapsulation is not supported on ATM2 IQ OC48 interfaces.

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

atm-ppp-llc—(ATM2 IQ interfaces and MX Series routers with MPC/MIC interfaces using the ATM MIC with SFP only) Use PPP over AAL5 LLC encapsulation.

atm-ppp-vc-mux—(ATM2 IQ interfaces and MX Series routers with MPC/MIC interfaces using the ATM MIC with SFP only) Use PPP over ATM AAL5 multiplex encapsulation.

atm-snap—(All interfaces including MX Series routers with MPC/MIC interfaces using the ATM MIC with SFP) Use ATM subnetwork attachment point (SNAP) encapsulation.

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

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

atm-vc-mux—(All interfaces including MX Series routers with MPC/MIC interfaces using the ATM MIC with SFP) Use ATM VC multiplex encapsulation. When you use this encapsulation type, you can configure the **inet** family only.

ether-over-atm-llc—(All IP interfaces including MX Series routers with MPC/MIC interfaces using the ATM MIC with SFP) For interfaces that carry IP traffic, use Ethernet over ATM LLC encapsulation. When you use this encapsulation type, you cannot configure multipoint interfaces.

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

ether-vpls-over-fr—For E1, T1, E3, T3, and SONET interfaces only, use the Ethernet virtual private LAN service (VPLS) over Frame Relay encapsulation to support Bridged Ethernet over Frame Relay encapsulated TDM interfaces for VPLS applications, per RFC 2427, *Multiprotocol Interconnect over Frame Relay*.

NOTE: **The SONET/SDH OC3/STM1 (Multi-Rate) MIC with SFP, the Channelized SONET/SDH OC3/STM1 (Multi-Rate) MIC with SFP, and the DS3/E3 MIC do not support Ethernet over Frame Relay encapsulation.**

ether-vpls-over-ppp—For E1, T1, E3, T3, and SONET interfaces only, use the Ethernet virtual private LAN service (VPLS) over Point-to-Point Protocol (PPP) encapsulation to support Bridged Ethernet over PPP-encapsulated TDM interfaces for VPLS applications.

ethernet—Use Ethernet II encapsulation (as described in RFC 894, *A Standard for the Transmission of IP Datagrams over Ethernet Networks*).

ethernet-ccc—Use Ethernet CCC encapsulation on Ethernet interfaces.

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

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

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

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

frame-relay-ether-type—Use Frame Relay ether type encapsulation for compatibility with Cisco Frame Relay. The physical interface must be configured with flexible-frame-relay encapsulation.

frame-relay-ether-type-tcc—Use Frame Relay ether type TCC for Cisco-compatible Frame Relay on TCC circuits to connect different media. The physical interface must be configured with flexible-frame-relay encapsulation.

frame-relay-ppp—Use PPP over Frame Relay circuits. When you use this encapsulation type, you can configure the **ppp** family only. J Series routers do not support frame-relay-ppp encapsulation.

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

gre-fragmentation—For adaptive services interfaces only, use GRE fragmentation encapsulation to enable fragmentation of IPv4 packets in GRE tunnels. This encapsulation clears the do not fragment (DF) bit in the packet header. If the packet's size exceeds the tunnel's maximum transmission unit (MTU) value, the packet is fragmented before encapsulation.

multilink-frame-relay-end-to-end—Use MLFR FRF.15 encapsulation. This encapsulation is used only on multilink, link services, and voice services interfaces and their constituent T1 or E1 interfaces, and is supported on LSQ and redundant LSQ interfaces.

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

ppp-over-ether—For underlying Ethernet interfaces on J Series routers, use PPP over Ethernet encapsulation. When you use this encapsulation type, you cannot configure the interface address. Instead, configure the interface address on the PPP interface. You also use PPP over Ethernet encapsulation to configure an underlying Ethernet interface for a dynamic PPPoE logical interface on M120 and M320 routers with Intelligent Queuing 2 (IQ2) PICs, and on MX Series routers with MPCs.

ppp-over-ether-over-atm-llc—(J Series routers and MX Series routers with MPCs using the ATM MIC with SFP only) For underlying ATM interfaces, use PPP over Ethernet over ATM LLC encapsulation. When you use this encapsulation type, you cannot configure the interface address. Instead, configure the interface address on the PPP interface.

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

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

vlan-vci-ccc—Use ATM-to-Ethernet interworking encapsulation on CCC circuits. When you use this encapsulation type, you can configure the **ccc** family only.

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

vlan-vpls—Use Ethernet VLAN encapsulation on VPLS circuits.

Required Privilege Level

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

Related Documentation

- *Configuring Layer 2 Switching Cross-Connects Using CCC*
- *Configuring the Encapsulation for Layer 2 Switching TCCs*
- *Configuring Interface Encapsulation on Logical Interfaces*
- *Configuring MPLS LSP Tunnel Cross-Connects Using CCC*
- *Circuit and Translational Cross-Connects Overview*
- *Identifying the Access Concentrator*
- *Configuring ATM Interface Encapsulation*
- *Configuring VLAN Encapsulation*
- *Configuring Extended VLAN Encapsulation*
- *Configuring ISDN Logical Interface Properties*
- *Configuring ATM-to-Ethernet Interworking*
- *Configuring Interface Encapsulation on PTX Series Packet Transport Routers*
- *Configuring CCC Encapsulation for Layer 2 VPNs*
- *Configuring TCC Encapsulation for Layer 2 VPNs and Layer 2 Circuits*
- *Configuring ATM for Subscriber Access*
- *Junos OS Services Interfaces Library for Routing Devices*
- *CoS on ATM IMA Pseudowire Interfaces Overview*
- *Configuring Policing on an ATM IMA Pseudowire*

flexible-vlan-tagging

Syntax flexible-vlan-tagging;

Hierarchy Level [edit interfaces ge-*fpc/pic/port*],
[edit interfaces et-*fpc/pic/port*],
[edit interfaces ps0]

Release Information Statement introduced in Junos OS Release 8.1.
Support for aggregated Ethernet added in Junos OS Release 9.0.
Statement introduced in Junos OS Release 12.1x48 for PTX Series Packet Transport Routers.

Description Support simultaneous transmission of 802.1Q VLAN single-tag and dual-tag frames on logical interfaces on the same Ethernet port, and on pseudowire logical interfaces.

This statement is supported on M Series and T Series routers, for Fast Ethernet and Gigabit Ethernet interfaces only on Gigabit Ethernet IQ2 and IQ2-E, IQ, and IQE PICs, and for aggregated Ethernet interfaces with member links in IQ2, IQ2-E, and IQ PICs or in MX Series DPCs, or on Ethernet interfaces for PTX Series Packet Transport Routers or 100-Gigabit Ethernet Type 5 PIC with CFP.

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

Related Documentation

- *Configuring Mixed Tagging*
- *Configuring Flexible VLAN Tagging on PTX Series Packet Transport Routers*

hierarchical-scheduler (Subscriber Interfaces on MX Series Routers)

Syntax

```
hierarchical-scheduler {
    implicit-hierarchy;
    maximum–hierarchy–levels number;
}
```

Hierarchy Level [edit interfaces *interface-name*]

Release Information Statement introduced in Junos OS Release 10.1.
implicit-hierarchy option introduced in Junos OS Release 13.1.

Description On MX Series routers with MPC/MICs, configure the parameters for hierarchical scheduling on the interface.

NOTE: To enable hierarchical scheduling on MX Series routers, configure the **hierarchical-scheduler** statement at each member physical interface level of a particular aggregated Ethernet interface as well as at that aggregated Ethernet interface level. On other routing platforms, it is enough if you include this statement at the aggregated Ethernet interface level.

Options

implicit-hierarchy—Configure three-level hierarchical scheduling. When you include the **implicit-hierarchy** option, a hierarchical relationship is formed between the CoS scheduler nodes at level 1, level 2, and level 3. The **implicit-hierarchy** option is supported only on MPC/MIC subscriber interfaces and interface sets running over aggregated Ethernet on MX Series routers.

maximum-hierarchy-levels ***number***—Configure two-level hierarchical scheduling. Specify the maximum number of hierarchical scheduling levels allowed for node scaling. The only supported value is 2. The **maximum-hierarchy-levels** option is supported on MPC/MIC or EQ DPC subscriber interfaces and interface sets running over aggregated Ethernet on MX Series routers.

- If you include the **maximum-hierarchy-levels** option, interface sets are allowed only at level 3; they are not allowed at level 2. In this case, if you configure a level 2 interface set, you generate Packet Forwarding Engine errors.
- If you do not include the **maximum-hierarchy-levels** option, interface sets can be at either level 2 or level 3, depending on whether the member logical interfaces within the interface set have a traffic control profile. If any member logical interface has a traffic control profile, then the interface set is a level 2 CoS scheduler node. If no member logical interface has a traffic control profile, the interface set is at level 3.

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

Related Documentation

- *Understanding Two-Level and Three-level Hierarchical CoS for Subscriber Interfaces on MX Series Routers*
- *Configuring Hierarchical CoS for a Subscriber Interface of Aggregated Ethernet Links*
- *Junos OS Class of Service Library for Routing Devices*
- *Configuring Hierarchical Schedulers for CoS*
- Hierarchical CoS on MPLS Pseudowire Subscriber Interfaces Overview on page 5

input-hierarchical-policer

Syntax input-hierarchical-policer *policer-name*;

Hierarchy Level [edit interfaces *interface-name* layer2-policer],
[edit interfaces *interface-name* unit *logical-unit-number* layer2-policer],

Release Information Statement introduced in Junos OS Release 9.5.

Description Apply a hierarchical policer to the Layer 2 input traffic for all protocol families at the physical or logical interface.

Options ***policer-name***—Name of the hierarchical policer.

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

Related Documentation

- *Hierarchical Policers*
- *layer2-policer (Hierarchical Policer)*

mtu

Syntax mtu *bytes*;

Hierarchy Level
[edit interfaces *interface-name*],
[edit interfaces *interface-name* unit *logical-unit-number* family *family*],
[edit interfaces interface-range *name*],
[edit logical-systems *logical-system-name* interfaces *interface-name* unit *logical-unit-number* family *family*],
[edit logical-systems *logical-system-name* protocols l2circuit local-switching interface *interface-name* backup-neighbor *address*],
[edit logical-systems *logical-system-name* protocols l2circuit neighbor *address* interface *interface-name*],
[edit logical-systems *logical-system-name* protocols l2circuit neighbor *address* interface *interface-name* backup-neighbor *address*],
[edit logical-systems *logical-system-name* routing-instances *routing-instance-name* protocols l2vpn interface *interface-name*],
[edit logical-systems *logical-system-name* routing-instances *routing-instance-name* protocols vpls],
[edit protocols l2circuit local-switching interface *interface-name* backup-neighbor *address*],
[edit protocols l2circuit neighbor *address* interface *interface-name*]
[edit protocols l2circuit neighbor *address* interface *interface-name* backup-neighbor *address*],
[edit routing-instances *routing-instance-name* protocols l2vpn interface *interface-name*],
[edit routing-instances *routing-instance-name* protocols vpls]

Release Information
Statement introduced before Junos OS Release 7.4.
Statement introduced in Junos OS Release 9.0 for EX Series switches.
Support for Layer 2 VPNs and VPLS introduced in Junos OS Release 10.4.
Statement introduced in Junos OS Release 12.1X48 for PTX Series Packet Transport Routers.
Statement introduced in Junos OS Release 12.2 for ACX Series Universal Access Routers.
Support at the **[set interfaces interface-name unit logical-unit-number family ccc]** hierarchy level introduced in Junos OS Release 12.3R3 for MX Series routers.

Description Specify the maximum transmission unit (MTU) size for the media or protocol. The default MTU size depends on the device type. Changing the media MTU or protocol MTU causes an interface to be deleted and added again.

To route jumbo data packets on the routed VLAN interface (RVI) on EX Series switches, you must configure the jumbo MTU size on the member physical interfaces and also on the RVI itself (the **vlan** interface).

CAUTION: For EX Series switches, setting or deleting the jumbo MTU size on the RVI (the vlan interface) while the switch is transmitting packets might cause packets to be dropped.

NOTE: If a packet whose size is larger than the configured MTU size is received on the receiving interface, the packet is eventually dropped. The value

considered for MRU (maximum receive unit) size is also the same as the MTU size configured on that interface.

NOTE: Not all devices allow you to set an MTU value, and some devices have restrictions on the range of allowable MTU values. You cannot configure an MTU for management Ethernet interfaces (fxp0, em0, or me0) or for loopback, multilink, and multicast tunnel devices.

For more information about configuring MTU for specific interfaces and router or switch combinations, see *Configuring the Media MTU*.

Options

bytes—MTU size.

Range: 256 through 9192 bytes, 256 through 9500 bytes (Junos OS 12.1X48R2 for PTX Series routers)

Default: 1500 bytes (INET, INET6, and ISO families), 1448 bytes (MPLS), 1514 bytes (EX Series switch interfaces)

Required Privilege Level

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

Related Documentation

- *Configuring Gigabit Ethernet Interfaces (CLI Procedure)*
- *Configuring the Media MTU*
- *Configuring the MTU for Layer 2 Interfaces*
- *Configuring Routed VLAN Interfaces (CLI Procedure)*
- *Setting the Protocol MTU*

no-gratuitous-arp-request

Syntax no-gratuitous-arp-request;

Hierarchy Level [edit interfaces *interface-name*]

Release Information Statement introduced in Junos OS Release 9.6 for EX Series switches.
Statement introduced in Junos OS Release 12.2 for ACX Series Universal Access Routers.

Description For Ethernet interfaces and pseudowire logical interfaces, do not respond to gratuitous ARP requests.

Default Gratuitous ARP responses are enabled on all Ethernet interfaces.

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

Related Documentation
- *Configuring Gratuitous ARP*
- *gratuitous-arp-reply*

no-vlan-id-validate

Syntax no-vlan-id-validate;

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

Release Information Statement introduced in Junos OS Release 13.1.

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

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

Related Documentation
- *Configuring Interfaces for Layer 2 Circuits*
- Pseudowire Subscriber Logical Interfaces Overview on page 3
- Configuring a Pseudowire Subscriber Logical Interface on page 13
- Configuring Layer 2 Circuit Signaling for Pseudowire Subscriber Logical Interfaces on page 17

ps0 (Pseudowire Subscriber Interfaces)

Syntax

```
ps0 {
  anchor-point lt-device;
  mtu bytes;
  no-gratuitous-arp-request;
  (flexible-vlan-tagging | stacked-vlan-tagging | untagged | vlan-tagging);
}
```

Hierarchy Level [edit logical-systems transport-ls interfaces]

Release Information Statement introduced in Junos OS Release 13.1.

Description Configure the pseudowire logical device.

The remaining statements are explained separately.

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

Related Documentation

- Pseudowire Subscriber Logical Interfaces Overview on page 3
- Configuring a Pseudowire Subscriber Logical Interface on page 13
- Configuring a Pseudowire Subscriber Logical Interface Device on page 15
- Configuring the Transport Logical Interface for a Pseudowire Subscriber Logical Interface on page 17
- Configuring the Service Logical Interface for a Pseudowire Subscriber Logical Interface on page 19

pseudowire-service (Pseudowire Subscriber Interfaces)

Syntax

```
pseudowire-service {
  device-count number;
}
```

Hierarchy Level [edit chassis]

Release Information Statement introduced in Junos OS Release 13.1.

Description Configure properties for the pseudowire devices on the router.

The remaining statement is explained separately.

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

Related Documentation

- Pseudowire Subscriber Logical Interfaces Overview on page 3
- Configuring a Pseudowire Subscriber Logical Interface on page 13
- Configuring the Maximum Number of Pseudowire Logical Interface Devices Supported on the Router on page 15

stacked-vlan-tagging

Syntax stacked-vlan-tagging;

Hierarchy Level [edit interfaces *interface-name*]

Release Information Statement introduced before Junos OS Release 7.4.
Statement introduced in Junos OS Release 12.2 for ACX Series Universal Access Routers.

Description For Gigabit Ethernet IQ interfaces, Gigabit Ethernet, 10-Gigabit Ethernet LAN/WAN PIC, and 100-Gigabit Ethernet Type 5 PIC with CFP, enable stacked VLAN tagging for all logical interfaces on the physical interface.

For pseudowire subscriber interfaces, enable stacked VLAN tagging for logical interfaces on the pseudowire service.

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

Related Documentation

- *Stacking and Rewriting Gigabit Ethernet VLAN Tags Overview*
- *vlan-tags (Stacked VLAN Tags)*

unit

Syntax

```
unit logical-unit-number {
  accept-source-mac {
    mac-address mac-address {
      policer {
        input cos-policer-name;
        output cos-policer-name;
      }
    }
  }
  accounting-profile name;
  advisory-options {
    downstream-rate rate;
    upstream-rate rate;
  }
  allow-any-vci;
  atm-scheduler-map (map-name | default);
  backup-options {
    interface interface-name;
  }
  bandwidth rate;
  cell-bundle-size cells;
  clear-dont-fragment-bit;
  compression {
    rtp {
      maximum-contexts number <force>;
      f-max-period number;
      queues [ queue-numbers ];
      port {
        minimum port-number;
        maximum port-number;
      }
    }
  }
  compression-device interface-name;
  copy-tos-to-outer-ip-header;
  demux-destination family;
  demux-source family;
  demux-options {
    underlying-interface interface-name;
  }
  description text;
  interface {
    l2tp-interface-id name;
    (dedicated | shared);
  }
  dialer-options {
    activation-delay seconds;
    callback;
    callback-wait-period time;
    deactivation-delay seconds;
    dial-string [ dial-string-numbers ];
    idle-timeout seconds;
```

```
    incoming-map {
      caller caller-id) | accept-all;
      initial-route-check seconds;
      load-interval seconds;
      load-threshold percent;
      pool pool-name;
      redial-delay time;
      watch-list {
        [ routes ];
      }
    }
}
disable;
disable-mlppp-inner-ppp-pfc;
dlci dlci-identifier;
drop-timeout milliseconds;
dynamic-call-admission-control {
  activation-priority priority;
  bearer-bandwidth-limit kilobits-per-second;
}
encapsulation type;
epd-threshold cells plp1 cells;
family family-name {
  ... the family subhierarchy appears after the main [edit interfaces interface-name unit
    logical-unit-number] hierarchy ...
}
fragment-threshold bytes;
inner-vlan-id-range start start-id end end-id;
input-vlan-map {
  (pop | pop-pop | pop-swap | push | push-push | swap |
  swap-push | swap-swap);
  inner-tag-protocol-id tpid;
  inner-vlan-id number;
  tag-protocol-id tpid;
  vlan-id number;
}
interleave-fragments;
inverse-arp;
layer2-policer {
  input-policer policer-name;
  input-three-color policer-name;
  output-policer policer-name;
  output-three-color policer-name;
}
link-layer-overhead percent;
minimum-links number;
mrru bytes;
multicast-dlci dlci-identifier;
multicast-vci vpi-identifier.vci-identifier;
multilink-max-classes number;
multipoint;
oam-liveness {
  up-count cells;
  down-count cells;
}
oam-period (disable | seconds);
```

```
output-vlan-map {
  (pop | pop-pop | pop-swap | push | push-push | swap |
  swap-push | swap-swap);
  inner-tag-protocol-id tpid;
  inner-vlan-id number;
  tag-protocol-id tpid;
  vlan-id number;
}
passive-monitor-mode;
peer-unit unit-number;
plp-to-clp;
point-to-point;
ppp-options {
  chap {
    access-profile name;
    default-chap-secret name;
    local-name name;
    passive;
  }
  compression {
    acfc;
    pfc;
  }
  dynamic-profile profile-name;
  lcp-restart-timer milliseconds;
  loopback-clear-timer seconds;
  ncp-restart-timer milliseconds;
  pap {
    access-profile name;
    default-pap-password password;
    local-name name;
    local-password password;
    passive;
  }
}
pppoe-options {
  access-concentrator name;
  auto-reconnect seconds;
  (client | server);
  service-name name;
  underlying-interface interface-name;
}
pppoe-underlying-options {
  access-concentrator name;
  dynamic-profile profile-name;
  max-sessions number;
}
proxy-arp;
service-domain (inside | outside);
shaping {
  (cbr rate | rtvbr peak rate sustained rate burst length | vbr peak rate sustained rate burst
    length);
  queue-length number;
}
short-sequence;
targeted-distribution;
```

```
transmit-weight number;
(traps | no-traps);
trunk-bandwidth rate;
trunk-id number;
tunnel {
  backup-destination address;
  destination address;
  key number;
  routing-instance {
    destination routing-instance-name;
  }
  source source-address;
  ttl number;
}
vci vpi-identifier.vci-identifier;
vci-range start start-vci end end-vci;
vpi vpi-identifier;
vlan-id number;
vlan-id-range number-number;
vlan-tags inner tpid.vlan-id outer tpid.vlan-id;
family family {
  accounting {
    destination-class-usage;
    source-class-usage {
      (input | output | input output);
    }
  }
  access-concentrator name;
  address address {
    ... the address subhierarchy appears after the main [edit interfaces interface-name unit
      logical-unit-number family family-name] hierarchy ...
  }
  bridge-domain-type (bvlan | svlan);
  bundle interface-name;
  core-facing;
  demux-destination {
    destination-prefix;
  }
  demux-source {
    source-prefix;
  }
  duplicate-protection;
  dynamic-profile profile-name;
  filter {
    group filter-group-number;
    input filter-name;
    input-list [ filter-names ];
    output filter-name;
    output-list [ filter-names ];
  }
  interface-mode (access | trunk);
  ipsec-sa sa-name;
  isid-list all-service-groups;
  keep-address-and-control;
  mac-validate (loose | strict);
  max-sessions number;
```

```
mtu bytes;
multicast-only;
no-redirects;
policer {
  arp policer-template-name;
  input policer-template-name;
  output policer-template-name;
}
primary;
protocols [inet iso mpls];
proxy inet-address address;
receive-options-packets;
receive-ttl-exceeded;
remote (inet-address address | mac-address address);
rpf-check {
  fail-filter filter-name
  mode loose;
}
sampling {
  input;
  output;
}
service {
  input {
    post-service-filter filter-name;
    service-set service-set-name <service-filter filter-name>;
  }
  output {
    service-set service-set-name <service-filter filter-name>;
  }
}
service-name-table table-name
(translate-discard-eligible | no-translate-discard-eligible);
(translate-fecn-and-becn | no-translate-fecn-and-becn);
translate-plp-control-word-de;
unnumbered-address interface-name destination address destination-profile profile-name;
vlan-id number;
vlan-id-list [number number-number];
address address {
  arp ip-address (mac | multicast-mac) mac-address <publish>;
  broadcast address;
  destination address;
  destination-profile name;
  eui-64;
  master-only;
  multipoint-destination address {
    dlci dlci-identifier;
    epd-threshold cells <plp1 cells>;
    inverse-arp;
    oam-liveness {
      up-count cells;
      down-count cells;
    }
    oam-period (disable | seconds);
    shaping {
```

```
            (cbr rate | rtvbr burst length peak rate sustained rate | vbr burst length peak rate
              sustained rate);
            queue-length number;
          }
          vci vpi-identifier.vci-identifier;
        }
        preferred;
        primary;
        (vrrp-group | vrrp-inet6-group) group-number {
          (accept-data | no-accept-data);
          advertise–interval seconds;
          authentication-type authentication;
          authentication-key key;
          fast-interval milliseconds;
          (preempt | no-preempt) {
            hold-time seconds;
          }
          priority number;
          track {
            interface interface-name {
              bandwidth-threshold bits-per-second priority-cost number;
            }
            priority-hold-time seconds;
            route ip-address/prefix-length routing-instance instance-name priority-cost cost;
          }
          virtual-address [ addresses ];
          virtual-link-local-address ipv6–address;
          vrrp-inherit-from {
            active-interface interface-name;
            active-group group-number;
          }
        }
      }
    }
  }
}
```

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

Release Information Statement introduced before Junos OS Release 7.4.

Description Configure a logical interface on the physical device. You must configure a logical interface to be able to use the physical device.

Options ***logical-unit-number***—Number of the logical unit.

Range: 0 through 1,073,741,823 for demux and PPPoE static interfaces only. 0 through 16,385 for all other static interface types.

The remaining statements are explained separately.

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

Related Documentation

- *Configuring Logical Interface Properties*
- *Example: Configuring E-LINE and E-LAN Services for a PBB Network on MX Series Routers*
- *Junos OS Services Interfaces Library for Routing Devices*

untagged

Syntax untagged;

Hierarchy Level [edit interfaces ps0]

Release Information Statement introduced in Junos OS Release 13.1.

Description Specify that the router supports untagged traffic on pseudowire subscriber interfaces.

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

Related Documentation

- Configuring a Pseudowire Subscriber Logical Interface Device on page 15

vlan-tagging

Syntax vlan-tagging;

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

Release Information Statement introduced before Junos OS Release 7.4.
Statement introduced in Junos OS Release 9.0 for EX Series switches.
Statement introduced in Junos OS Release 12.2 for ACX Series Universal Access Routers.
Statement introduced in Junos OS Release 13.2 for PTX Series Routers.

Description For Fast Ethernet and Gigabit Ethernet interfaces, aggregated Ethernet interfaces configured for VPLS, and pseudowire subscriber interfaces, enable the reception and transmission of 802.1Q VLAN-tagged frames on the interface.

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

Related Documentation

- *Example: Configuring Layer 3 Subinterfaces for a Distribution Switch and an Access Switch*
- *Example: Configuring BGP Autodiscovery for LDP VPLS*
- *Configuring a Layer 3 Subinterface (CLI Procedure)*
- *Configuring Tagged Aggregated Ethernet Interfaces*
- *Configuring Interfaces for VPLS Routing*
- *Enabling VLAN Tagging*
- *802.1Q VLANs Overview*
- *vlan-id*

PART 3

Administration

CHAPTER 5

Monitoring Commands

show interfaces ps0 (Pseudowire Subscriber Interfaces)

Syntax show interfaces ps0
<brief | detail | extensive | terse>

Release Information Command introduced at Junos OS Release 13.1.

Description Display status information about the pseudowire subscriber interface.

Options **brief | detail | extensive | terse**—(Optional) Display the specified level of output.

Required Privilege Level view

Related Documentation
- Pseudowire Subscriber Logical Interfaces Overview on page 3

List of Sample Output show interfaces ps0 on page 56

Output Fields Table 5 on page 54 lists the output fields for the **show interfaces ps0** command. Output fields are listed in the approximate order in which they appear.

Table 5: show interfaces ps0 Output Fields

Field Name	Field Description	Level of Output
Physical Interface		
Physical interface	Name of the physical interface.	**brief detail extensive** none
Enabled	State of the interface. Possible values are described in the "Enabled Field" section under *Common Output Fields Description*.	**brief detail extensive** none
Interface index	Physical interface index number, which reflects its initialization sequence.	**detail extensive** none
SNMP ifIndex	SNMP index number for the physical interface.	**detail extensive** none
Type	Physical interface type (Software-Pseudo).	**brief detail extensive** none
Link-level type	Encapsulation being used on the physical interface.	**brief detail extensive**
MTU	MTU size on the physical interface.	**brief detail extensive**
Clocking	Reference clock source. It can be **Internal** or **External**.	**brief detail extensive**
Speed	Speed at which the interface is running.	**brief detail extensive**
Device flags	Information about the physical device. Possible values are described in the "Device Flags" section under *Common Output Fields Description*.	**brief detail extensive** none

Table 5: show interfaces ps0 Output Fields *(continued)*

Field Name	Field Description	Level of Output
Interface flags	Information about the interface. Possible values are described in the "Interface Flags" section under *Common Output Fields Description*.	**brief detail extensive** none
Current address	Configured MAC address.	**detail extensive** none
Hardware address	MAC address of the hardware.	**detail extensive** none
Last flapped	Date, time, and how long ago the interface went from down to up or up to down. The format is Last flapped: *year-month-day hours:minutes:seconds*: *timezone* (*hours:minutes:seconds* ago). or Never. For example, Last flapped: 2002-04-26 10:52:40 PDT (04:33:20 ago).	**detail extensive** none
input packets	Number of packets received on the logical interface.	**detail extensive** none
output packets	Number of packets transmitted on the logical interface.	**detail extensive** none
Logical Interface		
Logical interface	Name of the logical interface.	**brief detail extensive** none
Index	Logical interface index number (which reflects its initialization sequence).	**detail extensive** none
SNMP ifIndex	Logical interface SNMP interface index number.	**detail extensive** none
Flags	Information about the logical interface. Possible values are described in the "Logical Interface Flags" section under *Common Output Fields Description*.	**brief detail extensive** none
Encapsulation	Type of encapsulation configured on the logical interface.	**brief extensive** none
Input packets	Number of packets received on the logical interface.	none
Output packets	Number of packets transmitted on the logical interface.	none
Protocol	Protocol family configured on the logical interface.	**detail extensive** none
MTU	MTU size on the logical interface.	**detail extensive** none
Flags	Information about the protocol family flags. Possible values are described in the "Family Flags" section under *Common Output Fields Description*.	**detail extensive** none
Addresses, Flags	Information about the addresses configured for the protocol family. Possible values are described in the "Addresses Flags" section under *Common Output Fields Description*.	**detail extensive** none
Destination	IP address of the remote side of the connection.	**detail extensive** none
Local	IP address of the logical interface.	**detail extensive terse** none

Table 5: show interfaces ps0 Output Fields *(continued)*

Field Name	Field Description	Level of Output
Broadcast	Broadcast address.	**detail extensive** none

Sample Output

show interfaces ps0

```
user@host> show interfaces ps0
Physical interface: ps0, Enabled, Physical link is Up
 Interface index: 166, SNMP ifIndex: 658
 Type: Software-Pseudo, Link-level type: 90, MTU: 1518, Clocking: 1, Speed: 800mbps

 Device flags : Present Running
 Interface flags: Point-To-Point Internal: 0x4000
 Current address: 00:1d:b5:a8:19:4a, Hardware address: 00:1d:b5:a8:19:4a
 Last flapped : Never
  Input packets : 0
  Output packets: 0

 Logical interface ps0.0 (Index 74) (SNMP ifIndex 656)
  Flags: Point-To-Point 0x4000 Encapsulation: Ethernet-CCC
  Input packets : 482
  Output packets: 0
  Protocol ccc, MTU: 1518
   Flags: Is-Primary

 Logical interface ps0.1 (Index 78) (SNMP ifIndex 665)
  Flags: Point-To-Point 0x4000 VLAN-Tag [ 0x8100.100 ] Encapsulation: ENET2
  Input packets : 0
  Output packets: 482
  Protocol inet, MTU: 1500
   Flags: Sendbcast-pkt-to-re
    Addresses, Flags: Is-Preferred Is-Primary
     Destination: 20.0.0/24, Local: 20.0.0.1, Broadcast: 20.0.0.255

 Logical interface ps0.32767 (Index 75) (SNMP ifIndex 692)
  Flags: Point-To-Point 0x4000 VLAN-Tag [ 0x0000.0 ] Encapsulation: ENET2
  Input packets : 0
  Output packets: 0
```

PART 4

Troubleshooting

CHAPTER 6

Acquiring Troubleshooting Information

Collecting Subscriber Access Logs Before Contacting Juniper Technical Support

Problem When you experience a subscriber access problem in your network, we recommend that you collect certain logs before you contact Juniper Technical Support. This topic shows you the most useful logs for a variety of network implementations. In addition to the relevant log information, you must also collect standard troubleshooting information and send it to Juniper Technical Support in your request for assistance.

Solution To collect standard troubleshooting information:

- Redirect the command output to a file.

  user@host> **request support information | save rsi-1**

To configure logging to assist Juniper Technical Support:

1. Review the following blocks of statements to determine which apply to your configuration.

```
[edit]
set system syslog archive size 100m files 25
set system auto-configuration traceoptions file filename
set system auto-configuration traceoptions file filename size 100m files 25
set protocols ppp-service traceoptions file filename size 100m files 25
set protocols ppp-service traceoptions level all
set protocols ppp-service traceoptions flag all
set protocols ppp traceoptions file filename size 100m files 25
set protocols ppp traceoptions level all
set protocols ppp traceoptions flag all
set protocols ppp monitor-session all
set interfaces pp0 traceoptions flag all
set demux traceoptions file filename size 100m files 25
set demux traceoptions level all
set demux traceoptions flag all
set system processes dhcp-service traceoptions file filename
set system processes dhcp-service traceoptions file size 100m
set system processes dhcp-service traceoptions file files 25
set system processes dhcp-service traceoptions flag all
set class-of-service traceoptions file filename
set class-of-service traceoptions file size 100m
set class-of-service traceoptions flag all
set class-of-service traceoptions file files 25
set routing-options traceoptions file filename
set routing-options traceoptions file size 100m
set routing-options traceoptions flag all
set routing-options traceoptions file files 25
set interfaces traceoptions file filename
set interfaces traceoptions file size 100m
set interfaces traceoptions flag all
set interfaces traceoptions file files 25
set system processes general-authentication-service traceoptions file filename
set system processes general-authentication-service traceoptions file size 100m
set system processes general-authentication-service traceoptions flag all
set system processes general-authentication-service traceoptions file files 25
```

2. Copy the relevant statements into a text file and modify the log filenames as you want.
3. Copy the statements from the text file and paste them into the CLI on your router to configure logging.
4. Commit the logging configuration to begin collecting information.

NOTE: The maximum file size for DHCP local server and DHCP relay log files is 1 GB. The maximum number of log files for DHCP local server and DHCP relay is 1000.

BEST PRACTICE: Enable these logs only to collect information when troubleshooting specific problems. Enabling these logs during normal operations can result in reduced system performance.

Related Documentation

- *Compressing Troubleshooting Logs from /var/logs to Send to Juniper Technical Support*

PART 5

Index

Index