Junos® OS

Interfaces Fundamentals for Routing Devices

Published
2020-06-26

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

YEAR 2000 NOTICE

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

END USER LICENSE AGREEMENT

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

Table of Contents

2

Other Interfaces

4

Configuration Statements

About the Documentation

IN THIS SECTION

Use this guide to configure, monitor and troubleshoot various interfaces installed on a Juniper Networks router with the Junos OS command-line interface (CLI).

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 https://www.juniper.net/documentation/.

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 https://www.juniper.net/books.

Using the Examples in This Manual

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

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

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

Merging a Full Example

To merge a full example, follow these steps:

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

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

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

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

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

Merging a Snippet

To merge a snippet, follow these steps:

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

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

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

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

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

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

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

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

Documentation Conventions

Table 1 on page xix 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.
	Tip	Indicates helpful information.
	Best practice	Alerts you to a recommended use or implementation.

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

Table 2: Text and Syntax Conventions

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

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

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

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

; (semicolon)	Identifies a leaf statement at a configuration hierarchy level.	
GUI Conventions		

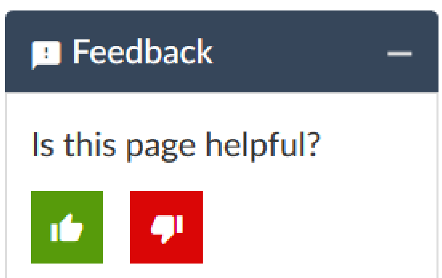
Table 2: Text and Syntax Conventions *(continued)*

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

Documentation Feedback

We encourage you to provide feedback so that we can improve our documentation. You can use either of the following methods:

- Online feedback system—Click TechLibrary Feedback, on the lower right of any page on the Juniper Networks TechLibrary site, and do one of the following:



  - Click the thumbs-up icon if the information on the page was helpful to you.
  - Click the thumbs-down icon if the information on the page was not helpful to you or if you have suggestions for improvement, and use the pop-up form to provide feedback.

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

Requesting Technical Support

Technical product support is available through the Juniper Networks Technical Assistance Center (JTAC). If you are a customer with an active Juniper Care or Partner Support Services 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 https://www.juniper.net/us/en/local/pdf/resource-guides/7100059-en.pdf.
- Product warranties—For product warranty information, visit https://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: https://www.juniper.net/customers/support/
- Search for known bugs: https://prsearch.juniper.net/
- Find product documentation: https://www.juniper.net/documentation/
- Find solutions and answer questions using our Knowledge Base: https://kb.juniper.net/
- Download the latest versions of software and review release notes: https://www.juniper.net/customers/csc/software/
- Search technical bulletins for relevant hardware and software notifications: https://kb.juniper.net/InfoCenter/
- Join and participate in the Juniper Networks Community Forum: https://www.juniper.net/company/communities/
- Create a service request online: https://myjuniper.juniper.net

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

Creating a Service Request with JTAC

You can create a service request with JTAC on the Web or by telephone.

- Visit https://myjuniper.juniper.net.
- 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 https://support.juniper.net/support/requesting-support/.

1

CHAPTER

Router Interfaces

Router Interfaces Overview

IN THIS SECTION

The interfaces on a router provide network connectivity to the router. This topic discusses about the various router interfaces supported on Junos like, transient interfaces, services interfaces, container interfaces, and internal ethernet interfaces. This topic also provides basic interface related information like, interface naming conventions, overview of interface encapsulation, and overview of interface descriptors.

Router Interfaces Overview

Routers typically contain several different types of interfaces suited to various functions. For the interfaces on a router to function, you must configure them. Specify the interface location (that is, the slot where the Flexible PIC Concentrator [FPC], Dense Port Concentrator [DPC], or Modular Port Concentrator [MPC] is installed. You must also specify the location of the Physical Interface Card [PIC] or Modular Interface Card [MIC] , and the interface type, for example, SONET/SDH, Asynchronous Transfer Mode [ATM], or Ethernet). Finally, you must specify the encapsulation type and any interface-specific properties that may apply.

You can configure interfaces that are currently present in the router, as well as interfaces that are not currently present but that are expected to be added in the future. Junos OS detects the interface once the hardware has been installed and applies the pre-set configuration to it.

To see which interfaces are currently installed in the router, issue the **show interfaces terse** operational mode command. If an interface is listed in the output, it is physically installed in the router. If an interface is not listed in the output, it is not installed in the router.

For information about which interfaces are supported on your router, see your router's *Interface Module Reference*.

You can configure Junos OS class-of-service (CoS) properties to provide a variety of classes of service for different applications, including multiple forwarding classes for managing packet transmission, congestion management, and CoS-based forwarding. For more information about configuring CoS properties, see the *Class of Service User Guide (Routers and EX9200 Switches)*.

Types of Interfaces

Interfaces can be permanent or transient, and are used for networking or services:

- Permanent interfaces—Interfaces that are always present in the router.

  Permanent interfaces in the router consist of management Ethernet interfaces and internal Ethernet interfaces, which are described separately in the following topics:

  - *Understanding Management Ethernet Interfaces*
  - Understanding Internal Ethernet Interfaces on page 65

- Transient interfaces—Interfaces that can be inserted into or removed from the router depending on your network configuration needs.
- Networking interfaces—Interfaces, such as Ethernet or SONET/SDH interfaces, that primarily provide traffic connectivity.

- Services interfaces—Interfaces that provide specific capabilities for manipulating traffic before it is delivered to its destination.
- Container interfaces—Interfaces that support automatic protection switching (APS) on physical SONET links using a virtual container infrastructure.

Junos OS internally generates nonconfigurable interfaces which are described in *Interfaces Command Reference* and *Services Interfaces*.

SEE ALSO

ATM Interfaces Overview
Channelized Interfaces Overview
Circuit Emulation Interfaces: *Understanding Mobile Backhaul*
E1 Interfaces Overview and *E3 Interfaces Overview*
Ethernet Interfaces Overview
Frame Relay Overview
SONET/SDH Interfaces Overview
T1 Interfaces Overview and *T3 Interfaces Overview*

Interface Naming Overview

IN THIS SECTION

Each interface has an interface name, which specifies the media type, the slot in which the FPC or DPC is located, the location on the FPC where the PIC is installed, and the PIC or DPC port. The interface name uniquely identifies an individual network connector in the system. You use the interface name when configuring interfaces and when enabling various functions and properties, such as routing protocols, on individual interfaces. The system uses the interface name when displaying information about the interface, for example, in the **show interfaces** command.

The interface name is represented by a physical part, a channel part, and a logical part in the following format:

```
physical<:channel>.logical
```

The channel part of the name is optional for all interfaces except channelized DS3, E1, OC12, and STM1 interfaces.

The EX Series, QFX Series, NFX Series, OCX1100, QFabric System, and EX4600 devices use a naming convention for defining the interfaces that are similar to that of other platforms running under Juniper Networks Junos OS. For more information, see *Understanding Interface Naming Conventions*.

The following sections provide interface naming configuration guidelines:

Physical Part of an Interface Name

The physical part of an interface name identifies the physical device, which corresponds to a single physical network connector.

NOTE:

The internal interface is dependent on the Routing Engine. To identify if the Routing Engine is using this type of interface, use the following command:

`user@host>` **show interfaces terse**

```
Interface               Admin Link Proto    Local                 Remote
pfe-1/0/0               up    up
pfe-1/0/0.16383         up    up   inet
                                   inet6
pfh-1/0/0               up    up
pfh-1/0/0.16383         up    up   inet
[..........]
bcm0                    up    up <----------------
bcm0.0                  up    up   inet 10.0.0.1/8
[..........]
lsi                     up    up
mtun                    up    up
pimd                    up    up
pime                    up    up
tap                     up    up
```

For more information on the Routing Engines that each chassis supports, the first supported release for the Routing Engine in the specified chassis, the management Ethernet interface, and the internal Ethernet interfaces for each Routing Engine, please refer the link titled *Supported Routing Engines by Chassis* under Related Documentation section.

This part of the interface name has the following format:

type-fpc/pic/port

type is the media type, which identifies the network device that can be one of the following:

- **ae**—Aggregated Ethernet interface. This is a virtual aggregated link and has a different naming format from most PICs; for more information, see *Aggregated Ethernet Interfaces Overview*.
- **as**—Aggregated SONET/SDH interface. This is a virtual aggregated link and has a different naming format from most PICs; for more information, see *Configuring Aggregated SONET/SDH Interfaces*.
- **at**—ATM1 or ATM2 intelligent queuing (IQ) interface or a virtual ATM interface on a circuit emulation (CE) interface.

- **bcm**—The bcm0 internal Ethernet process is supported on specific Routing engines for various M series and T series routers. For more information please refer the link titled *Supported Routing Engines by Chassis* under Related Documentation section.
- **cau4**—Channelized AU-4 IQ interface (configured on the Channelized STM1 IQ or IQE PIC or Channelized OC12 IQ and IQE PICs).
- **ce1**—Channelized E1 IQ interface (configured on the Channelized E1 IQ PIC or Channelized STM1 IQ or IQE PIC).
- **ci**—Container interface.
- **coc1**—Channelized OC1 IQ interface (configured on the Channelized OC12 IQ and IQE or Channelized OC3 IQ and IQE PICs).
- **coc3**—Channelized OC3 IQ interface (configured on the Channelized OC3 IQ and IQE PICs).
- **coc12**—Channelized OC12 IQ interface (configured on the Channelized OC12 IQ and IQE PICs).
- **coc48**—Channelized OC48 interface (configured on the Channelized OC48 and Channelized OC48 IQE PICs).
- **cp**—Collector interface (configured on the Monitoring Services II PIC).
- **cstm1**—Channelized STM1 IQ interface (configured on the Channelized STM1 IQ or IQE PIC).
- **cstm4**—Channelized STM4 IQ interface (configured on the Channelized OC12 IQ and IQE PICs).
- **cstm16**—Channelized STM16 IQ interface (configured on the Channelized OC48/STM16 and Channelized OC48/STM16 IQE PICs).
- **ct1**—Channelized T1 IQ interface (configured on the Channelized DS3 IQ and IQE PICs, Channelized OC3 IQ and IQE PICs, Channelized OC12 IQ and IQE PICs, or Channelized T1 IQ PIC).
- **ct3**—Channelized T3 IQ interface (configured on the Channelized DS3 IQ and IQE PICs, Channelized OC3 IQ and IQE PICs, or Channelized OC12 IQ and IQE PICs).
- **demux**—Interface that supports logical IP interfaces that use the IP source or destination address to demultiplex received packets. Only one demux interface (**demux0**) exists per chassis. All demux logical interfaces must be associated with an underlying logical interface.
- **dfc**—Interface that supports dynamic flow capture processing on T Series or M320 routers containing one or more Monitoring Services III PICs. Dynamic flow capture enables you to capture packet flows on the basis of dynamic filtering criteria. Specifically, you can use this feature to forward passively monitored packet flows that match a particular filter list to one or more destinations using an on-demand control protocol.
- **ds**—DS0 interface (configured on the Multichannel DS3 PIC, Channelized E1 PIC, Channelized OC3 IQ and IQE PICs, Channelized OC12 IQ and IQE PICs, Channelized DS3 IQ and IQE PICs, Channelized E1 IQ PIC, Channelized STM1 IQ or IQE PIC, or Channelized T1 IQ).
- **dsc**—Discard interface.
- **e1**—E1 interface (including channelized STM1-to-E1 interfaces).

- **e3**—E3 interface (including E3 IQ interfaces).
- **em**—Management and internal Ethernet interfaces. For M Series routers, MX Series routers, T Series routers, and TX Series routers, you can use the **show chassis hardware** command to display hardware information about the router, including its Routing Engine model. To determine which management interface is supported on your router and Routing Engine combination, see *Understanding Management Ethernet Interfaces* and *Supported Routing Engines by Router*.
- **es**—Encryption interface.
- **et**—100-Gigabit Ethernet interfaces (10, 40, and 100-Gigabit Ethernet interface for PTX Series Packet Transport Routers only).
- **fe**—Fast Ethernet interface.
- **fxp**—Management and internal Ethernet interfaces. For M Series routers, MX Series routers, T Series routers, and TX Series routers, you can use the **show chassis hardware** command to display hardware information about the router, including its Routing Engine model. To determine which management interface is supported on your router and Routing Engine combination, see *Understanding Management Ethernet Interfaces* and *Supported Routing Engines by Router*.
- **ge**—Gigabit Ethernet interface.

NOTE:

- The XENPAK 10-Gigabit Ethernet interface PIC, which is supported only on M series routers, is configured using the **ge** interface naming convention instead of the **xe** interface naming convention. Refer the following show commands for more information:

  user@host> **show chassis hardware**

  ```
  ..
  FPC 4            REV 02   710-015839   CZ1853            M120 FPC Type
   3
    PIC 0          REV 09   750-009567   NH1857            1x
  10GE(LAN),XENPAK
      Xcvr 0       REV 01   740-012045   535TFZX6          XENPAK-SR
  ```

  user@host> **show configuration interfaces ge-4/0/0**

  ```
  unit 0 {
      family inet {
          address 100.0.0.1/24;
      }
  }
  ```

- In MX and SRX series devices, the 1 and 10-Gigabit SFP or SFP+ optical interfaces are always named as **xe** even if a 1-Gigabit SFP is inserted. However, in EX and QFX series devices, the interface name is shown as **ge** or **xe** based on the speed of the optical device inserted.

- **gr**—Generic routing encapsulation (GRE) tunnel interface.
- **gre**—Internally generated interface that is configurable only as the control channel for Generalized MPLS (GMPLS). For more information about GMPLS, see the *MPLS Applications User Guide*.

NOTE: You can configure GRE interfaces (gre-x/y/z) only for GMPLS control channels. GRE interfaces are not supported or configurable for other applications..

- **ip**—IP-over-IP encapsulation tunnel interface.
- **ipip**—Internally generated interface that is not configurable.
- **ixgbe**—The internal Ethernet process ixgbe0 and ixgbe1 are used by the RE-DUO-C2600-16G Routing Engine, which is supported on TX Matrix Plus and PTX5000.

- **iw**—Logical interfaces associated with the endpoints of Layer 2 circuit and Layer 2 VPN connections (pseudowire stitching Layer 2 VPNs). For more information about VPNs, see the *Junos OS VPNs Library for Routing Devices*.
- **lc**—Internally generated interface that is not configurable.
- **lo**—Loopback interface. The Junos OS automatically configures one loopback interface (**lo0**). The logical interface **lo0.16383** is a nonconfigurable interface for router control traffic.
- **ls**—Link services interface.
- **lsi**—Internally generated interface that is not configurable.
- **ml**—Multilink interface (including Multilink Frame Relay and MLPPP).
- **mo**—Monitoring services interface (including monitoring services and monitoring services II). The logical interface **mo-*fpc/pic/port*.16383** is an internally generated, nonconfigurable interface for router control traffic.
- **ms**—Multiservices interface.
- **mt**—Multicast tunnel interface (internal router interface for VPNs). If your router has a Tunnel PIC, the Junos OS automatically configures one multicast tunnel interface (**mt**) for each virtual private network (VPN) you configure. Although it is not necessary to configure multicast interfaces, you can use the **multicast-only** statement to configure the unit and family so that the tunnel can transmit and receive multicast traffic only. For more information, see *multicast-only*.
- **mtun**—Internally generated interface that is not configurable.
- **oc3**—OC3 IQ interface (configured on the Channelized OC12 IQ and IQE PICs or Channelized OC3 IQ and IQE PICs).
- **pd**—Interface on the rendezvous point (RP) that de-encapsulates packets.
- **pe**—Interface on the first-hop PIM router that encapsulates packets destined for the RP router.
- **pimd**—Internally generated interface that is not configurable.
- **pime**—Internally generated interface that is not configurable.
- **rlsq**—Container interface, numbered from 0 through 127, used to tie the primary and secondary LSQ PICs together in high availability configurations. Any failure of the primary PIC results in a switch to the secondary PIC and vice versa.
- **rms**—Redundant interface for two multiservices interfaces.
- **rsp**—Redundant virtual interface for the adaptive services interface.
- **se**—Serial interface (including EIA-530, V.35, and X.21 interfaces).
- **si**—Services-inline interface, which is hosted on a Trio-based line card.
- **so**—SONET/SDH interface.

- **sp**—Adaptive services interface. The logical interface **sp-*fpc/pic/port*.16383** is an internally generated, nonconfigurable interface for router control traffic.
- **stm1**—STM1 interface (configured on the OC3/STM1 interfaces).
- **stm4**—STM4 interface (configured on the OC12/STM4 interfaces).
- **stm16**—STM16 interface (configured on the OC48/STM16 interfaces).
- **t1**—T1 interface (including channelized DS3-to-DS1 interfaces).
- **t3**—T3 interface (including channelized OC12-to-DS3 interfaces).
- **tap**—Internally generated interface that is not configurable.
- **umd**—USB modem interface.
- **vsp**—Voice services interface.
- **vc4**—Virtually concatenated interface.
- **vt**—Virtual loopback tunnel interface.
- **xe**—10-Gigabit Ethernet interface. Some older 10-Gigabit Ethernet interfaces use the **ge** media type (rather than **xe**) to identify the physical part of the network device.
- **xt**—Logical interface for Protected System Domains to establish a Layer 2 tunnel connection.

fpc identifies the number of the FPC or DPC card on which the physical interface is located. Specifically, it is the number of the slot in which the card is installed.

M40, M40e, M160, M320, M120, T320, T640, and T1600 routers each have eight FPC slots that are numbered 0 through 7, from left to right as you are facing the front of the chassis. For information about compatible FPCs and PICs, see the hardware guide for your router.

On PTX1000 routers, the FPC number is always 0.

The M20 router has four FPC slots that are numbered 0 through 3, from top to bottom as you are facing the front of the chassis. The slot number is printed adjacent to each slot.

MX Series routers support DPCs, FPCs, and Modular Interface Cards (MICs). For information about compatible DPCs, FPCs, PICs, and MICs, see the *MX Series Interface Module Reference*.

For M5, M7i, M10, and M10i routers, the FPCs are built into the chassis; you install the PICs into the chassis.

The M5 and M7i routers have space for up to four PICs. The M7i router also comes with an integrated Tunnel PIC, or an optional integrated AS PIC, or an optional integrated MS PIC.

The M10 and M10i routers have space for up to eight PICs.

A routing matrix can have up to 32 FPCs (numbered 0 through 31).

For more information about interface naming for a routing matrix, see "Interface Naming for a Routing Matrix Based on a TX Matrix Router" on page 35.

pic identifies the number of the PIC on which the physical interface is located. Specifically, it is the number of the PIC location on the FPC. FPCs with four PIC slots are numbered 0 through 3. FPCs with three PIC slots are numbered 0 through 2. The PIC location is printed on the FPC carrier board. For PICs that occupy more than one PIC slot, the lower PIC slot number identifies the PIC location.

port identifies a specific port on a PIC or DPC. The number of ports varies depending on the PIC. The port numbers are printed on the PIC.

Logical Part of an Interface Name

The logical unit part of the interface name corresponds to the logical unit number. The range of number available varies for different interface types. See *unit* for current range values.

In the virtual part of the name, a period (.) separates the port and logical unit numbers:

- Other platforms:

```
type-fpc/pic/port.logical
```

Separators in an Interface Name

There is a separator between each element of an interface name.

In the physical part of the name, a hyphen (-) separates the media type from the FPC number, and a slash (/) separates the FPC, PIC, and port numbers.

In the virtual part of the name, a period (.) separates the channel and logical unit numbers.

A colon (:) separates the physical and virtual parts of the interface name.

Channel Part of an Interface Name

The channel identifier part of the interface name is required only on channelized interfaces. For channelized interfaces, channel 0 identifies the first channelized interface. For channelized IQ and channelized IQE interfaces, channel 1 identifies the first channelized interface. A nonconcatenated (that is, channelized) SONET/SDH OC48 interface has four OC12 channels, numbered 0 through 3.

To determine which types of channelized PICs are currently installed in the router, use the **show chassis hardware** command from the top level of the command-line interface (CLI). Channelized IQ and IQE PICs are listed in the output with "intelligent queuing IQ" or "enhanced intelligent queuing IQE" in the description. For more information, see *Channelized Interfaces Overview*.

For ISDN interfaces, you specify the B-channel in the form **bc-*pim*/0/*port*:*n***. ***n*** is the B-channel ID and can be 1 or 2. You specify the D-channel in the form **dc-*pim*/0/*port*:0**.

NOTE: For ISDN, the B-channel and D-channel interfaces do not have any configurable parameters. However, when interface statistics are displayed, B-channel and D-channel interfaces have statistical values.

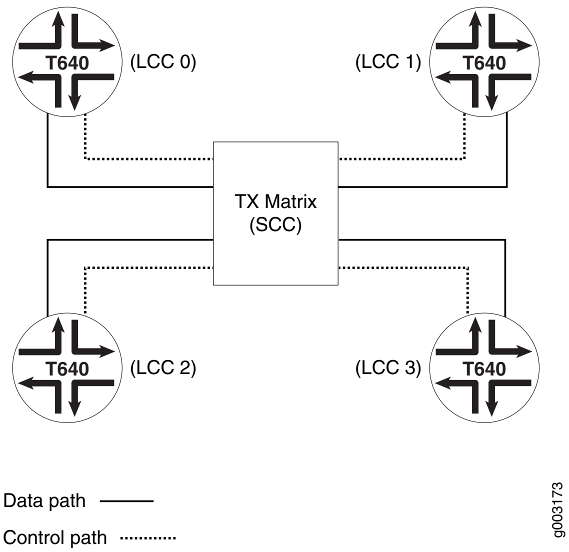
NOTE: In the Junos OS implementation, the term *logical interfaces* generally refers to interfaces you configure by including the **unit** statement at the **[edit interfaces *interface-name*]** hierarchy level. Logical interfaces have the ***.logical*** descriptor at the end of the interface name, as in **ge-0/0/0.1** or **t1-0/0/0:0.1**, where the logical unit number is **1**.

Although channelized interfaces are generally thought of as logical or virtual, the Junos OS sees T3, T1, and NxDS0 interfaces within a channelized IQ or IQE PIC as physical interfaces. For example, both **t3-0/0/0** and **t3-0/0/0:1** are treated as physical interfaces by the Junos OS. In contrast, **t3-0/0/0.2** and **t3-0/0/0:1.2** are considered logical interfaces because they have the **.2** at the end of the interface names.

Interface Naming for a Routing Matrix Based on a TX Matrix Router

A routing matrix based on a Juniper Networks TX Matrix router is a multichassis architecture composed of one TX Matrix router and from one to four interconnected T640 routers. From the perspective of the user interface, the routing matrix appears as a single router. The TX Matrix router controls all the T640 routers, as shown in Figure 1 on page 36.

Figure 1: Routing Matrix



A TX Matrix router is also referred to as a *switch-card chassis* (SCC). The CLI uses **scc** to refer to the TX Matrix router. A T640 router in a routing matrix is also referred to as a *line-card chassis* (LCC). The CLI uses **lcc** as a prefix to refer to a specific T640 router.

LCCs are assigned numbers 0 through 3, depending on the hardware setup and connectivity to the TX Matrix router. For more information, see the *TX Matrix Router Hardware Guide*. A routing matrix can have up to four T640 routers, and each T640 router has up to eight FPCs. Therefore, the routing matrix as a whole can have up to 32 FPCs (0 through 31).

In the Junos OS CLI, an interface name has the following format:

```
type-fpc/pic/port
```

When you specify the ***fpc*** number for a T640 router in a routing matrix, the Junos OS determines which T640 router contains the specified FPC based on the following assignment:

- On LCC 0, FPC hardware slots 0 through 7 are configured as 0 through 7.
- On LCC 1, FPC hardware slots 0 through 7 are configured as 8 through 15.
- On LCC 2, FPC hardware slots 0 through 7 are configured as 16 through 23.
- On LCC 3, FPC hardware slots 0 through 7 are configured as 24 through 31.

For example, the **1** in **se-1/0/0** refers to FPC hardware slot 1 on the T640 router labeled **lcc0**. The **11** in **t1-11/2/0** refers to FPC hardware slot 3 on the T640 router labeled **lcc1**. The **20** in **so-20/0/1** refers to FPC hardware slot 4 on the T640 router labeled **lcc2**. The **31** in **t3-31/1/0** refers to FPC hardware slot 7 on the T640 router labeled **lcc3**.

Table 3 on page 37 summarizes the FPC numbering for a T640 router in a routing matrix.

Table 3: FPC Numbering for T640 Routers in a Routing Matrix

LCC Numbers Assigned to the T640 Router	Configuration Numbers
0	0 through 7
1	8 through 15
2	16 through 23
3	24 through 31

Table 4 on page 37 lists each FPC hardware slot and the corresponding configuration numbers for LCCs 0 through 3.

Table 4: One-to-One FPC Numbering for T640 Routers in a Routing Matrix

FPC Numbering	T640 Routers							
	LCC 0							
Hardware Slots	0	1	2	3	4	5	6	7
Configuration Numbers	0	1	2	3	4	5	6	7
	LCC 1							
Hardware Slots	0	1	2	3	4	5	6	7
Configuration Numbers	8	9	10	11	12	13	14	15
	LCC 2							
Hardware Slots	0	1	2	3	4	5	6	7
Configuration Numbers	16	17	18	19	20	21	22	23
	LCC 3							
Hardware Slots	0	1	2	3	4	5	6	7

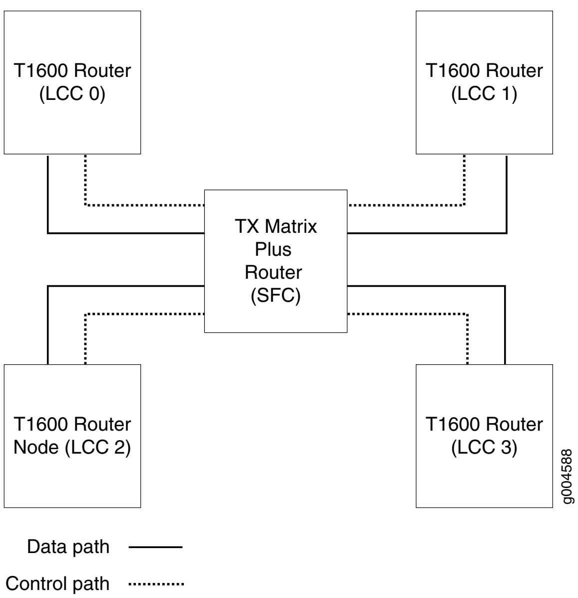
Table 4: One-to-One FPC Numbering for T640 Routers in a Routing Matrix *(continued)*

FPC Numbering	T640 Routers							
Configuration Numbers	24	25	26	27	28	29	30	31

Interface Naming for a Routing Matrix Based on a TX Matrix Plus Router

A routing matrix based on a Juniper Networks TX Matrix Plus Router is a multichassis architecture composed of one TX Matrix Plus router and from one to four interconnected T1600 routers. From the perspective of the user interface, the routing matrix appears as a single router. The TX Matrix Plus router controls all the T1600 routers, as shown in Figure 2 on page 38.

Figure 2: Routing Matrix Based on a TX Matrix Plus Router



A TX Matrix Plus router is also referred to as a *switch-fabric chassis* (SFC). The CLI uses **sfc** to refer to the TX Matrix Plus router. A T1600 router in a routing matrix is also referred to as a *line-card chassis* (LCC). The CLI uses **lcc** as a prefix to refer to a specific T1600 router.

LCCs are assigned numbers, 0 through 3, depending on the hardware setup and connectivity to the TX Matrix Plus router. For more information, see the *TX Matrix Plus Router Hardware Guide*. A routing matrix based on a TX Matrix Plus router can have up to four T1600 routers, and each T1600 router has up to eight FPCs. Therefore, the routing matrix as a whole can have up to 32 FPCs (0 through 31).

In the Junos OS CLI, an interface name has the following format:

```
type-fpc/pic/port
```

When you specify the ***fpc*** number for a T1600 router in a routing matrix, the Junos OS determines which T1600 router contains the specified FPC based on the following assignment:

- On LCC 0, FPC hardware slots 0 through 7 are configured as 0 through 7.
- On LCC 1, FPC hardware slots 0 through 7 are configured as 8 through 15.
- On LCC 2, FPC hardware slots 0 through 7 are configured as 16 through 23.
- On LCC 3, FPC hardware slots 0 through 7 are configured as 24 through 31.

For example, the **1** in **se-1/0/0** refers to FPC hardware slot 1 on the T1600 router labeled **lcc0**. The **11** in **t1-11/2/0** refers to FPC hardware slot 3 on the T1600 router labeled **lcc1**. The **20** in **so-20/0/1** refers to FPC hardware slot 4 on the T1600 router labeled **lcc2**. The **31** in **t3-31/1/0** refers to FPC hardware slot 7 on the T1600 router labeled **lcc3**.

Table 5 on page 39 summarizes the FPC numbering for a routing matrix based on a TX Matrix Plus router.

Table 5: FPC Numbering for T1600 Routers in a Routing Matrix

LCC Numbers Assigned to the T1600 Router	Configuration Numbers
0	0 through 7
1	8 through 15
2	16 through 23
3	24 through 31

Table 6 on page 39 lists each FPC hardware slot and the corresponding configuration numbers for LCCs 0 through 3.

Table 6: One-to-One FPC Numbering for T1600 Routers in a Routing Matrix

FPC Numbering	T1600 Routers							
	LCC 0							
Hardware Slots	0	1	2	3	4	5	6	7
Configuration Numbers	0	1	2	3	4	5	6	7

Table 6: One-to-One FPC Numbering for T1600 Routers in a Routing Matrix *(continued)*

FPC Numbering	T1600 Routers							
	LCC 1							
Hardware Slots	0	1	2	3	4	5	6	7
Configuration Numbers	8	9	10	11	12	13	14	15
	LCC 2							
Hardware Slots	0	1	2	3	4	5	6	7
Configuration Numbers	16	17	18	19	20	21	22	23
	LCC 3							
Hardware Slots	0	1	2	3	4	5	6	7
Configuration Numbers	24	25	26	27	28	29	30	31

Chassis Interface Naming

You configure some PIC properties, such as framing, at the **[edit chassis]** hierarchy level. Chassis interface naming varies depending on the routing hardware.

- To configure PIC properties for a standalone router, you must specify the FPC and PIC numbers, as follows:

```
[edit chassis]
fpc slot-number {
    pic pic-number {
        ...
    }
}
```

- To configure PIC properties for a T640 or T1600 router configured in a routing matrix, you must specify the LCC, FPC, and PIC numbers, as follows:

```
[edit chassis]
```

```
lcc lcc-number {
    fpc slot-number { # Use the hardware FPC slot number
        pic pic-number {
            ...
        }
    }
}
```

For the FPC slot in a T640 router in a routing matrix, specify the actual hardware slot number, as labeled on the T640 router chassis. Do not use the corresponding software FPC configuration numbers shown in Table 4 on page 37.

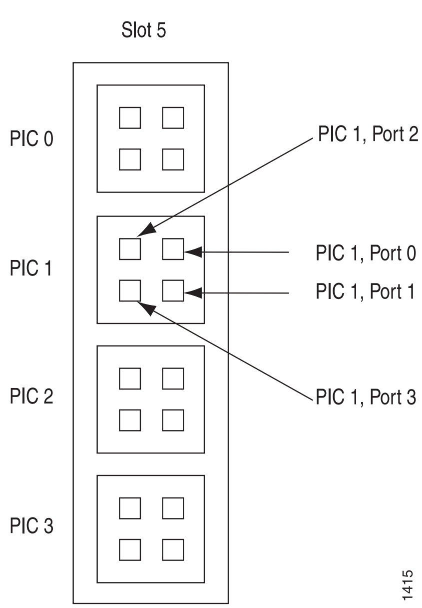
For the FPC slot in a T1600 router in a routing matrix, specify the actual hardware slot number, as labeled on the T1600 router chassis. Do not use the corresponding software FPC configuration numbers shown in Table 5 on page 39.

For more information about the **[edit chassis]** hierarchy, see the *Junos OS Administration Library*.

Examples: Interface Naming

This section provides examples of naming interfaces. For an illustration of where slots, PICs, and ports are located, see Figure 3 on page 41.

Figure 3: Interface Slot, PIC, and Port Locations



For an FPC in slot 1 with two OC3 SONET/SDH PICs in PIC positions 0 and 1, each PIC with two ports uses the following names:

```
so-1/0/0.0
so-1/0/1.0
so-1/1/0.0
so-1/1/1.0
```

An OC48 SONET/SDH PIC in slot 1 and in concatenated mode appears as a single FPC with a single PIC, which has a single port. If this interface has a single logical unit, it has the following name:

```
so-1/0/0.0
```

An OC48 SONET/SDH PIC in slot 1 and in channelized mode has a number for each channel. For example:

```
so-1/0/0:0
so-1/0/0:1
```

For an FPC in slot 1 with a Channelized OC12 PIC in PIC position 2, the DS3 channels have the following names:

```
t3-1/2/0:0
t3-1/2/0:1
t3-1/2/0:2
...
t3-1/2/0:11
```

For an FPC in slot 1 with four OC12 ATM PICs (the FPC is fully populated), the four PICs, each with a single port and a single logical unit, have the following names:

```
at-1/0/0.0
at-1/1/0.0
at-1/2/0.0
at-1/3/0.0
```

In a routing matrix on the T640 router labeled **lcc1**, for an FPC in slot 5 with four SONET OC192 PICs, the four PICs, each with a single port and a single logical unit, have the following names:

```
so-13/0/0.0
so-13/1/0.0
```

```
so-13/2/0.0
so-13/3/0.0
```

For an FPC in slot 1 with one 4-port ISDN BRI interface card, port 4 has the following name:

```
br-1/0/4
```

The first B-channel, the second B-channel, and the control channel have the following names:

```
bc-1/0/4:1
bc-1/0/4:2
dc-1/0/4:0
```

SEE ALSO

Supported Routing Engines by Chassis

Interface Descriptors Overview

When you configure an interface, you are effectively specifying the properties for a physical interface descriptor. In most cases, the physical interface descriptor corresponds to a single physical device and consists of the following parts:

- The interface name, which defines the media type
- The slot in which the FPC or DPC is located
- The location on the FPC in which the PIC is installed
- The PIC or DPC port
- The interface's channel and logical unit numbers (optional)

Each physical interface descriptor can contain one or more logical interface descriptors. These allow you to map one or more logical (or virtual) interfaces to a single physical device. Creating multiple logical interfaces is useful for ATM, Frame Relay, and Gigabit Ethernet networks, in which you can associate multiple virtual circuits, data-link connections, or virtual LANs (VLANs) with a single interface device.

Each logical interface descriptor can have one or more family descriptors to define the protocol family that is associated with and allowed to run over the logical interface.

The following protocol families are supported:

- Internet Protocol version 4 (IPv4) suite (inet)
- Internet Protocol version 6 (IPv6) suite (inet6)
- Circuit cross-connect (CCC)
- Translational cross-connect (TCC)
- International Organization for Standardization (ISO)
- Multilink Frame Relay end-to-end (MLFR end-to-end)
- Multilink Frame Relay user-to-network interface network-to-network interface (MLFR UNI NNI)
- Multilink Point-to-Point Protocol (MLPPP)
- Multiprotocol Label Switching (MPLS)
- Trivial Network Protocol (TNP)
- (M Series, T Series, and MX Series routers only) Virtual private LAN service (VPLS)

Finally, each family descriptor can have one or more address entries, which associate a network address with a logical interface and hence with the physical interface.

You configure the various interface descriptors as follows:

- You configure the physical interface descriptor by including the **interfaces *interface-name*** statement.
- You configure the logical interface descriptor by including the **unit** statement within the **interfaces *interface-name*** statement or by including the ***.logical*** descriptor at the end of the interface name, as in **t3-0/0/0.1**, where the logical unit number is 1, as shown in the following examples:

```
[edit]
user@host# set interfaces t3-0/0/0 unit 1
[edit]
user@host# edit interfaces t3-0/0/0.1
[edit interfaces t3-0/0/0]
user@host# set unit 1
```

- You configure the family descriptor by including the *family* statement within the **unit** statement.
- You configure address entries by including the *address* statement within the *family* statement.
- You configure tunnels by including the *tunnel* statement within the **unit** statement.

NOTE: The address of a logical interface cannot be the same as a tunnel interface’s source or destination address. If you try to configure a logical interface with a tunnel interface’s address or vice versa, a commit failure will occur.

Physical Part of an Interface Name

IN THIS SECTION

Interface Names for ACX Series Universal Metro Routers

ACX Series routers do not have actual PIC devices. Instead they have built-in network ports on the front panel of the router. These ports are named using the same naming convention used for routers with PIC devices with the understanding that the FPC, PIC and port are pseudo devices. When you display information about one of these ports, you specify the interface type, the slot for the Flexible PIC Concentrator (FPC), the slot on the FPC for the Physical Interface Card (PIC), and the configured port number.

In the physical part of the interface name, a hyphen (-) separates the media type from the FPC number, and a slash (/) separates the FPC, PIC, and port numbers:

```
type-fpc/pic/port
```

SEE ALSO

Understanding Encapsulation on an Interface

Configuring Inverse Multiplexing for ATM (IMA) on ACX Series

Interface Names for M Series and T Series Routers

On M Series and T Series routers, when you display information about an interface, you specify the interface type, the slot in which the Flexible PIC Concentrator (FPC) is installed, the slot on the FPC in which the Physical Interface Card (PIC) is located, and the configured port number.

In the physical part of the interface name, a hyphen (-) separates the media type from the FPC number, and a slash (/) separates the FPC, PIC, and port numbers:

```
type-fpc/pic/port
```

> **NOTE:** Exceptions to the ***type-fpc/pic/port*** physical description include the aggregated Ethernet and aggregated SONET/SDH interfaces, which use the syntax **ae *number*** and **as *number***, respectively.

MX Series Router Interface Names

On MX Series routers when you display information about an interface, you specify the interface type, the Dense Port Concentrator (DPC), Flexible PIC Concentrator (FPC), or Modular Port Concentrator (MPC) slot, the PIC or MIC slot, and the configured port number.

> **NOTE:** Although the MX Series routers use DPCs, FPCs, MPCs, MICs, and PICs, command syntax in this book is shown as *fpc/pic/port* for simplicity.

In the physical part of the interface name, a hyphen (-) separates the media type from the FPC number, and a slash (/) separates the DPC, FPC or MPC, MIC or PIC, and port numbers:

```
type-fpc/pic/port
```

- *fpc*—Slot in which the DPC, FPC, or MPC is installed.
- *pic*—Slot on the FPC in which the PIC is located.

  For DPCs, MICs, and the 16-port MPC, the PIC value is a logical grouping of ports and varies on different platforms.
- *port*—Port number on the DPC, PIC, MPC, or MIC.

Interface Names for PTX Series Routers

On PTX Series Packet Transport Routers, when you display information about an interface, you specify the interface type, the slot in which the Flexible PIC Concentrator (FPC) is installed, the slot on the FPC in which the Physical Interface Card (PIC) is located, and the configured port number.

NOTE:

- The PTX router supports Ethernet type interfaces only. The media type portion of the physical interface name,*type* supports the Ethernet interface type only: **et**.
- In the CLI, all PTX3000 PICs are represented as **pic0**. For more information, see *PTX3000 PIC Description*

In the physical part of the interface name, a hyphen (**-**) separates the media type (**et**) from the FPC number, and a slash (**/**) separates the FPC, PIC, and port numbers:

```
type-fpc/pic/port
```

Displaying Interface Configurations

To display a configuration, use either the **show** command in configuration mode or the **show configuration** top-level command. Interfaces are listed in numerical order, from lowest to highest slot number, then from lowest to highest PIC number, and finally from lowest to highest port number.

Interface Encapsulations Overview

Table 7 on page 48 lists encapsulation support by interface type.

Table 7: Encapsulation Support by Interface Type

Interface Type	Physical Interface Encapsulation	Logical Interface Encapsulation
ae—Aggregated Ethernet interface	**ethernet-ccc**—Ethernet cross-connect **extended-vlan-ccc**—Nonstandard TPID tagging for a cross-connect **extended-vlan-vpls**—Extended VLAN virtual private LAN service **flexible-ethernet-services**—Allows per-unit Ethernet encapsulation configuration **vlan-ccc**—802.1Q tagging for a cross-connect **ethernet-vpls**—Ethernet virtual private LAN service **vlan-vpls**—VLAN virtual private LAN service	**dix**—Ethernet DIXv2 (RFC 894) **vlan-ccc**—802.1Q tagging for a cross-connect
as—Aggregated SONET/SDH interface	**cisco-hdlc**—Cisco-compatible HDLC framing **ppp**—Serial PPP device	NA
at—ATM1 interface	**atm-ccc-cell-relay**—ATM cell relay encapsulation for a cross-connect **atm-pvc**—ATM permanent virtual circuits **ethernet-over-atm**—Ethernet over ATM encapsulation	**atm-ccc-cell-relay**—ATM cell relay for CCC **atm-ccc-vc-mux**—ATM VC for CCC **atm-cisco-nlpid**—Cisco-compatible ATM NLPID encapsulation **atm-nlpid**—ATM NLPID encapsulation **atm-snap**—ATM LLC/SNAP encapsulation **atm-tcc-snap**—ATM LLC/SNAP for a translational cross-connect **atm-tcc-vc-mux**—ATM VC for a translational cross-connect **atm-vc-mux**—ATM VC multiplexing **ether-over-atm-llc**—Ethernet over ATM (LLC/SNAP) encapsulation

Table 7: Encapsulation Support by Interface Type *(continued)*

Interface Type	Physical Interface Encapsulation	Logical Interface Encapsulation
at—ATM2 intelligent queuing (IQ) interface	**atm-ccc-cell-relay**—ATM cell relay encapsulation for a cross-connect **atm-pvc**—ATM permanent virtual circuits **ethernet-over-atm**—Ethernet over ATM encapsulation	**atm-ccc-cell-relay**—ATM cell relay for CCC **atm-ccc-vc-mux**—ATM VC for CCC **atm-cisco-nlpid**—Cisco-compatible ATM NLPID encapsulation **atm-mlppp-llc**—ATM MLPPP over AAL5/LLC **atm-nlpid**—ATM NLPID encapsulation **atm-ppp-llc**—ATM PPP over AAL5/LLC **atm-ppp-vc-mux**—ATM PPP over raw AAL5 **atm-snap**—ATM LLC/SNAP encapsulation **atm-tcc-snap**—ATM LLC/SNAP for a translational cross-connect **atm-tcc-vc-mux**—ATM VC for a translational cross-connect **atm-vc-mux**—ATM VC multiplexing **ether-over-atm-llc**—Ethernet over ATM (LLC/SNAP) encapsulation **ether-vpls-over-atm-llc**—Ethernet VPLS over ATM (bridging) encapsulation
bcm—Gigabit Ethernet internal interfaces	NA	NA
br—Integrated Services Digital Network (ISDN) interface	NA	NA
ci—Container interface	**cisco-hdlc**—Cisco-compatible HDLC framing **ppp**—Serial PPP device	**aps**—SONET interface required for APS configuration.

Table 7: Encapsulation Support by Interface Type *(continued)*

Interface Type	Physical Interface Encapsulation	Logical Interface Encapsulation
ds—DS0 interface	**cisco-hdlc**—Cisco-compatible HDLC framing **cisco-hdlc-ccc**—Cisco-compatible HDLC framing for a cross-connect **cisco-hdlc-tcc**—Cisco-compatible HDLC framing for a translational cross-connect **extended-frame-relay-ccc**—Any Frame Relay DLCI for a cross-connect **extended-frame-relay-tcc**—Any Frame Relay DLCI for a translational cross-connect **flexible-frame-relay**—Multiple Frame Relay encapsulations **frame-relay**—Frame Relay encapsulation **frame-relay-ccc**—Frame Relay for a cross-connect **frame-relay-port-ccc**—Frame Relay port encapsulation for a cross-connect **frame-relay-tcc**—Frame Relay for a translational cross-connect **multilink-frame-relay-uni-nni**—Multilink Frame Relay UNI NNI (FRF.16) encapsulation **ppp**—Serial PPP device **ppp-ccc**—Serial PPP device for a cross-connect **ppp-tcc**—Serial PPP device for a translational cross-connect	**frame-relay-ccc**—Frame Relay DLCI for CCC **frame-relay-ppp**—PPP over Frame Relay **frame-relay-tcc**—Frame Relay DLCI for a translational cross-connect
dsc—Discard interface	NA	NA

Table 7: Encapsulation Support by Interface Type *(continued)*

Interface Type	Physical Interface Encapsulation	Logical Interface Encapsulation
e1—E1 interface (including channelized STM1-to-E1 interfaces)	**cisco-hdlc**—Cisco-compatible HDLC framing **cisco-hdlc-ccc**—Cisco-compatible HDLC framing for a cross-connect **cisco-hdlc-tcc**—Cisco-compatible HDLC framing for a translational cross-connect **extended-frame-relay-ccc**—Any Frame Relay DLCI for a cross-connect **extended-frame-relay-tcc**—Any Frame Relay DLCI for a translational cross-connect **flexible-frame-relay**—Multiple Frame Relay encapsulations **frame-relay**—Frame Relay encapsulation **frame-relay-ccc**—Frame Relay for a cross-connect **frame-relay-port-ccc**—Frame Relay port encapsulation for a cross-connect **frame-relay-tcc**—Frame Relay for a translational cross-connect **multilink-frame-relay-uni-nni**—Multilink Frame Relay UNI NNI (FRF.16) encapsulation **ppp**—Serial PPP device **ppp-ccc**—Serial PPP device for a cross-connect **ppp-tcc**—Serial PPP device for a translational cross-connect	**frame-relay-ccc**—Frame Relay DLCI for CCC **frame-relay-ppp**—PPP over Frame Relay **frame-relay-tcc**—Frame Relay DLCI for a translational cross-connect

Table 7: Encapsulation Support by Interface Type *(continued)*

Interface Type	Physical Interface Encapsulation	Logical Interface Encapsulation
e3—E3 interface (including E3 IQ and IQE interfaces)	**cisco-hdlc**—Cisco-compatible HDLC framing **cisco-hdlc-ccc**—Cisco-compatible HDLC framing for a cross-connect **cisco-hdlc-tcc**—Cisco-compatible HDLC framing for a translational cross-connect **extended-frame-relay-ccc**—Any Frame Relay DLCI for a cross-connect **extended-frame-relay-tcc**—Any Frame Relay DLCI for a translational cross-connect **flexible-frame-relay**—Multiple Frame Relay encapsulations **frame-relay**—Frame Relay encapsulation **frame-relay-ccc**—Frame Relay for a cross-connect **frame-relay-port-ccc**—Frame Relay port encapsulation for a cross-connect **frame-relay-tcc**—Frame Relay for a translational cross-connect **ppp**—Serial PPP device **ppp-ccc**—Serial PPP device for a cross-connect **ppp-tcc**—Serial PPP device for a translational cross-connect	**frame-relay-ccc**—Frame Relay DLCI for CCC **frame-relay-ppp**—PPP over Frame Relay **frame-relay-tcc**—Frame Relay DLCI for a translational cross-connect
em—Management and internal Ethernet interfaces	NA	NA

Table 7: Encapsulation Support by Interface Type *(continued)*

Interface Type	Physical Interface Encapsulation	Logical Interface Encapsulation
fe—Fast Ethernet interface	**ethernet-ccc**—Ethernet cross-connect **ethernet-tcc**—Ethernet translational cross-connect **ethernet-vpls**—Ethernet virtual private LAN service **extended-vlan-ccc**—Nonstandard TPID tagging for a cross-connect **extended-vlan-tcc**—802.1Q tagging for a translational cross-connect **extended-vlan-vpls**—Extended VLAN virtual private LAN service **vlan-ccc**—802.1Q tagging for a cross-connect **vlan-vpls**—VLAN virtual private LAN service	**dix**—Ethernet DIXv2 (RFC 894) **vlan-ccc**—802.1Q tagging for a cross-connect **vlan-vpls**—VLAN virtual private LAN service
fxp—Management and internal Ethernet interfaces	NA	NA

Table 7: Encapsulation Support by Interface Type *(continued)*

Interface Type	Physical Interface Encapsulation	Logical Interface Encapsulation
ge—Gigabit Ethernet interface (including Gigabit Ethernet IQ interfaces)	**ethernet-ccc**—Ethernet cross-connect **ethernet-tcc**—Ethernet translational cross-connect **ethernet-vpls**—Ethernet virtual private LAN service **extended-vlan-ccc**—Nonstandard TPID tagging for a cross-connect **extended-vlan-tcc**—802.1Q tagging for a translational cross-connect **extended-vlan-vpls**—Extended VLAN virtual private LAN service **flexible-ethernet-services**—Allows per-unit Ethernet encapsulation configuration **vlan-ccc**—802.1Q tagging for a cross-connect **vlan-vpls**—VLAN virtual private LAN service	**dix**—Ethernet DIXv2 (RFC 894) **vlan-ccc**—802.1Q tagging for a cross-connect **vlan-tcc**—802.1Q tagging for a translational cross-connect **vlan-vpls**—VLAN virtual private LAN service
ixgbe—10-Gigabit Ethernet internal interfaces	NA	NA
lo—Loopback interface; the Junos OS automatically configures one loopback interface (**lo0**)	NA	NA
ls—Link services interface	**multilink-frame-relay-uni-nni**—Multilink Frame Relay UNI NNI (FRF.16) encapsulation	**multilink-frame-relay-end-to-end**—Multilink Frame Relay end-to-end (FRF.15) **multilink-ppp**—Multilink PPP
lsq—Link services IQ interface	**multilink-frame-relay-uni-nni**—Multilink Frame Relay UNI NNI (FRF.16) encapsulation	**multilink-frame-relay-end-to-end**—Multilink Frame Relay end-to-end (FRF.15) **multilink-ppp**—Multilink PPP

Table 7: Encapsulation Support by Interface Type *(continued)*

Interface Type	Physical Interface Encapsulation	Logical Interface Encapsulation
lt—Logical tunnel interface	NA	**ethernet**—Ethernet service **ethernet-vpls**—Ethernet virtual private LAN service **ethernet-ccc**—Ethernet cross-connect **frame-relay**—Frame Relay encapsulation **frame-relay-ccc**—Frame Relay for a cross-connect **vlan**—VLAN service **vlan-ccc**—802.1Q tagging for a cross-connect **vlan-vpls**—VLAN virtual private LAN service
ml—Multilink interface (including Multilink Frame Relay and MLPPP)	NA	**multilink-frame-relay-end-to-end**—Multilink Frame Relay end-to-end (FRF.15) **multilink-ppp**—Multilink PPP

Table 7: Encapsulation Support by Interface Type *(continued)*

Interface Type	Physical Interface Encapsulation	Logical Interface Encapsulation
se—Serial interface (including EIA-530, V.35, and X.21 interfaces)	**cisco-hdlc**—Cisco-compatible HDLC framing **cisco-hdlc-ccc**—Cisco-compatible HDLC framing for a cross-connect **cisco-hdlc-tcc**—Cisco-compatible HDLC framing for a translational cross-connect **frame-relay**—Frame Relay encapsulation **frame-relay-ccc**—Frame Relay for a cross-connect **frame-relay-port-ccc**—Frame Relay port encapsulation for a cross-connect **frame-relay-tcc**—Frame Relay for a translational cross-connect **ppp**—Serial PPP device **ppp-ccc**—Serial PPP device for a cross-connect **ppp-tcc**—Serial PPP device for a translational cross-connect	**frame-relay-ccc**—Frame Relay DLCI for CCC **frame-relay-ppp**—PPP over Frame Relay **frame-relay-tcc**—Frame Relay DLCI for a translational cross-connect

Table 7: Encapsulation Support by Interface Type *(continued)*

Interface Type	Physical Interface Encapsulation	Logical Interface Encapsulation
so—SONET/SDH interface	**cisco-hdlc**—Cisco-compatible HDLC framing **cisco-hdlc-ccc**—Cisco-compatible HDLC framing for a cross-connect **cisco-hdlc-tcc**—Cisco-compatible HDLC framing for a translational cross-connect **extended-frame-relay-ccc**—Any Frame Relay DLCI for a cross-connect **extended-frame-relay-tcc**—Any Frame Relay DLCI for a translational cross-connect **flexible-frame-relay**—Multiple Frame Relay encapsulations **frame-relay**—Frame Relay encapsulation **frame-relay-ccc**—Frame Relay for a cross-connect **frame-relay-port-ccc**—Frame Relay port encapsulation for a cross-connect **frame-relay-tcc**—Frame Relay for a translational cross-connect **ppp**—Serial PPP device **ppp-ccc**—Serial PPP device for a cross-connect **ppp-tcc**—Serial PPP device for a translational cross-connect	**frame-relay-ccc**—Frame Relay DLCI for CCC **frame-relay-ppp**—PPP over Frame Relay **frame-relay-tcc**—Frame Relay DLCI for a translational cross-connect **multilink-frame-relay-end-to-end**—IQE SONET PICs support Multilink Frame Relay end-to-end (FRF.15) **multilink-ppp**—IQE SONET PICs support Multilink PPP

Table 7: Encapsulation Support by Interface Type *(continued)*

Interface Type	Physical Interface Encapsulation	Logical Interface Encapsulation
t1—T1 interface (including channelized DS3-to-DS1 interfaces)	**cisco-hdlc**—Cisco-compatible HDLC framing **cisco-hdlc-ccc**—Cisco-compatible HDLC framing for a cross-connect **cisco-hdlc-tcc**—Cisco-compatible HDLC framing for a translational cross-connect **extended-frame-relay-ccc**—Any Frame Relay DLCI for a cross-connect **extended-frame-relay-tcc**—Any Frame Relay DLCI for a translational cross-connect **flexible-frame-relay**—Multiple Frame Relay encapsulations **frame-relay**—Frame Relay encapsulation **frame-relay-ccc**—Frame Relay for a cross-connect **frame-relay-port-ccc**—Frame Relay port encapsulation for a cross-connect **frame-relay-tcc**—Frame Relay for a translational cross-connect **multilink-frame-relay-uni-nni**—Multilink Frame Relay UNI NNI (FRF.16) encapsulation **ppp**—Serial PPP device **ppp-ccc**—Serial PPP device for a cross-connect **ppp-tcc**—Serial PPP device for a translational cross-connect	**frame-relay-ccc**—Frame Relay DLCI for CCC **frame-relay-ppp**—PPP over Frame Relay **frame-relay-tcc**—Frame Relay DLCI for a translational cross-connect

Table 7: Encapsulation Support by Interface Type *(continued)*

Interface Type	Physical Interface Encapsulation	Logical Interface Encapsulation
t3—T3 interface (including channelized OC12-to-DS3 interfaces)	**cisco-hdlc**—Cisco-compatible HDLC framing **cisco-hdlc-ccc**—Cisco-compatible HDLC framing for a cross-connect **cisco-hdlc-tcc**—Cisco-compatible HDLC framing for a translational cross-connect **extended-frame-relay-ccc**—Any Frame Relay DLCI for a cross-connect **extended-frame-relay-tcc**—Any Frame Relay DLCI for a translational cross-connect **flexible-frame-relay**—Multiple Frame Relay encapsulations **frame-relay**—Frame Relay encapsulation **frame-relay-ccc**—Frame Relay for a cross-connect **frame-relay-port-ccc**—Frame Relay port encapsulation for a cross-connect **frame-relay-tcc**—Frame Relay for a translational cross-connect **ppp**—Serial PPP device **ppp-ccc**—Serial PPP device for a cross-connect **ppp-tcc**—Serial PPP device for a translational cross-connect	**frame-relay-ccc**—Frame Relay DLCI for CCC **frame-relay-ppp**—PPP over Frame Relay **frame-relay-tcc**—Frame Relay DLCI for a translational cross-connect
Controller-level channelized IQ interfaces (**cau4**, **coc1**, **coc3**, **coc12**, **cstm1**, **ct1**, **ct3**, **ce1)**	NA	NA
Services interfaces (**cp**, **gr**, **ip**, **mo**, **vt**, **es**, **mo**, **rsp**, **sp**)	NA	NA

Table 7: Encapsulation Support by Interface Type *(continued)*

Interface Type	Physical Interface Encapsulation	Logical Interface Encapsulation
Unconfigurable, internally generated interfaces (**gre**, **ipip**, **learning-chip (lc)**, **lsi**, **tap**, **mt**, **mtun**, **pd**, **pe**, **pimd**, **pime**)	NA	NA

NOTE: You can configure GRE interfaces (gre-x/y/z) only for GMPLS control channels. GRE interfaces are not supported or configurable for other applications. For more information about GMPLS, see the *MPLS Applications User Guide*.

Understanding Transient Interfaces

The M Series, MX Series, and T Series routers contain slots for installing Flexible PIC Concentrator [FPC] or Dense Port Concentrator [DPC] (for MX Series routers) or Modular Port Concentrator [MPC] (for MX Series routers). Physical Interface Card [PIC] can be installed in FPCs. Modular Interface Card [MIC] can be inserted into MPCs.

The number of PICs that can be installed varies by router and type of FPC. The PICs provide the actual physical interfaces to the network. The MX Series routers contain slots for installing either DPC boards that provide the physical interfaces to the network or for installing FPCs in which PICs can be installed.

You can insert any DPC or FPC into any slot that supports them in the appropriate router. Typically, you can place any combination of PICs, compatible with your router, in any location on an FPC. (You are limited by the total FPC bandwidth, and by the fact that some PICs physically require two or four of the PIC locations on the FPC. In some cases, power limitations or microcode limitations may also apply.) To determine DPC and PIC compatibility, see the see your router's *Interface Module Reference*.

You can insert MPC into any slot that supports them in the appropriate router. You can install up to two MICs of different media types in the same MPC as long as the MPC supports those MICs.

These physical interfaces are transient interfaces of the router. They are referred to as transient because you can hot-swap a DPC or FPC or MPC and its PICs or MICs at any time.

You must configure each transient interface based on the slot in which the FPC or DPC or MPC is installed, the location in which the PIC or MIC is installed, and for multiple port PICs or MICs , the port to which you are connecting.

You can configure the interfaces on PICs or MICs that are already installed in the router as well as interfaces on PICs or MICs that you plan to install later. The Junos OS detects which interfaces are actually present, so when the software activates its configuration, it activates only the present interfaces and retains the configuration information for the interfaces that are not present. When the Junos OS detects that an FPC containing PICs or MPC containing MICs has been inserted into the router, the software activates the configuration for those interfaces.

Understanding Services Interfaces

Services interfaces enable you to incrementally add services to your network. The Junos OS supports the following services PICs:

- Adaptive Services (AS) PICs—Allow you to provide multiple services on a single PIC by configuring a set of services and applications. The AS PICs offer a special range of services you configure in one or more service sets.
- ES PIC—Provides a security suite for the IP version 4 (IPv4) and IP version 6 (IPv6) network layers. The suite provides functionality such as authentication of origin, data integrity, confidentiality, replay protection, and nonrepudiation of source. It also defines mechanisms for key generation and exchange, management of security associations, and support for digital certificates.
- Monitoring Services PICs—Enable you to monitor traffic flow and export the monitored traffic. Monitoring traffic allows you to gather and export detailed information about IPv4 traffic flows between source and destination nodes in your network; sample all incoming IPv4 traffic on the monitoring interface and present the data in cflowd record format; perform discard accounting on an incoming traffic flow; encrypt or tunnel outgoing cflowd records, intercepted IPv4 traffic, or both; and direct filtered traffic to different packet analyzers and present the data in its original format. On a Monitoring Services II PIC, you can configure either monitoring interfaces or collector interfaces. A collector interface allows you to combine multiple cflowd records into a compressed ASCII data file and export the file to an FTP server.
- Multilink Services, MultiServices, Link Services, and Voice Services PICs—Enable you to split, recombine, and sequence datagrams across multiple logical data links. The goal of multilink operation is to coordinate multiple independent links between a fixed pair of systems, providing a virtual link with greater bandwidth than any of the members.
- Tunnel Services PIC—By encapsulating arbitrary packets inside a transport protocol, tunneling provides a private, secure path through an otherwise public network. Tunnels connect discontinuous subnetworks

and enable encryption interfaces, virtual private networks (VPNs), and Multiprotocol Label Switching (MPLS).

- On M Series and T Series routers, logical tunnel interfaces allow you to connect logical systems, virtual routers, or VPN instances. For more information about VPNs, see the *Junos OS VPNs Library for Routing Devices*. For more information about configuring tunnels, see the *Junos OS Services Interfaces Library for Routing Devices*.

Understanding Container Interfaces

IN THIS SECTION

Container interfaces provide the following features:

- Automatic protection switching (APS) on SONET/SDH and ATM links are supported using the container infrastructure.
- Container physical interfaces and logical interfaces remain up on switchover.
- APS parameters are auto-copied from the container interface to the member links.

NOTE: Paired groups and true unidirectional APS are not currently supported.

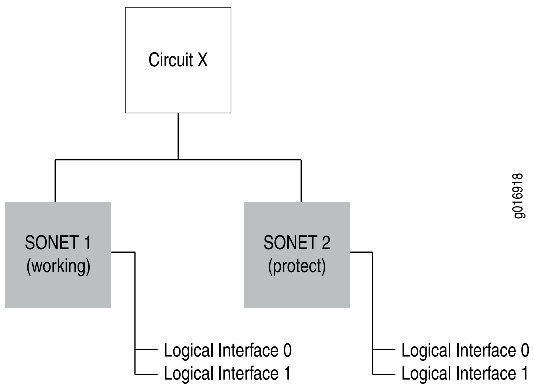
For more information on SONET/SDH configuration, see *Configuring Container Interfaces for APS on SONET Links*.

Container interfaces features are described in the following sections:

Understanding Traditional APS Concept

Traditional APS is configured on two independent physical SONET/SDH interfaces: one configured as the working circuit and the other as the protect circuit (see Figure 4 on page 63). The circuit, named Circuit X in the figure, is the link between the two SONET interfaces.

Figure 4: APS Interface

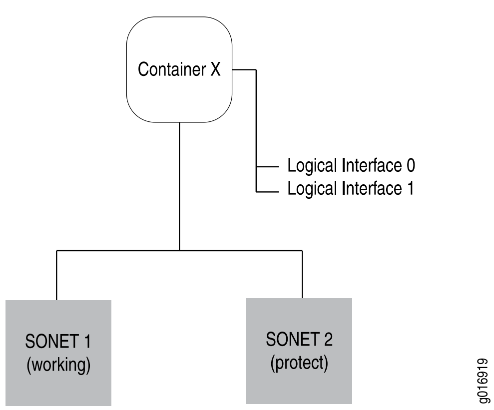


Traditional APS uses routing protocols that run on each individual SONET/SDH interface (since circuit is an abstract construct, instead of being an actual interface). When the working link goes down, the APS infrastructure brings up the protect link and its underlying logical interfaces, and brings down the working link and its underlying logical interfaces, causing the routing protocols to reconverge. This consumes time and leads to traffic loss even though the APS infrastructure has performed the switch quickly.

Container Interfaces Concept

To solve this problem, the Junos OS provides a soft interface construct called a container interface (see Figure 5 on page 63).

Figure 5: Container Interface



The container interface allows routing protocols to run on the logical interfaces associated with a virtual *container interface* instead of on the physical SONET/SDH and ATM interfaces. When APS switches the underlying physical link based on a fault condition, the container interface remains up, and the logical interface on the container interface does not flap. The routing protocols remain unaware of the APS switching.

APS Support for Container-Based Interfaces

With the container interface, APS is configured on the container interface itself. Individual member SONET/SDH and ATM links are either marked as primary (corresponding to the working circuit) or standby (corresponding to the protect circuit) in the configuration. No circuit or group name is specified in the container interface model; physical SONET/SDH and ATM links are put in an APS group by linking them to a single container interface. APS parameters are specified at the container interface level, and are propagated to the individual SONET/SDH and ATM links by the APS daemon.

Autocopy of APS Parameters

Typical applications require copying APS parameters from the working circuit to the protect circuit, since most of the parameters must be the same for both circuits. This is automatically done in the container interface. APS parameters are specified only once under the container physical interface configuration, and are internally copied over to the individual physical SONET/SDH and ATM links.

SEE ALSO

Configuring Container Interfaces for APS on SONET Links

Displaying APS Using a Container Interface with ATM Encapsulation

Understanding Internal Ethernet Interfaces

Within a router or packet transport router, internal Ethernet interfaces provide communication between the Routing Engine and the Packet Forwarding Engines. The Junos OS automatically configures internal Ethernet interfaces when the Junos OS boots. The Junos OS boots the packet-forwarding component hardware. When these components are running, the Control Board uses the internal Ethernet interface to transmit hardware status information to the Routing Engine. Information transmitted includes the internal router temperature, the condition of the fans, whether an FPC has been removed or inserted, and information from the LCD on thecraft interface.

To determine the supported internal Ethernet interfaces for your router, see *Supported Routing Engines by Router*.

> **NOTE:** Do not modify or remove the configuration for the internal Ethernet interface that the Junos OS automatically configures. If you do, the router or packet transport routerwill stop functioning.

- M Series, and MX Series routers and T Series routers—The Junos OS creates the internal Ethernet interface. The internal Ethernet interface connects the Routing Engine **re0** to the Packet Forwarding Engines.

  If the router has redundant Routing Engines, another internal Ethernet interface is created on each Routing Engine (**re0** and **re1**) in order to support fault tolerance, two physical links between **re0** and **re1** connect the independent control planes. If one of the links fails, both Routing Engines can use the other link for IP communication.

- TX Matrix Plus routers—On a TX Matrix Plus router, the Routing Engine and Control Board function as a unit, or host subsystem. For each host subsystem in the router, the Junos OS automatically creates two internal Ethernet interfaces, **ixgbe0** and **ixgbe1**.

  The ixgbe0 and ixgbe1 interfaces connect the TX Matrix Plus Routing Engine to the Routing Engines of every line-card chassis (LCC) configured in the routing matrix.

  The TX Matrix Plus Routing Engine connects to a high-speed switch through a 10-Gbps link within the host subsystem. The switch provides a 1-Gbps link to each T1600 Routing Engine. The 1-Gbps links are provided through the UTP Category 5 Ethernet cable connections between the TXP-CBs and the LCC-CBs in the LCCs.

  - The TX Matrix Plus Routing Engine connects to a high-speed switch in the local Control Board through a 10-Gbps link within the host subsystem.

  - The Gigabit Ethernet switch connects the Control Board to the remote Routing Engines of every LCC configured in the routing matrix.

If a TX Matrix Plus router contains redundant host subsystems, the independent control planes are connected by two physical links between the two 10-Gigabit Ethernet ports on their respective Routing Engines.

- The primary link to the remote Routing Engine is at the **ixgbe0** interface; the 10-Gigabit Ethernet switch on the local Control Board also connects the Routing Engine to the 10-Gigabit Ethernet port accessed by the **ixgbe1** interface on the remote Routing Engine.
- The alternate link to the remote Routing Engine is the 10-Gigabit Ethernet port at the **ixgbe1** interface. This second port connects the Routing Engine to the 10-Gigabit Ethernet switch on the remote Control Board, which connects to the 10-Gigabit Ethernet port at the **ixgbe0** interface on the remote Routing Engine.

If one of the two links between the host subsystems fails, both Routing Engines can use the other link for IP communication.

- LCC in a routing matrix—On an LCC configured in a routing matrix, the Routing Engine and Control Board function as a unit, or host subsystem. For each host subsystem in the LCC, the Junos OS automatically creates two internal Ethernet interfaces, **bcm0** and **em1**, for the two Gigabit Ethernet ports on the Routing Engine.

  The **bcm0** interface connects the Routing Engine in each LCCto the Routing Engines of every other LCC configured in the routing matrix.

  - The Routing Engine connects to a Gigabit Ethernet switch on the local Control Board through a.
  - The switch connects the Control Board to the remote Routing Engines of every other LCC configured in the routing matrix.

  If an LCC in a routing matrix contains redundant host subsystems, the independent control planes are connected by two physical links between the Gigabit Ethernet ports on their respective Routing Engines.

  - The primary link to the remote Routing Engine is at the **bcm0** interface; the Gigabit Ethernet switch on the local Control Board also connects the Routing Engine to the Gigabit Ethernet port accessed by the **em1** interface on the remote Routing Engine.
  - The alternate link to the remote Routing Engine is at the **em1** interface. This second port connects the Routing Engine to the Gigabit Ethernet switch on the remote Control Board, which connects to the Gigabit Ethernet port at the **bcm0** interface on the remote Routing Engine.

  If one of the two links between the host subsystems fails, both Routing Engines can use the other link for IP communication.

Each router also has two serial ports, labeled *console* and *auxiliary*, for connecting tty type terminals to the router using standard PC-type tty cables. Although these ports are not network interfaces, they do provide access to the router.

SEE ALSO

Supported Routing Engines by Router

Displaying Internal Ethernet Interfaces for a Routing Matrix with a TX Matrix Plus Router

show interfaces (M Series, MX Series, T Series Routers, and PTX Series Management and Internal Ethernet)

Understanding Interfaces on ACX Series Universal Metro Routers

The ACX Series routers support time-division multiplexing (TDM) T1 and E1 interfaces and Ethernet (1 GbE copper, 1GbE, 10 GbE, and 40 GbE fiber) interfaces to support both the legacy and evolution needs of the mobile network. Support for Power over Ethernet (PoE+) at 65 watts per port mitigates the need for additional electrical cabling for microwaves or other access interfaces.

The ACX Series routers support the following:

- TDM T1 and E1 ports:
  - The ACX1000 router contains eight T1 or E1 ports.
  - The ACX2000 router contains 16 T1 or E1 ports.
  - Inverse Multiplexing for ATM (IMA)

  > **NOTE:** ACX5048 and ACX5096 routers do not support T1 or E1 ports and Inverse Multiplexing for ATM (IMA).

- Gigabit Ethernet ports:
  - The ACX1000 router contains eight Gigabit Ethernet ports. The ACX1000 router also supports either four RJ45 (Cu) ports or installation of four Gigabit Ethernet small form-factor pluggable (SFP) transceivers.
  - The ACX2000 router contains 16 Gigabit Ethernet ports and two PoE ports. The ACX2000 router also supports installation of two Gigabit Ethernet SFP transceivers and two 10-Gigabit Ethernet SFP+ transceivers.
  - The ACX5448 router is a 10-Gigabit Ethernet enhanced small form-factor pluggable (SFP+) top-of-rack router with 48 SFP+ ports, and four 100-Gigabit Ethernet QSFP28 ports. Each SFP+ port can operate as a native 10-Gigabit Ethernet port, or as a 1-Gigabit Ethernet port when 1-Gigabit optics are inserted. The 48 ports on ACX5448 router can be configured as 1GE or 10GE modes and these ports are represented by **xe** interface type. The PIC 1 of FPC 0 has 4x100GE ports, where each port can be channelized as 1x100GE, or 1x40GE, or 4x25GE modes and these ports are represented by **et** interface type. By default, the port speed in PIC 1 is 100GE.

NOTE: The ACX5448 router do not support Pseudowire Services interface.

NOTE: 40GbE is supported only on ACX5048, ACX5096, and ACX5448 routers. ACX5448 router support 40GbE channeling to 10GbE.

T1 and E1 Time-Division Multiplexing (TDM) Interfaces

On the ACX Series routers, existing Junos OS TDM features are supported without changes to statements or functionality. The following key TDM features for T1 (**ct1**) interfaces and E1 (**ce1**) interfaces are supported:

- T1 and E1 channelization
- T1 and E1 encapsulation
- Alarms, defects, and statistics
- External and internal loopback
- TDM class of service (CoS)

T1 and E1 mode selection is at the PIC level. To set the T1 or E1 mode at the PIC level, include the **framing** statement with the **t1** or **e1** option at the [**chassis fpc *slot-number* pic *slot-number***] hierarchy level. All ports can be T1 or E1. Mixing T1s and E1s is not supported.

T1 or E1 BITS Interface (ACX2000)

The ACX2000 router has a T1 or E1 building-integrated timing supply (BITS) interface that you can connect to an external clock. After you connect the interface to the external clock, you can configure the BITS interface so that the BITS interface becomes a candidate source for chassis synchronization to the external clock. The frequency of the BITS interface depends on the Synchronous Ethernet equipment slave clock (EEC) selected with the **network-option** statement at the [**edit chassis synchronization**] hierarchy level.

NOTE: The ACX1000 router does not support the BITS interface.

Inverse Multiplexing for ATM (IMA)

Defined by the ATM Forum, IMA specification version 1.1 is a standardized technology used to transport ATM traffic over a bundle of T1 and E1 interfaces, also known as an IMA group. Up to eight links per bundle and 16 bundles per PIC are supported. The following key IMA features are supported:

- IMA Layer 2 encapsulation
- ATM CoS
- ATM policing and shaping
- Denied packets counter in the output for the **show interfaces at-*fpc/pic/port* extensive** command

Gigabit Ethernet interfaces

On the ACX Series routers, existing Junos OS Ethernet features are supported without changes to statements or functionality. The following key features are supported:

- Media type specification (ACX1000 router with Gigabit Ethernet SFP and RJ45 interfaces)
- Autonegotiation for RJ45 Gigabit Ethernet interfaces
- Event handling of SFP insertion and removal
- Explicit disabling of the physical interface
- Flow control

> **NOTE:** The ACX Series router does not support flow control based on PAUSE frames.

- Loopback
- Loss of signal (LOS) alarm
- Media access control (MAC) layer features
- Maximum transmission unit (MTU)
- Remote fault notification for 10-Gigabit Ethernet interfaces
- Statistics collection and handling
- Power over Ethernet (PoE) (ACX2000 router)
- High power mode

The Gigabit Ethernet ports on the router have the capacity to work as a 1 or 10-Gigabit Ethernet interface, depending on the type of small form-factor pluggable (SFP) transceiver inserted. When you insert an SFP+ transceiver, the interface works at the 10-Gigabit speed. When you insert an SFP transceiver, the interface works at the 1-Gigabit speed. Configuration is not required because the speed is determined automatically based on the type of inserted SFP transceiver. The dual-speed interface is automatically created with the **xe** prefix, for example, **xe-4/0/0**.

The same configuration statements are used for both speeds and CoS parameters are scaled as a percentage of the port speed. To configure a dual-speed Gigabit Ethernet interface, include the **interface xe-*fpc/pic/port*** statement at the [**edit interfaces**] hierarchy level. To display the interface speed and other details, issue the **show interfaces** command.

NOTE: You need to use industrial grade of SFP below 0dC for ACX 1100 and ACX 2100 boards.

SEE ALSO

Understanding Encapsulation on an Interface

Configuring Inverse Multiplexing for ATM (IMA) on ACX Series

Interface Names for ACX Series Universal Metro Routers | 45

TX Matrix Plus and T1600 Router (Routing Matrix) Management Ethernet Interfaces

For TX Matrix Plus Routers and for T1600 Core Routers with RE-C1800 configured in a routing matrix, the Junos OS automatically creates the router's management Ethernet interface, **em0**. To use **em0** as a management port, you must configure its logical port, **em0.0**, with a valid IP address.

When you enter the **show interfaces** command on a TX Matrix Plus router, the management Ethernet interfaces (and logical interfaces) are displayed:

`user@host>` **show interfaces ?**

```
...
em0
em0.0
...
```

NOTE: The Routing Engines in the TX Matrix Plus router and in the T1600 routers with RE-C1800 configured in a routing matrix do not support the management Ethernet interface **fxp0**, or the internal Ethernet interfaces **fxp1** or **fxp2**.

SEE ALSO

Displaying Internal Ethernet Interfaces for a Routing Matrix with a TX Matrix Plus Router

show interfaces (M Series, MX Series, T Series Routers, and PTX Series Management and Internal Ethernet)

T1600 Routers (Routing Matrix) Internal Ethernet Interfaces

On a T1600 router configured in a routing matrix, the Routing Engine (RE-TXP-LCC) and Control Board (LCC-CB) function as a unit, or host subsystem. For each host subsystem in the router, the Junos OS automatically creates two internal Ethernet interfaces, **bcm0** and **em1**, for the two Gigabit Ethernet ports on the Routing Engine.

SEE ALSO

Displaying Internal Ethernet Interfaces for a Routing Matrix with a TX Matrix Plus Router

show interfaces (M Series, MX Series, T Series Routers, and PTX Series Management and Internal Ethernet)

Physical Interface Properties

IN THIS SECTION

This topic discusses on how to configure various physical interface properties with examples.

Physical Interface Properties Overview

The software driver for each network media type sets reasonable default values for general interface properties, such as the interface's maximum transmission unit (MTU) size, receive and transmit leaky bucket properties, link operational mode, and clock source.

To modify any of the default general interface properties, include the appropriate statements at the **[edit interfaces *interface-name*]** hierarchy level:

Media MTU Overview

The media maximum transmission unit (MTU) is the largest data unit that can be forwarded without fragmentation.

The default media MTU size used on a physical interface depends on the encapsulation used on that interface. In some cases, the default IP Protocol MTU depends on whether the protocol used is IP version 4 (IPv4) or International Organization for Standardization (ISO).

The default media MTU is calculated as follows:

```
Default media MTU = Default IP MTU + encapsulation overhead
```

When you are configuring point-to-point connections, the MTU sizes on both sides of the connections must be the same. Also, when you are configuring point-to-multipoint connections, all interfaces in the subnet must use the same MTU size.

NOTE: The actual frames transmitted also contain cyclic redundancy check (CRC) bits, which are not part of the media MTU. For example, the media MTU for a Gigabit Ethernet Version 2 interface is specified as 1514 bytes, but the largest possible frame size is actually 1518 bytes; you need to consider the extra bits in calculations of MTUs for interoperability.

The physical MTU for Ethernet interfaces does not include the 4-byte frame check sequence (FCS) field of the Ethernet frame.

A SONET/SDH interface operating in concatenated mode has a “c” added to the rate descriptor. For example, a concatenated OC48 interface is referred to as OC48c.

If you do not configure an MPLS MTU, the Junos OS derives the MPLS MTU from the physical interface MTU. From this value, the software subtracts the encapsulation-specific overhead and space for the maximum number of labels that might be pushed in the Packet Forwarding Engine. Currently, the software provides for three labels of four bytes each, for a total of 12 bytes.

In other words, the formula used to determine the MPLS MTU is the following:

```
MPLS MTU = physical interface MTU – encapsulation overhead – 12
```

Media MTU Sizes by Interface Type

The media maximum transmission unit (MTU) is the largest data unit that can be forwarded without fragmentation.

If you change the size of the media MTU, you must ensure that the size is equal to or greater than the sum of the protocol MTU and the encapsulation overhead.

This topic includes following information:

Media MTU Sizes by Interface Type for M5 and M7i Routers with CFEB, M10 and M10i Routers with CFEB, and M20 and M40 Routers

Table 8: Media MTU Sizes by Interface Type for M5 and M7i Routers with CFEB, M10 and M10i Routers with CFEB, and M20 and M40 Routers

Interface Type	Default Media MTU (Bytes)	Maximum MTU (Bytes)	Default IP Protocol MTU (Bytes)
Adaptive Services (MTU size not configurable)	9192	N/A	N/A
ATM	4482	9192	4470
E1/T1	1504	9192	1500
E3/T3	4474	9192	4470
Fast Ethernet	1514	1533 (4-port) 1532 (8-port) 1532 (12-port) **NOTE:** The maximum MTU for two 100Base-TX Fast Ethernet port FIC is 9192 bytes.	1500 (IPv4), 1497 (ISO)
Gigabit Ethernet	1514	9192 **NOTE:** The maximum MTU for one Gigabit Ethernet port FIC is 9192 bytes.	1500 (IPv4), 1497 (ISO)
Serial	1504	9192	1500 (IPv4), 1497 (ISO)

Table 8: Media MTU Sizes by Interface Type for M5 and M7i Routers with CFEB, M10 and M10i Routers with CFEB, and M20 and M40 Routers *(continued)*

Interface Type	Default Media MTU (Bytes)	Maximum MTU (Bytes)	Default IP Protocol MTU (Bytes)
SONET/SDH	4474	9192	4470

Media MTU Sizes by Interface Type for M40e Routers

Table 9: Media MTU Sizes by Interface Type for M40e Routers

Interface Type	Default Media MTU (Bytes)	Maximum MTU (Bytes)	Default IP Protocol MTU (Bytes)
Adaptive Services (MTU size not configurable)	9192	N/A	N/A
ATM	4482	9192	4470
E1/T1	1504	4500	1500
E3/T3	4474	4500 9192 (4-port)	4470
E3/DS3 IQ	4474	9192	4470
Fast Ethernet	1514	1533	1500 (IPv4), 1497 (ISO)
Gigabit Ethernet	1514	9192 (1- or 2-port) 9192 (4-port)	1500 (IPv4), 1497 (ISO)
Serial	1504	9192	1500 (IPv4), 1497 (ISO)

Table 9: Media MTU Sizes by Interface Type for M40e Routers *(continued)*

Interface Type	Default Media MTU (Bytes)	Maximum MTU (Bytes)	Default IP Protocol MTU (Bytes)
SONET/SDH	4474	4500 (1-port nonconcatenated) 9192 (4-port OC3) 9192 (4-port OC3c) 4500 (1-port OC12) 4500 (4-port OC12) 4500 (4-port OC12c) 4500 (1-port OC48) 9192 (2-port OC3) 9192 (2-port OC3c) 9192 (1-port OC12c) 9192 (1-port OC48c) 4500 (1-port OC192) 9192 (1-port OC192c)	4470

Media MTU Sizes by Interface Type for M160 Routers

Table 10: Media MTU Sizes by Interface Type for M160 Routers

Interface Type	Default Media MTU (Bytes)	Maximum MTU (Bytes)	Default IP Protocol MTU (Bytes)
Adaptive Services (MTU size not configurable)	9192	N/A	N/A
ATM	4482	9192	4470
E1/T1	1504	4500	1500

Table 10: Media MTU Sizes by Interface Type for M160 Routers *(continued)*

Interface Type	Default Media MTU (Bytes)	Maximum MTU (Bytes)	Default IP Protocol MTU (Bytes)
E3/T3	4474	4500	4470
E3/DS3 IQ	4474	9192	4470
Fast Ethernet	1514	1533	1500 (IPv4), 1497 (ISO)
Gigabit Ethernet	1514	9192 (1- or 2-port) 4500 (4-port)	1500 (IPv4), 1497 (ISO)
Serial	1504	9192	1500 (IPv4), 1497 (ISO)
SONET/SDH	4474	4500 (1-port nonconcatenated) 9192 (1- or 2-port) 4500 (4-port)	4470

Media MTU Sizes by Interface Type for M7i Routers with CFEB-E, M10i Routers with CFEB-E, and M320 and M120 Routers

Table 11: Media MTU Sizes by Interface Type for M7i Routers with CFEB-E, M10i Routers with CFEB-E, and M320 and M120 Routers

Interface Type	Default Media MTU (Bytes)	Maximum MTU (Bytes)	Default IP Protocol MTU (Bytes)
ATM2 IQ	4482	9192	4470
Channelized DS3 IQ	4471	4500	4470
Channelized E1 IQ	1504	4500	1500
Channelized OC12 IQ	4474	9192	4470

Table 11: Media MTU Sizes by Interface Type for M7i Routers with CFEB-E, M10i Routers with CFEB-E, and M320 and M120 Routers *(continued)*

Interface Type	Default Media MTU (Bytes)	Maximum MTU (Bytes)	Default IP Protocol MTU (Bytes)
Channelized STM1 IQ	4474	9192	4470
DS3	4471	4500	4470
E1	1504	4500	1500
E3 IQ	4471	4500	4470
Fast Ethernet	1514	1533 (4-port) 1532 (8-, 12- and 48-port)	1500 (IPv4), 1497 (ISO)
Gigabit Ethernet	1514	9192	1500 (IPv4), 1497 (ISO)
SONET/SDH	4474	9192	4470
T1	1504	4500	1500
CT3 IQ (excluding M120)	4474	9192	4470

Media MTU Sizes by Interface Type for MX Series Routers

Table 12: Media MTU Sizes by Interface Type for MX Series Routers

Interface Type	Default Media MTU (Bytes)	Maximum MTU (Bytes)	Default IP Protocol MTU (Bytes)
Gigabit Ethernet	1514	• 9192 • 9500 (Junos OS 16.1R1 and later releases)	1500 (IPv4), 1488 (MPLS), 1497 (ISO)

Table 12: Media MTU Sizes by Interface Type for MX Series Routers *(continued)*

Interface Type	Default Media MTU (Bytes)	Maximum MTU (Bytes)	Default IP Protocol MTU (Bytes)
10-Gigabit Ethernet	1514	• 9192 • 9500 (Junos OS 16.1R1 and later releases)	1500 (IPv4), 1488 (MPLS), 1497 (ISO)
Multi-Rate Ethernet	1514	• 9192 • 9500 (Junos OS 16.1R1 and later releases)	1500 (IPv4), 1488 (MPLS), 1497 (ISO)
Tri-Rate Ethernet	1514	• 9192 • 9500 (Junos OS 16.1R1 and later releases)	1500 (IPv4), 1488 (MPLS), 1497 (ISO)
Channelized SONET/SDH OC3/STM1 (Multi-Rate)	1514	9192	1500 (IPv4), 1488 (MPLS), 1497 (ISO)
DS3/E3 (Multi-Rate)	1514	9192	1500 (IPv4), 1488 (MPLS), 1497 (ISO)

NOTE: Starting in Junos OS Release 16.1R1, the MTU size for a media or protocol is increased from 9192 to 9500 for Ethernet interfaces on the following MX Series MPCs:

- MPC1
- MPC2
- MPC2E
- MPC3E
- MPC4E
- MPC5E
- MPC6E

NOTE: Starting in Junos OS Release 16.1R1, the MTU size for a media or protocol is increased from 9192 to 9500 for Ethernet interfaces on the following MX Series MPCs:

- MPC1
- MPC2
- MPC2E
- MPC3E
- MPC4E
- MPC5E
- MPC6E

Starting in Junos OS Release 16.1R1, the MTU size has been increased to 16,000 bytes for certain MPCs. The MTU size for the following MPCs has been increased to 16000 bytes:

- MPC7E (MPC7E-MRATE and MP7E-10G)
- MPC8E (MX2K-MPC8E)
- MPC9E (MX2K-MPC9E)

Starting in Junos OS Release 17.3R1, the MTU size for MX10003 MPC is 16,000 bytes.

Starting in Junos OS Release 17.4R1, the MTU size for MX204 is 16,000 bytes.

In all Junos OS releases, the maximum MTU size for MX5, MX10, MX40, and MX80 routers is 9192 bytes.

In all Junos OS releases, the maximum MTU size for MPC2E-NG and MPC3E-NG is 9500 bytes.

Media MTU Sizes by Interface Type for T320 Routers

Table 13: Media MTU Sizes by Interface Type for T320 Routers

Interface Type	Default Media MTU (Bytes)	Maximum MTU (Bytes)	Default IP Protocol MTU (Bytes)
ATM	4482	9192	4470
ATM2 IQ	4482	9192	4470
Channelized OC12 IQ	4474	9192	4470

Table 13: Media MTU Sizes by Interface Type for T320 Routers *(continued)*

Interface Type	Default Media MTU (Bytes)	Maximum MTU (Bytes)	Default IP Protocol MTU (Bytes)
Channelized STM1 IQ	4474	9192	4470
DS3	4471	4500	4470
Fast Ethernet	1514	1533 (4-port) 1532 (12- and 48-port)	1500 (IPv4), 1497 (ISO)
Gigabit Ethernet	1514	9192	1500 (IPv4), 1497 (ISO)
SONET/SDH	4474	9192	4470
CT3 IQ	4474	9192	4470

Media MTU Sizes by Interface Type for T640 Platforms

Table 14: Media MTU Sizes by Interface Type for T640 Platforms

Interface Type	Default Media MTU (Bytes)	Maximum MTU (Bytes)	Default IP Protocol MTU (Bytes)
ATM2 IQ	4482	9192	4470
48-port Fast Ethernet	1514	1532	1500 (IPv4), 1497 (ISO)
Gigabit Ethernet	1514	9192	1500 (IPv4), 1497 (ISO)
SONET/SDH	4474	9192	4470
CT3 IQ	4474	9192	4470

Media MTU Sizes by Interface Type for EX Series Switches and ACX Series Routers

Table 15: Media MTU Sizes by Interface Type for EX Series Switches

Interface Type	Default Media MTU (Bytes)	Maximum MTU (Bytes)	Default IP Protocol MTU (Bytes)
Gigabit Ethernet	1514	9216	1500 (IPv4), 1497 (ISO)
10-Gigabit Ethernet	1514	9216	1500 (IPv4), 1497 (ISO)

Table 16: Media MTU Sizes by Interface Type for ACX Series Routers

Interface Type	Switch	Default Media MTU (Bytes)	Maximum MTU (Bytes)	Default IP Protocol MTU (Bytes)
Gigabit Ethernet and 10-Gigabit Ethernet	ACX1000, ACX2000, ACX4000, ACX5048, ACX5096 line of routers, and ACX500.	1514	9216	1500 (IPv4), 1497 (ISO)
Gigabit Ethernet and 10-Gigabit Ethernet	ACX5448 series and ACX710 Series	1514	10000	1500 (IPv4), 1497 (ISO)

NOTE: On ACX Series routers, you can configure the protocol MTU by including the **mtu** statement at the **[edit interfaces *interface-name* unit *logical-unit-number* family inet]** or **[edit interfaces *interface-name* unit *logical-unit-number* family inet6]** hierarchy level.

- If you configure the protocol MTU at any of these hierarchy levels, the configured value is applied to all families that are configured on the logical interface.
- If you are configuring the protocol MTU for both **inet** and **inet6** families on the same logical interface, you must configure the same value for both the families. It is not recommended to configure different MTU size values for **inet** and **inet6** families that are configured on the same logical interface.

Media MTU Sizes by Interface Type for PTX Series Packet Transport Routers

Table 17: Media MTU Sizes by Interface Type for PTX Series Packet Transport Routers

Interface Type	Default Media MTU (Bytes)	Maximum MTU (Bytes)	Default IP Protocol MTU (Bytes)
10-Gigabit Ethernet	1514	9500	1500 (IPv4), 1488 (MPLS), 1497 (ISO)
40-Gigabit Ethernet	1514	9500	1500 (IPv4), 1488 (MPLS), 1497 (ISO)
100-Gigabit Ethernet	1514	9500	1500 (IPv4), 1488 (MPLS), 1497 (ISO)

Media MTU Sizes by Interface Type for JRR200 Series Routers

Table 18: Media MTU Sizes by Interface Type for JRR200 Series Routers

Interface Type	Default Media MTU (Bytes)	Maximum MTU (Bytes)	Default IP Protocol MTU (Bytes)
Management Ethernet Interfaces (**em0**,**em2 -em9**)	1514	9192	1500 (IPv4), 1497 (ISO)

Configuring the Media MTU

The media maximum transmission unit (MTU) is the largest data unit that can be forwarded without fragmentation. The default media MTU size used on a physical interface depends on the encapsulation being used on that interface. For a listing of MTU sizes for each encapsulation type, see "Media MTU Sizes by Interface Type" on page 74.

To configure the media-MTU size:

1. In configuration mode, go to the **[edit interfaces *interface-name*]** hierarchy level.

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

2. Include the **mtu** statement.

```
[edit interfaces interface-name]
mtu bytes;
```

- If you change the size of the media MTU, you must ensure that the size is equal to or greater than the sum of the protocol MTU and the encapsulation overhead. You configure the protocol MTU by including the **mtu** statement at the following hierarchy levels:
  - **[edit interfaces *interface-name* unit *logical-unit-number* family *family*]**
  - **[edit logical-systems *logical-system-name* interfaces *interface-name* unit *logical-unit-number* family *family*]**

NOTE:

- Changing the media MTU or protocol MTU causes an interface to be deleted and added again.
- Because tunnel services interfaces are considered logical interfaces, you cannot configure the MTU setting for the physical interface. This means you cannot include the mtu statement at the [edit interfaces interface-name] hierarchy level for the following interface types: generic routing encapsulation (gr-), IP-IP (ip-), loopback (lo-), link services (ls-), multilink services (ml-), and multicast (pe-, pd-). You can, however, configure the protocol MTU on all tunnel interfaces except virtual tunnel (vt) interfaces. Starting in Junos OS Release 17.1R3, you cannot configure the maximum transmission unit (MTU) size for vt interfaces because the **mtu bytes** option is deprecated for vt interfaces. Junos OS sets the MTU size for vt interfaces by default to unlimited.
- If you configure an MTU value by including the mtu statement at the [edit interfaces interface-name unit logical-unit-number family mpls] hierarchy level, the configured value is used.

Configuring the Media MTU on ACX Series Routers

IN THIS SECTION

Media MTU Overview

The default media MTU size used on a physical interface depends on the encapsulation used on that interface. In some cases, the default IP Protocol MTU depends on whether the protocol used is IP version 4 (IPv4) or International Organization for Standardization (ISO).

The default media MTU is calculated as follows:

```
Default media MTU = Default IP MTU + encapsulation overhead
```

When you are configuring point-to-point connections, the MTU sizes on both sides of the connections must be the same. Also, when you are configuring point-to-multipoint connections, all interfaces in the subnet must use the same MTU size.

NOTE: The actual frames transmitted also contain cyclic redundancy check (CRC) bits, which are not part of the media MTU. For example, the media MTU for a Gigabit Ethernet Version 2 interface is specified as 1514 bytes, but the largest possible frame size is actually 1518 bytes; you need to consider the extra bits in calculations of MTUs for interoperability.

The physical MTU for Ethernet interfaces does not include the 4-byte frame check sequence (FCS) field of the Ethernet frame.

If you do not configure an MPLS MTU, the Junos OS derives the MPLS MTU from the physical interface MTU. From this value, the software subtracts the encapsulation-specific overhead and space for the maximum number of labels that might be pushed in the Packet Forwarding Engine. Currently, the software provides for three labels of four bytes each, for a total of 12 bytes.

In other words, the formula used to determine the MPLS MTU is the following:

```
MPLS MTU = physical interface MTU – encapsulation overhead – 12
```

If you configure an MTU value by including the **mtu** statement at the **[edit interfaces *interface-name* unit *logical-unit-number* family mpls]** hierarchy level, the configured value is used. Junos OS Release 16.2R1.6 and later releases do not support **family mpls** MTU.

How to Configure the Media MTU

To modify the default media MTU size for a physical interface, include the **mtu** statement at the **[edit interfaces *interface-name*]** hierarchy level:

```
[edit interfaces interface-name]
mtu bytes;
```

If you change the size of the media MTU, you must ensure that the size is equal to or greater than the sum of the protocol MTU and the encapsulation overhead.

NOTE: Changing the media MTU or protocol MTU causes an interface to be deleted and added again.

You configure the protocol MTU by including the **mtu** statement at the following hierarchy levels:

- **[edit interfaces *interface-name* unit *logical-unit-number* family inet]**
- **[edit interfaces *interface-name* unit *logical-unit-number* family inet6]**

If you configure the protocol MTU at any of these hierarchy levels, the configured value is applied to all families that are configured on the logical interface.

NOTE: If you are configuring the protocol MTU for both **inet** and **inet6** families on the same logical interface, you must configure the same value for both the families. It is not recommended to configure different MTU size values for **inet** and **inet6** families that are configured on the same logical interface.

Encapsulation Overhead by Interface Encapsulation Type

If you change the size of the media MTU, you must ensure that the size is equal to or greater than the sum of the protocol MTU and the encapsulation overhead. The following table lists the interface encapsulation and corresponding encapsulation overhead.

Table 19: Encapsulation Overhead by Encapsulation Type

Interface Encapsulation	Encapsulation Overhead (Bytes)
802.1Q/Ethernet 802.3	21
802.1Q/Ethernet Subnetwork Access Protocol (SNAP)	26
802.1Q/Ethernet version 2	18
ATM Cell Relay	4
ATM permanent virtual connection (PVC)	12
Cisco HDLC	4
Ethernet 802.3	17
Ethernet circuit cross-connect (CCC) and virtual private LAN service (VPLS)	4
Ethernet over ATM	32
Ethernet SNAP	22

Table 19: Encapsulation Overhead by Encapsulation Type *(continued)*

Interface Encapsulation	Encapsulation Overhead (Bytes)
Ethernet translational cross-connect (TCC)	18
Ethernet version 2	14
Extended virtual local area network (VLAN) CCC and VPLS	4
Extended VLAN TCC	22
Frame Relay	4
PPP	4
VLAN CCC	4
VLAN VPLS	4
VLAN TCC	22

Configuring Interface Description

You can include a text description of each physical interface in the configuration file. Any descriptive text you include is displayed in the output of the **show interfaces** commands, and is also exposed in the **ifAlias** Management Information Base (MIB) object. It has no impact on the interface's configuration.

To add a text description, include the **description** statement at the **[edit interfaces *interface-name*]** hierarchy level:

```
[edit]
user@host# set interfaces interface-name  description text
```

For example:

```
[edit]
user@host# set interfaces fe-0/0/1 description "Backbone connection to PHL01"
```

The description can be a single line of text. If the text contains spaces, enclose it in quotation marks.

NOTE: You can configure the extended DHCP relay to include the interface description in the option 82 Agent Circuit ID suboption. See *Using DHCP Relay Agent Option 82 Information* in the *Junos OS Broadband Subscriber Management and Services Library*.

For information about describing logical units, see "Adding a Logical Unit Description to the Configuration" on page 152.

To display the description from the router or switch CLI, use the **show interfaces** command:

```
user@host> show interfaces fe-0/0/1
```

```
Physical interface: fe-0/0/1, Enabled, Physical link is Up
  Interface index: 129, SNMP ifIndex: 23
  Description: Backbone connection to PHL01
  ...
```

To display the interface description from the interfaces MIB, use the **snmpwalk** command from a server. To isolate information for a specific interface, search for the interface index shown in the **SNMP ifIndex** field of the **show interfaces** command output. The **ifAlias** object is in **ifXTable**.

```
user-server> snmpwalk host-fxp0.mylab public ifXTable | grep -e '\.23'
 snmpwalk host-fxp0.mylab public ifXTable | grep -e '\.23'
ifMIB.ifMIBObjects.ifXTable.ifXEntry.ifName.23 = fe-0/0/1
ifMIB.ifMIBObjects.ifXTable.ifXEntry.ifInMulticastPkts.23 = Counter32: 0
ifMIB.ifMIBObjects.ifXTable.ifXEntry.ifInBroadcastPkts.23 = Counter32: 0
ifMIB.ifMIBObjects.ifXTable.ifXEntry.ifOutMulticastPkts.23 = Counter32: 0
ifMIB.ifMIBObjects.ifXTable.ifXEntry.ifOutBroadcastPkts.23 = Counter32: 0
ifMIB.ifMIBObjects.ifXTable.ifXEntry.ifHCInOctets.23 = Counter64: 0
ifMIB.ifMIBObjects.ifXTable.ifXEntry.ifHCInUcastPkts.23 = Counter64: 0
ifMIB.ifMIBObjects.ifXTable.ifXEntry.ifHCInMulticastPkts.23 = Counter64: 0
ifMIB.ifMIBObjects.ifXTable.ifXEntry.ifHCInBroadcastPkts.23 = Counter64: 0
ifMIB.ifMIBObjects.ifXTable.ifXEntry.ifHCOutOctets.23 = Counter64: 42
ifMIB.ifMIBObjects.ifXTable.ifXEntry.ifHCOutUcastPkts.23 = Counter64: 0
ifMIB.ifMIBObjects.ifXTable.ifXEntry.ifHCOutMulticastPkts.23 = Counter64: 0
ifMIB.ifMIBObjects.ifXTable.ifXEntry.ifHCOutBroadcastPkts.23 = Counter64: 0
```

```
ifMIB.ifMIBObjects.ifXTable.ifXEntry.ifLinkUpDownTrapEnable.23 = enabled(1)
ifMIB.ifMIBObjects.ifXTable.ifXEntry.ifHighSpeed.23 = Gauge32: 100
ifMIB.ifMIBObjects.ifXTable.ifXEntry.ifPromiscuousMode.23 = false(2)
ifMIB.ifMIBObjects.ifXTable.ifXEntry.ifConnectorPresent.23 = true(1)
ifMIB.ifMIBObjects.ifXTable.ifXEntry.ifAlias.23 = Backbone connection to PHL01
ifMIB.ifMIBObjects.ifXTable.ifXEntry.ifCounterDiscontinuityTime.23 = Timeticks:
(0) 0:00:00.00
```

SEE ALSO

Using DHCP Relay Agent Option 82 Information

Configuring Interface Ranges

NOTE: This task uses Junos OS for EX Series switches that does not support the Enhanced Layer 2 Software (ELS) configuration style. If your switch runs software that supports ELS, see *Configuring Interface Ranges for EX Series Switches with ELS*. For ELS details, see *Using the Enhanced Layer 2 Software CLI*.

The Junos OS allows you to group a range of identical interfaces into an *interface range*. You first specify the group of identical interfaces in the interface range. Then you can apply a common configuration to the specified interface range, reducing the number of configuration statements required and saving time while producing a compact configuration.

Configuring Interface Ranges

To configure an interface range, include the **interface-range** statement at the **[edit interfaces]** hierarchy level.

The **interface-range** statement accepts only physical networking interface names in its definition. The following interface types are supported and example CLI descriptors are shown:

- ATM—**at-*fpc/pic/port***
- Channelized—**(coc | cstm)*n-fpc/pic/port***
- DPC—**xe-*fpc/pic/port***
- E1/E3—**(e1 | e3)-*fpc/pic/port***
- Ethernet—**(xe | ge | fe)-*fpc/pic/port***
- ISDN—**isdn-*fpc/pic/port***
- Serial—**se-*fpc/pic/port***
- SONET/SDH—**so-*fpc/pic/port***
- T1/T3—**(t1 | t3)-*fpc/pic/port***

Interfaces can be grouped either as a range of interfaces or using a number range under the **interface-range** statement definition.

Interfaces in an **interface-range** definition can be added as part of a member range or as individual members or multiple members using a number range.

To specify a member range, use the **member-range** statement at the **[edit interfaces interface-range *name*]** hierarchy level.

To specify interfaces in lexical order, use the **member-range *start-range* to *end-range*** statement.

A range for a member statement should contain the following:

- *—All, specifies sequential interfaces from 0 through 47.

> **CAUTION:** The wildcard * in a member statement does not take into account the interface numbers supported by a specific interface type. Irrespective of the interface type, * includes interface numbers ranging from 0 through 47 to the interface group. Therefore, use * in a member statement with caution.

- **num**—Number, specifies one specific interface by its number.
- **[low-high]**—Numbers between low to high, specifies a range of sequential interfaces.
- **[num1, num2, num3]**—Numbers **num1**, **num2**, and **num3** specify multiple specific interfaces.

Example: Specifying an Interface Range Member Range

```
member-range ge-0/0/0 to ge-4/0/40;
```

To specify one or multiple members, use the **member** statement at the **[edit interfaces interface-range *name*]** hierarchy level.

To specify the list of interface range members individually or for multiple interfaces using regex, use the **member *list of interface names*** statement.

Example: Specifying an Interface Range Member

```
member ge-0/0/0;
member ge-0/*/*
member ge-0/[1-10]/0;
member ge-0/[1,2,3]/3;
```

Regex or wildcards are not supported for interface-type prefixes. For example, prefixes **ge**, **fe**, and **xe** must be mentioned explicitly.

An **interface-range** definition can contain both **member** and **member-range** statements within it. There is no maximum limit on the number of **member** or **member-range** statements within an interface-range. However, at least one **member** or **member-range** statement must exist within an **interface-range** definition.

Example: Interface Range Common Configuration

Configuration common to an interface range can be added as a part of the **interface-range** definition, as follows:

```
[edit]
interfaces {
  + interface-range foo {
  + member-range ge-1/0/0 to ge-4/0/40;
  + member ge-0/1/1;
  + member ge-5/[1-10]/*;
    /*Common configuration is added as part of interface-range definition*/
    mtu 256;
    hold-time up 10;
```

```
            ether-options {
                flow-control;
                speed {
                    100m;
                }
                802.3ad primary;
            }
        }
    }
```

An **interface-range** definition having just **member** or **member-range** statements and no common configurations statements is valid.

These defined interface ranges can be used in other configuration hierarchies, in places where an **interface** node exists.

Example: Interface-Range foo Used Under the Protocols Hierarchy

```
    protocols {
        dot1x {
            authenticator {
                interface foo{
                    retries 1;
                }
            }
        }
    }
```

foo should be an **interface-range** defined at the **[interfaces]** hierarchy level. In the above example, the **interface** node can accept both individual interfaces and interface ranges.

> **TIP:** To view an interface range in expanded configuration, use the **(show | display inheritance)** command. For more information, see the *CLI User Guide*.

By default, **interface-range** is not available to configure in the CLI where the **interface** statement is available. The following locations are supported; however, some of the hierarchies shown in this list are product specific:

- **protocols dot1x authentication interface**
- **protocols dvmrp interface**
- **protocols oam ethernet lmi interface**
- **protocols esis interface**
- **protocols igmp interface**
- **protocols igmp-host client *num* interface**
- **protocols mld-host client *num* interface**
- **protocols router-advertisement interface**
- **protocols isis interface**
- **protocols ldp interface**
- **protocols oam ethernet link-fault-management interface**
- **protocols lldp interface**
- **protocols link-management peer lmp-control-channel interface**
- **protocols link-management peer control-channel**
- **protocols link-management te-link *name* interface**
- **protocols mld interface**
- **protocols ospf area *id* interface**
- **protocols pim interface**
- **protocols router-discovery interface**
- **protocols rip group *name* neighbour**
- **protocols ripng group *name* neighbour**
- **protocols rsvp interface**
- **protocols snmp interface**
- **protocols layer2-control bpdu-block interface**
- **protocols layer2-control mac-rewrite interface**
- **protocols mpls interface**
- **protocols stp interface**
- **protocols rstp interface**

- **protocols mstp interface**
- **protocols vstp interface**
- **protocols mstp msti *id* interface**
- **protocols mstp msti vlan *id* interface**
- **protocols vstp vlan *name* interface**
- **protocols gvrp interface**
- **protocols igmp-snooping vlan *name* interface**
- **protocols lldp interface**
- **protocols lldp-med interface**
- **protocols sflow interfaces**
- **ethernet-switching-options analyzer *name* input [egress | ingress] interface**
- **ethernet-switching-options analyzer *name* output interface**
- **ethernet-switching-options secure-access-port interface**
- **ethernet-switching-options interfaces ethernet-switching-options voip interface**
- **ethernet-switching-options redundant-trunk-group group g1 interface**
- **ethernet-switching-options redundant-trunk-group group g1 interface**
- **ethernet-switching-options bpdu-block interface**
- **poe interface vlans pro-bng-mc1-bsd1 interface**

SEE ALSO

Expanding Interface Range Member and Member Range Statements

Configuration Inheritance for Member Interfaces

Member Interfaces Inheriting Configuration from Configuration Groups

Interfaces Inheriting Common Configuration

Configuring Inheritance Range Priorities

Configuration Expansion Where Interface Range Is Used

Physical Interfaces

Expanding Interface Range Member and Member Range Statements

All **member** and **member-range** statements in an interface range definition are expanded to generate the final list of interface names for the specified interface range.

Example: Expanding Interface Range Member and Member Range Statements

```
[edit]
  interfaces {
  interface-range range-1 {
    member-range ge-0/0/0 to ge-4/0/20;
    member ge-10/1/1;
    member ge-5/[0-5]/*;
    /*Common configuration is added part of the interface-range definition*/
    mtu 256;
    hold-time up 10;
    ether-options {
      flow-control;
      speed {
        100m;
      }
      802.3ad primary;
    }
  }
}
```

For the **member-range** statement, all possible interfaces between **start-range** and **end-range** are considered in expanding the members. For example, the following **member-range** statement:

member-range ge-0/0/0 to ge-4/0/20

expands to:

```
[ge-0/0/0, ge-0/0/1 ... ge-0/0/max_ports
 ge-0/1/0  ge-0/1/1 ... ge-0/1/max_ports
 ge-0/2/0  ge-0/2/1 ... ge-0/2/max_ports
                     .
                     .
 ge-0/MAX_PICS/0 ... ge-0/max_pics/max_ports
 ge-1/0/0  ge-1/0/1 ... ge-1/0/max_ports
                     .
 ge-1/MAX_PICS/0 ... ge-1/max_pics/max_ports
                     .
```

```
                             .
        ge-4/0/0 ge-4/0/1  ... ge-4/0/max_ports]
```

The following **member** statement:

ge-5/[0-5]/*

expands to:

```
ge-5/0/0 ... ge-5/0/max_ports
ge-5/1/0 ... ge-5/0/max_ports
    .
     .
ge-5/5/0 ... ge-5/5/max_ports
```

The following **member** statement:

ge-5/1/[2,3,6,10]

expands to:

```
ge-5/1/2
ge-5/1/3
ge-5/1/6
ge-5/1/10
```

Configuration Inheritance for Member Interfaces

When the Junos OS expands the **member** and **member-range** statements present in an **interface-range**, it creates *interface objects* if they are not explicitly defined in the configuration. The common configuration is copied to all its member interfaces in the **interface-range**.

Example: Configuration Priorities

Foreground interface configuration takes priority compared to configuration inherited by the interface through the **interface-range**.

```
interfaces {
```

```
        interface-range range-1 {
            member-range ge-1/0/0/ to ge-10/0/47;
            mtu 256;
            }
            ge-1/0/1 {
                mtu 1024;
            }
    }
```

In the preceding example, interface **ge-1/0/1** will have an MTU value of 1024.

This can be verified with output of the **show interfaces | display inheritance** command, as follows:

`user@host: #` **show interfaces | display inheritance**

```
## 'ge-1/0/0' was expanded from interface-range 'range-1'
##
ge-1/0/0 {
    ##
    ## '256' was expanded from interface-range 'range-1'
    ##
    mtu 256;
}
ge-1/0/1 {
    mtu 1024;
}
##
## 'ge-1/0/2' was expanded from interface-range 'range-1'
##
ge-1/0/2 {
    ##
    ## '256' was expanded from interface-range 'range-1'
    ##
    mtu 256;
}
      .........
      .........
##
## 'ge-10/0/47' was expanded from interface-range 'range-1'
##
ge-10/0/47 {
    ##
```

```
    ## '256' was expanded from interface-range 'range-1'
    ##
    mtu 256;
}
```

Member Interfaces Inheriting Configuration from Configuration Groups

Interface range member interfaces inherit the config-groups configuration like any other foreground configuration. **interface-range** is similar to any other foreground configuration statement. The only difference is that the **interface-range** goes through a member interfaces expansion before Junos OS reads this configuration.

```
groups {
global {
  interfaces {
    <*> {
      hold-time up 10;
      }
    }
    }
    }
  apply-groups [global];
    interfaces {
      interface-range range-1 {
        member-range ge-1/0/0 to ge-10/0/47;
        mtu 256;
      }
    }
```

The **hold-time** configuration is applied to all members of **interface-range *range-1***.

This can be verified with **show interfaces | display inheritance** as follows:

```
user@host# show interfaces | display inheritance
```

```
ge-1/0/0 {
    ##
    ## '256' was expanded from interface-range 'range-1'
    ##
    mtu 256;
    ##
    ## 'hold-time' was inherited from group 'global'
```

```
        ## '10' was inherited from group 'global'
        ##
        hold-time up 10;
    }
    ge-1/0/1 {
        ##
        ## '256' was expanded from interface-range 'range-1'
        ##
        mtu 256;
        ##
        ## 'hold-time' was inherited from group 'global'
        ## '10' was inherited from group 'global'
        ##
        hold-time up 10;
    }
    ge-10/0/47 {
        ##
        ## '256' was expanded from interface-range 'range-1'
        ##
        mtu 256;
        ##
        ## 'hold-time' was inherited from group 'global'
        ## '10' was inherited from group 'global'
        ##
        hold-time up 10;
    }
```

SEE ALSO

Using Wildcards with Configuration Groups

Interfaces Inheriting Common Configuration

If an interface is a member of several interface ranges, that interface will inherit the common configuration from all of those interface ranges.

```
[edit]
interfaces {
    interface-range range-1 {
        member-range ge-1/0/0 to ge-10/0/47;
        mtu 256;
```

```
    }
    }
    interfaces {
        interface-range range-1 {
            member-range ge-10/0/0 to ge-10/0/47;
            hold-time up 10;
        }
    }
```

In this example, interfaces **ge-10/0/0** through **ge-10/0/47** will have both **hold-time** and **mtu**.

Configuring Inheritance Range Priorities

The interface ranges are defined in the order of inheritance priority, with the first interface range configuration data taking priority over subsequent interface ranges.

```
[edit]
interfaces {
    interface-range int-grp-one {
        member-range ge-0/0/0 to ge-4/0/40;
        member ge-1/1/1;
        /*Common config is added part of the interface-range definition*/
        mtu 256;
        hold-time up 10;
    }
    }
    interfaces {
        interface-range int-grp-two {
            member-range ge-5/0/0 to ge-10/0/40;
            member ge-1/1/1;
            mtu 1024;
        }
    }
```

Interface **ge-1/1/1** exists in both **interface-range *int-grp-one*** and **interface-range *int-grp-two***. This interface inherits **mtu *256*** from **interface-range *int-grp-one*** because it was defined first.

Configuration Expansion Where Interface Range Is Used

In this example, **interface-range *range-1*** is used under the **protocols** hierarchy:

```
[edit]
```

```
interfaces {
    interface-range range-1 {
        member ge-10/1/1;
        member ge-5/5/1;
        mtu 256;
        hold-time up 10;
        ether-options {
            flow-control;
            speed {
                100m;
            }
            802.3ad primary;
        }
    }
    protocols {
        dot1x {
            authenticator {
                interface range-1 {
                    retries 1;
                }
            }
        }
    }
}
```

The **interface** node present under **authenticator** is expanded into member interfaces of the **interface-range** ***range-1*** as follows:

```
protocols {
    dot1x {
        authenticator {
            interface ge-10/1/1 {
                retries 1;
            }
            interface ge-5/5/1 {
                retries 1;
            }
        }
    }
}
```

The **interface** ***range-1*** statement is expanded into two interfaces, ge-10/1/1 and ge-5/5/1, and configuration **retries** ***1*** is copied under those two interfaces.

This configuration can be verified using the **show protocols dot1x | display inheritance** command.

SEE ALSO

Physical Interfaces

Specifying an Aggregated Interface

The M Series, MX Series, and T Series routers support aggregated interfaces. To specify an aggregated interface assign a number with the aggregated interface name. For example, configure **aex** at the **[edit interfaces]** hierarchy level, where *x* is an integer ranging 0 through 127 for M Series and T Series routers and 0 through 479 on MX Series routers.

For aggregated SONET/SDH interfaces, configure **asx** at the **[edit interfaces]** hierarchy level.

> **NOTE:** SONET/SDH aggregation is proprietary to the Junos OS and might not work with other software.

If you are configuring VLANs for aggregated Ethernet interfaces, you must include the **vlan-tagging** statement at the **[edit interfaces aex]** hierarchy level to complete the association.

SEE ALSO

Aggregated Ethernet Interfaces Overview

Configuring Aggregated SONET/SDH Interfaces

Configuring the Interface Speed

IN THIS SECTION

You can configure the interface speed in following ways:

Configuring the Interface Speed on Ethernet Interfaces

For M Series and T Series Fast Ethernet 12-port and 48-port PIC interfaces, the management Ethernet interface (**fxp0** or **em0**), and the MX Series Tri-Rate Ethernet copper interfaces, you can explicitly set the interface speed. The Fast Ethernet, **fxp0**, and **em0** interfaces can be configured for 10 Mbps or 100 Mbps **(10m | 100m)**. The MX Series Tri-Rate Ethernet copper interfaces can be configured for 10 Mbps, 100 Mbps, or 1 Gbps **(10m | 100m | 1g)**. For information about management Ethernet interfaces and to determine the management Ethernet interface type for your router, see *Understanding Management Ethernet Interfaces* and *Supported Routing Engines by Router*MX Series routers, with MX-DPC and Tri-Rate Copper SFPs, support 20x1 Copper to provide backwards compatibility with 100/10BASE-T and 1000BASE-T operation through an Serial Gigabit Media Independent Interface (SGMII) interface.

1. In configuration mode, go to the **[edit interfaces interface-name]** hierarchy level.

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

2. To configure the speed, include the **speed** statement at the **[edit interfaces *interface-name*]** hierarchy level.

```
[edit interfaces interface-name]
user@host# set speed (10m | 100m | 1g | auto | auto-10m-100m);
```

NOTE:

- By default, the M Series and T Series routers management Ethernet interface autonegotiates whether to operate at 10 megabits per second (Mbps) or 100 Mbps. All other interfaces automatically choose the correct speed based on the PIC type and whether the PIC is configured to operate in multiplexed mode (using the **no-concatenate** statement in the **[edit chassis]** configuration hierarchy.
- Starting with Junos OS Release 14.2 the **auto-10m-100m** option allows the fixed tri-speed port to auto negotiate with ports limited by **100m** or **10m**maximum speed. This option must be enabled only for Tri-rate MPC port, that is, 3D 40x 1GE (LAN) RJ45 MIC on MX platform. This option does not support other MICs on MX platform.,
- When you manually configure Fast Ethernet interfaces on the M Series and T Series routers, link mode and speed must both be configured. If both these values are not configured, the router uses autonegotiation for the link and ignores the user-configured settings.
- If the link partner does not support autonegotiation, configure either Fast Ethernet port manually to match its link partner's speed and link mode. When the link mode is configured, autonegotiation is disabled.
- On MX Series routers with tri-rate copper SFP interfaces, if the port speed is negotiated to the configured value and the negotiated speed and interface speed do not match, the link will not be brought up.
- When you configure the Tri-Rate Ethernet copper interface to operate at 1 Gbps, autonegotiation must be enabled.
- Starting with Junos OS Release 11.4, half-duplex mode is not supported on Tri-Rate Ethernet copper interfaces. When you include the **speed** statement, you must include the **link-mode full-duplex** statement at the same hierarchy level.

SEE ALSO

speed

Configuring Aggregated Ethernet Link Speed

On aggregated Ethernet interfaces, you can set the required link speed for all interfaces included in the bundle. Generally, all interfaces that make up a bundle must have the same speed. If you include in the aggregated Ethernet interface an individual link that has a speed different from the speed that you specify in the **link-speed** parameter, an error message is logged. However, there are exceptions.

Starting with Junos OS Release 13.2, aggregated Ethernet supports mixed rates and mixed modes on T640, T1600, T4000, and TX Matrix Plus routers. For example, these mixes are supported:

- Member links of different modes (WAN and LAN) for 10-Gigabit Ethernet links.
- Member links of different rates: 10-Gigabit Ethernet, 40-Gigabit Ethernet, 50-Gigabit Ethernet, 100-Gigabit Ethernet, and OC192 (10-Gigabit Ethernet WAN mode)

Starting with Junos OS Release 14.1R1 and 14.2, support for mixed rates on aggregated Ethernet bundles is extended to MX240, MX480, MX960, MX2010, and MX2020 routers.

Starting with Junos OS Release 14.2, aggregated Ethernet supports mixed link speeds on PTX Series Packet Transport Routers.

NOTE:

- Member links of 50-Gigabit Ethernet can only be configured using the 50-Gigabit Ethernet interfaces of 100-Gigabit Ethernet PIC with CFP (PD-1CE-CFP-FPC4).
- Starting with Junos OS Release 13.2, 100-Gigabit Ethernet member links can be configured using the two 50-Gigabit Ethernet interfaces of 100-Gigabit Ethernet PIC with CFP. This 100-Gigabit Ethernet member link can be included in an aggregated Ethernet link that includes member links of other interfaces as well. In releases before Junos OS Release 13.2, the 100-Gigabit Ethernet member link configured using the two 50-Gigabit Ethernet interfaces of 100-Gigabit Ethernet PIC with CFP cannot be included in an aggregated Ethernet link that includes member links of other interfaces.

To configure member links of mixed rates and mixed modes on T640, T1600, T4000, TX Matrix Plus, and PTX routers, you need to configure the **mixed** option for the [**edit interfaces aex aggregated-ether-options link-speed**] statement.

To set the required link speed:

1. Specify that you want to configure the aggregated Ethernet options.

   ```
   user@host# edit interfaces interface-name aggregated-ether-options
   ```

2. Configure the link speed.

   ```
   [edit interfaces interface-name aggregated-ether-options ]
   user@host# set link-speed speed
   ```

   speed can be in bits per second either as a complete decimal number or as a decimal number followed by the abbreviation **k** (1000), **m** (1,000,000), or **g** (1,000,000,000).

Aggregated Ethernet interfaces on the M120 router can have one of the following speeds:

- **100m**—Links are 100 Mbps.
- **10g**—Links are 10 Gbps.
- **1g**—Links are 1 Gbps.
- **oc192**—Links are OC192 or STM64c.

Aggregated Ethernet links on EX Series switches can be configured to operate at one of the following speeds:

- **10m**—Links are 10 Mbps.
- **100m**—Links are 100 Mbps.
- **1g**—Links are 1 Gbps.
- **10g**—Links are 10 Gbps.
- **50g**—Links are 50 Gbps.

Aggregated Ethernet links on T Series, MX Series, PTX Series routers, and QFX5100, QFX10002, QFX10008, and QFX10016 switches can be configured to operate at one of the following speeds:

- **100g**—Links are 100 Gbps.
- **100m**—Links are 100 Mbps.
- **10g**—Links are 10 Gbps.
- **1g**—Links are 1 Gbps.
- **40g**—Links are 40 Gbps.
- **50g**—Links are 50 Gbps.
- **80g**—Links are 80 Gbps.
- **8g**—Links are 8 Gbps.
- **mixed**—Links are of various speeds.
- **oc192**—Links are OC192.

SEE ALSO

aggregated-ether-options

Configuring SONET/SDH Interface Speed

To configure the speed of SONET/SDH interfaces in concatenated mode:

1. In configuration mode, go to the **[edit interfaces *interface-name*]** hierarchy level, where the *interface-name* is **so-*fpc/pic/port***.

```
[edit]
user@host# edit interfaces so-fpc/pic/port
```

2. Configure interface speed in concatenated mode.

   For example, each port of 4-port OC12 PIC can be configured to be in OC3 or OC12 speed independently when this PIC is in 4xOC12 concatenated mode.

```
[edit interfaces so-fpc/pic/port]
user@host# set speed (oc3 | oc12 | oc48)
```

To configure the speed of SONET/SDH interfaces in nonconcatenated mode:

1. In configuration mode, go to the **[edit interfaces *interface-name*]** hierarchy level, where the ***interface-name*** is **so-*fpc/pic/port***.

```
[edit]
user@host# edit interfaces so-fpc/pic/port
```

2. Configure interface speed in nonconcatenated mode.

   For example, each port of 4-port OC12 PIC can be configured to be in OC3 or OC12 speed independently when this PIC is in 4xOC12 concatenated mode.

```
[edit interfaces so-fpc/pic/port]
user@host# set speed (oc3 | oc12)
```

To configure the PIC to operate in channelized (multiplexed) mode:

1. In configuration mode, go to the **[edit chassis fpc *slot-number* pic *pic-number*]** hierarchy level.

```
[edit]
user@host# [edit chassis fpc slot-number pic pic-number]
```

2. Configure the **no-concatenate** option.

```
[edit interfaces so-fpc/pic/port]
user@host# set no-concatenate
```

NOTE: On SONET/SDH OC3/STM1 (Multi-Rate) MIC with SFP, Channelized SONET/SDH OC3/STM1 (Multi-Rate) MIC with SFP, and Channelized OC3/STM1 (Multi-Rate) Circuit Emulation MIC with SFP, you cannot set the interface speed at the [**edit interfaces**] hierarchy level. To enable the speed on these MICs, you need to set the port speed at the **[edit chassis fpc *slot-number* pic *pic-number* port *port-number*]** hierarchy level.

For more information about using the **non-concatenate** statement, see the *Junos OS Administration Library*.

SEE ALSO

Configuring SONET/SDH Physical Interface Properties

SONET/SDH Interface Speed Overview

SONET/SDH Interfaces Overview

Configuring the Link Characteristics

By default, the router's management Ethernet interface, **fxp0** or **em0**, autonegotiates whether to operate in full-duplex or half-duplex mode. Fast Ethernet interfaces can operate in either full-duplex or half-duplex mode, and all other interfaces can operate only in full-duplex mode. For Gigabit Ethernet, the link partner must also be set to full duplex.

NOTE: When you configure the Tri-Rate Ethernet copper interface to operate at 1 Gbps, autonegotiation must be enabled.

NOTE: When you manually configure Fast Ethernet interfaces on the M Series and T Series routers, link mode and speed must both be configured. If both these values are not configured, the router uses autonegotiation for the link and ignores the user-configured settings.

NOTE: When the Fast Ethernet interface on Juniper Networks routers with autonegotiation enabled interoperates with a device configured to operate in half-duplex mode (autonegotiation disabled), the interface defaults to half-duplex mode after the PIC is taken offline and brought back online. This results in packet loss and cyclic redundancy check (CRC) errors.

To explicitly configure an Ethernet interface to operate in either full-duplex or half-duplex mode, include the **link-mode** statement at the **[edit interfaces *interface-name*]** hierarchy level:

```
[edit interfaces interface-name]
link-mode (full-duplex | half-duplex);
```

Interface Alias Names Overview

You can configure a textual description of a logical unit on a physical interface to be the alias of an interface name. Interface aliasing is supported only at the unit level. If you configure an alias name, the alias name is displayed instead of the interface name in the output of all **show**, **show interfaces**, and other operational mode commands. In Junos OS Release 12.3R8 and later, display of the alias can be suppressed in favor of the actual interface name by using the **display no-interface-alias** parameter along with the show command. Configuring an alias for a logical unit of an interface has no effect on how the interface on the router or switch operates.

When you configure the alias name of an interface, the CLI saves the alias name as the value of the ***interface-name*** variable in the configuration database. To enable backward compatibility with Junos OS releases in which the support for interface aliases is not available, when the Junos OS processes query the configuration database for the ***interface-name*** variable, the actual, exact value of the ***interface-name*** variable is returned instead of the alias name for system operations and computations.

This capability to define interface alias names for physical and logical interfaces is useful in a Junos Node Unifier (JNU) environment that contains a Juniper Networks MX Series 5G Universal Routing Platform as a controller and EX Series Ethernet switches, QFX Series devices, and ACX Series Universal Metro Routers as satellite devices. The following are the benefits of configuring an alias name, which enables a meaningful, single, and easily identifiable name to be allocated to an interface:

- You can group physical interfaces as one aggregated interface (link aggregation group or LAG bundle) and name that bundle as a satellite connection interface (for example, sat1).
- You can select a logical interface as a member of the LAG bundle or the entire LAG, and name that interface to represent a satellite device port or a service instance (for example, ge-0/0/1).
- You can combine the satellite name and the interface name aliases to wholly represent the satellite port name (for example, sat1:ge-0/0/1 or ge-sat1/0/0/1 or ge-1/0/0/1) in the most easily distinguishable format that denotes a combination of port and satellite parts of the name.

To specify an interface alias, you can use the **alias** statement at the **[edit interfaces *interface-name* unit *logical-unit-number*]** and **[edit logical-systems *logical-system-name* interfaces *interface-name* unit *logical-unit-number*]** hierarchy levels.

> **NOTE:** In Juniper Networks M Series Multiservice Edge Routers, if the same alias name is configured on more than one logical interface, the router displays an error message and commit fails.

SEE ALSO

alias | 293

Example: Adding an Interface Alias Name

IN THIS SECTION

This example shows how to add an alias to the logical unit of an interface. Using an alias to identify interfaces as they appear in the output for operational commands can allow for more meaningful naming conventions and easier identification.

Requirements

This example uses the following hardware and software components:

- One MX Series router that acts as a controller
- One EX4200 switch that acts as a satellite device
- Junos OS Release 13.3R1 or later

Overview

You can create an alias for each logical unit on a physical interface. The descriptive text you define for the alias is displayed in the output of the **show interfaces** commands. In Junos OS Release 12.3R8 and later, display of the alias can be suppressed in favor of the actual interface name by using the **display no-interface-alias** parameter along with the show command. The alias configured for a logical unit of an interface has no effect on how the interface on the router or switch operates – it is only a cosmetic label.

Configuration

IN THIS SECTION

Consider a scenario in which alias names are configured on the interfaces of a JNU controller that are connected to a satellite, sat1, in the downlink direction in the JNU management network by using two links. The alias names enable effective, streamlined identification of these interfaces in the operational mode commands that are run on the controller and satellites.

CLI Quick Configuration

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

```
set interfaces ae0 unit 0 alias "controller-sat1-downlink1"
set interfaces ae0.0 family inet address 10.0.0.1/24
set interfaces ae1 unit 0 alias "controller-sat1-downlink1"
set interfaces ae0.0 family inet address 192.0.2.128/25
set interfaces ge-0/0/0 vlan-tagging
```

```
set interfaces ge-0/0/0 unit 0 alias "ge-to-corp-gw1"
set interfaces ge-0/0/0.0 vlan-id 101
set interfaces ge-0/0/0.0 family inet address 1.1.1.1/23
set interfaces ge-0/1/0 gigether-options 802.3ad ae0
set interfaces ge-0/1/1 gigether-options 802.3ad ae0
set protocols rip group corporate-firewall neighbor ge-to-corp-gw1
```

Configuring Alias Names for the Controller Interfaces

Step-by-Step Procedure

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

To add an alias name to the controller interfaces that are used to connect to the satellite devices in the downlink direction:

1. Configure an alias name for the logical unit of an aggregated Ethernet interface that is used to connect to a satellite, sat1, in the downlink direction. Configure **inet** family and address for the interface.

   ```
   [edit]
   user@host# set interfaces ae0 unit 0 alias "controller-sat1-downlink1"
   user@host# set interfaces ae0.0 family inet address 10.0.0.1/24
   ```

2. Configure an alias name for the logical unit of another aggregated Ethernet interface that is used to connect to the same satellite, sat1, in downlink direction. Configure INET family and address for the interface.

   ```
   [edit]
   user@host# set interfaces ae0 unit 1 alias "controller-sat1-downlink2"
   user@host# set interfaces ae0.0 family inet address 10.0.0.3/24
   ```

3. Configure an alias name for the Gigabit Ethernet interface on the controller and configure its parameters.

   ```
   [edit]
   user@host# set interfaces ge-0/0/0 vlan-tagging
   user@host# set interfaces ge-0/0/0 unit 0 alias "ge-to-corp-gw1"
   user@host# set interfaces ge-0/0/0.0 vlan-id 101
   user@host# set interfaces ge-0/0/0.0 family inet address 1.1.1.1/23
   ```

4. Configure Gigabit Ethernet interfaces to be member links of an **ae-** logical interface.

```
[edit]
user@host# set interfaces ge-0/1/0 gigether-options 802.3ad ae0
user@host# set interfaces ge-0/1/1 gigether-options 802.3ad ae0
```

5. Configure RIP in the network between the controller and the firewall gateway.

```
[edit]
user@host# set protocols rip group corporate-firewall neighbor ge-to-corp-gw1
```

Results

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

```
[edit]
        interfaces {
            ae0 {
                unit 0 {
                    alias "controller-sat1-downlink1";
                    family inet {
                        address 10.0.0.1/24;
                    }
                }
                unit 1 {
                    alias "controller-sat1-downlink2";
                    family inet {
                        address 10.0.0.3/24;
                    }
                }
            }
            ge-0/0/0 {
                vlan-tagging;
                unit 0 {
                    alias "ge-to-corp-gw1";
                    vlan-id 101;
                    family inet {
                        address 1.1.1.1/23;
                    }
                }
            }
            ge-0/1/0 {
  gigether-options {
```

```
                    802.3ad ae0;
                }
            }
            ge-0/1/1 {
    gigether-options {
                    802.3ad ae0;
                }
            }
        }
        protocols rip {
            group corporate-firewall {
                neighbor ge-to-corp-gw1;
            }
        }
```

After you have confirmed that the interfaces are configured, enter the **commit** command in configuration mode.

> **NOTE:** In Juniper Networks M Series Multiservice Edge Routers, if the same alias name is configured on more than one logical interface, the router displays an error message and commit fails.

Verification

IN THIS SECTION

To verify that the alias name is displayed instead of the interface name, perform these steps:

Verifying the Configuration of the Alias Name for the Controller Interfaces

Purpose

Verify that the alias name is displayed instead of the interface name.

Action

Display information about all RIP neighbors.

```
user@router> show rip neighbor
                  Local  Source          Destination     Send   Receive   In
Neighbor          State  Address         Address         Mode   Mode     Met
ge-to-corp-gw1       DN  (null)          255.255.255.255 mcast  both       1
```

Meaning

The output displays the details of the benchmarking test that was performed. For more information about the **show rip neighbor** operational command, see **show rip neighbor** in the CLI Explorer.

SEE ALSO

alias | 293

Clock Source Overview

For both the router and interfaces, the clock source can be an external clock that is received on the interface or the router's internal Stratum 3 clock.

For example, interface A can transmit on interface A's received clock (external, loop timing) or the Stratum 3 clock (internal, line timing, or normal timing). Interface A cannot use a clock from any other source. For interfaces such as SONET/SDH that can use different clock sources, you can configure the source of the transmit clock on each interface.

The clock source resides on the System Control Board (SCB) for M40 routers, the System and Switch Board (SSB) for M20 routers, the Control Board (CB) for M120 routers, and the Miscellaneous Control Subsystem (MCS) for M40e and M160 routers. M7i and M10i routers have a clock source on the Compact Forwarding Engine Board (CFEB) and Enhanced Compact Forwarding Engine Board (CFEB-E).

For T Series and MX Series, the clock source internal Stratum 3 clock resides on the SONET Clock Generator and Switch Control Board (SCB) respectively. By default, the 19.44-MHz Stratum 3 reference clock generates the clock signal for all serial PICs (SONET/SDH) and Plesiochronous Digital Hierarchy (PDH) PICs. PDH PICs include DS3, E3, T1, and E1 PICs.

> **NOTE:** M7i and M10i routers do not support external clocking of SONET interfaces.

For information about clocking on channelized interfaces, see *Channelized IQ and IQE Interfaces Properties*. Also see *Configuring the Clock Source on SONET/SDH Interfaces* and *Configuring the Channelized T3 Loop Timing*.

For information about configuring an external synchronization interface that can be used to synchronize the internal Stratum 3 clock to an external source on the M40e, M120, M320, routers and T Series routers, see *Junos OS Administration Library*, *Configuring Junos OS to Support an External Clock Synchronization Interface for M Series, MX Series, and T Series Routers*.

For information about configuring Synchronous Ethernet on MX 80, MX240, MX480, and MX960 Universal Routing Platforms, see *Junos OS Administration Library*, *Synchronous Ethernet Overview* and *Configuring Clock Synchronization Interface on MX Series Routers*.

SEE ALSO

Configuring an External Synchronization Interface

Configuring Junos OS to Support an External Clock Synchronization Interface for M Series, MX Series, and T Series Routers

Synchronous Ethernet Overview

Configuring Clock Synchronization Interface on MX Series Routers

Configuring the Clock Source

For both the router and interfaces, the clock source can be an external clock that is received on the interface or the router's internal Stratum 3 clock.

To set the clock source as external or internal:

1. In configuration mode, go to the **[edit interfaces *interface-name*]** hierarchy level:

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

2. Configure the **clocking** option as external or internal.

```
[edit interfaces interface-name]
user@host# set clocking (external | internal)
```

NOTE: M7i and M10i routers do not support external clocking of SONET interfaces.

NOTE: On Channelized SONET/SDH PICs, if you set the parent (or the master) controller clock to **external**, then you must set the child controller clocks to the default value—that is, **internal**.

For example, on the Channelized STM1 PIC, if the clock on the Channelized STM1 interface (which is the master controller) is set to **external**, then you must not configure the CE1 interface (which is the child controller) clock to **external**. Instead you must configure the CE1 interface clock to **internal**.

For information about clocking on channelized interfaces, see *Channelized IQ and IQE Interfaces Properties*. Also see *Configuring the Clock Source on SONET/SDH Interfaces* and *Configuring the Channelized T3 Loop Timing*.

For information about configuring an external synchronization interface that can be used to synchronize the internal Stratum 3 clock to an external source on the M40e, M120, and M320 routers and on the T Series routers, see *Junos OS Administration Library*, *Configuring Junos OS to Support an External Clock Synchronization Interface for M Series, MX Series, and T Series Routers*.

For information about configuring Synchronous Ethernet on MX80, MX240, MX480, and MX960 Universal Routing Platforms, see *Junos OS Administration Library*, *Synchronous Ethernet Overview* and *Configuring Clock Synchronization Interface on MX Series Routers*.

SEE ALSO

Configuring an External Synchronization Interface

clocking

Configuring Junos OS to Support an External Clock Synchronization Interface for M Series, MX Series, and T Series Routers

Synchronous Ethernet Overview

Configuring Clock Synchronization Interface on MX Series Routers

Configuring Interface Encapsulation on Physical Interfaces

IN THIS SECTION

Understanding Interface Encapsulation on Physical Interfaces

Point-to-Point Protocol (PPP) encapsulation is the default encapsulation type for physical interfaces. You need not configure encapsulation for any physical interfaces that support PPP encapsulation. If you do not configure encapsulation, PPP is used by default.

For physical interfaces that do not support PPP encapsulation, you must configure an encapsulation to use for packets transmitted on the interface. You can optionally configure an encapsulation on a logical interface, which is the encapsulation used within certain packet types.

Encapsulation Capabilities of Physical Interfaces

When you configure a point-to-point encapsulation (such as PPP or Cisco HDLC) on a physical interface, the physical interface can have only one logical interface (that is, only one **unit** statement) associated with it. When you configure a multipoint encapsulation (such as Frame Relay), the physical interface can have multiple logical units, and the units can be either point-to-point or multipoint.

Ethernet CCC encapsulation for Ethernet interfaces with standard TPID tagging requires that the physical interface have only a single logical interface. Ethernet interfaces in VLAN mode can have multiple logical interfaces.

For Ethernet interfaces in VLAN mode, VLAN IDs are applicable as follows:

- VLAN ID 0 is reserved for tagging the priority of frames.
- For encapsulation type **vlan-ccc**, VLAN IDs 1 through 511 are reserved for normal VLANs. VLAN IDs 512 and above are reserved for VLAN CCCs.
- For encapsulation type **vlan-vpls**, VLAN IDs 1 through 511 are reserved for normal VLANs, and VLAN IDs 512 through 4094 are reserved for VPLS VLANs. For 4-port Fast Ethernet interfaces, you can use VLAN IDs 512 through 1024 for VPLS VLANs.
- For Gigabit Ethernet interfaces and Gigabit Ethernet IQ and IQE PICs with SFPs (except the 10-port Gigabit Ethernet PIC and the built-in Gigabit Ethernet port on the M7i router), you can configure flexible Ethernet services encapsulation on the physical interface. For interfaces with **flexible-ethernet-services** encapsulation, all VLAN IDs are valid. VLAN IDs from 1 through 511 are not reserved.
- For encapsulation types **extended-vlan-ccc** and **extended-vlan-vpls**, all VLAN IDs are valid.

The upper limits for configurable VLAN IDs vary by interface type.

When you configure a TCC encapsulation, some modifications are needed to handle VPN connections over unlike Layer 2 and Layer 2.5 links and terminate the Layer 2 and Layer 2.5 protocol locally.

The router performs the following media-specific changes:

- PPP TCC—Both Link Control Protocol (LCP) and Network Control Protocol (NCP) are terminated on the router. Internet Protocol Control Protocol (IPCP) IP address negotiation is not supported. The Junos OS strips all PPP encapsulation data from incoming frames before forwarding them. For output, the next hop is changed to PPP encapsulation.
- Cisco HDLC TCC—Keepalive processing is terminated on the router. The Junos OS strips all Cisco HDLC encapsulation data from incoming frames before forwarding them. For output, the next hop is changed to Cisco HDLC encapsulation.

- Frame Relay TCC—All Local Management Interface (LMI) processing is terminated on the router. The Junos OS strips all Frame Relay encapsulation data from incoming frames before forwarding them. For output, the next hop is changed to Frame Relay encapsulation.
- ATM—Operation, Administration, and Maintenance (OAM) and Interim Local Management Interface (ILMI) processing is terminated at the router. Cell relay is not supported. The Junos OS strips all ATM encapsulation data from incoming frames before forwarding them. For output, the next hop is changed to ATM encapsulation.

Configuring the Encapsulation on a Physical Interface

By default, PPP is the encapsulation type for physical interfaces. To configure the encapsulation on a physical interface, include the encapsulation statement at the **[edit interfaces *interface-name*]** hierarchy level:

To configure encapsulation on a physical interface:

1. In configuration mode, go to the **[edit interfaces *interface-name*]** hierarchy level.

```
[edit]
user@host# set interfaces so-fpc/pic/port
```

2. Configure the encapsulation type as described in *encapsulation*.

```
[edit interfaces mo-fpc/pic/port]
user@host# set encapsulation encapsulation-type
```

NOTE:

- When you configure a point-to-point encapsulation (such as PPP or Cisco HDLC) on a physical interface, the physical interface can have only one logical interface (that is, only one unit statement) associated with it. When you configure a multipoint encapsulation (such as Frame Relay), the physical interface can have multiple logical units, and the units can be either point-to-point or multipoint.
- When the encapsulation type is set to **Cisco-compatible Frame Relay** encapsulation, ensure that the LMI type is set to ANSI or Q933-A.
- When **vlan-vpls** encapsulation is set at the physical interface level, commit check will validate that there should not be any **inet** family configured within it.

Displaying the Encapsulation on a Physical SONET/SDH Interface

Purpose

To display the configured encapsulation and its associated set options on a physical interface when the following are set at the **[edit interfaces *interface-name*]** hierarchy level:

- interface-name—so-7/0/0
- Encapsulation—**ppp**
- Unit—0
- Family—**inet**
- Address—192.168.1.113/32
- Destination—192.168.1.114
- Family—**iso** and **mpls**

Action

Run the **show** command at the **[edit interfaces *interface-name*]** hierarchy level.

```
[edit interfaces so-7/0/0]
user@host# show
encapsulation ppp;
unit 0 {
    point-to-point;
    family inet {
        address 192.168.1.113/32 {
            destination 192.168.1.114;
        }
    }
    family iso;
    family mpls;
}
```

Meaning

The configured encapsulation and its associated set options are displayed as expected. Note that the second set of two **family** statements allow IS-IS and MPLS to run on the interface.

RELATED DOCUMENTATION

encapsulation

Configuring Interface Encapsulation on PTX Series Packet Transport Routers

This topic describes how to configure interface encapsulation on PTX Series Packet Transport Routers. Use the **flexible-ethernet-services** configuration statement to configure different encapsulation for different logical interfaces under a physical interface. With flexible Ethernet services encapsulation, you can configure each logical interface encapsulation without range restrictions for VLAN IDs.

Supported encapsulations for physical interfaces include:

- **flexible-ethernet-services**
- **ethernet-ccc**
- **ethernet-tcc**

Supported encapsulations for logical interfaces include:

- **ethernet**
- **vlan-ccc**
- **vlan-tcc**

> **NOTE:** PTX Series Packet Transport Routers do not support **extended-vlan-cc** and **extended-vlan-tcc** encapsulation on logical interfaces. Instead, you can configure a tag protocol ID (TPID) value of 0x9100 to achieve the same results.

To configure flexible Ethernet services encapsulation, include the **encapsulation flexible-ethernet-services** statement at the **[edit interfaces et-*fpc/pic/port*]** hierarchy level. For example:

```
interfaces {
    et-fpc/pic/port {
        vlan-tagging;
        encapsulation flexible-ethernet-services;
        unit 0 {
            vlan-id 1000;
            family inet {
                address 11.0.0.20/24;
            }
        }
        unit 1 {
            encapsulation vlan-ccc;
            vlan-id 1010;
        }
        unit 2 {
```

```
            encapsulation vlan-tcc;
            vlan-id 1020;
            family tcc {
                proxy {
                    inet-address 11.0.2.160;
                }
                remote {
                    inet-address 11.0.2.10;
                }
            }
        }
    }
}
```

Configuring Keepalives

By default, physical interfaces configured with Cisco HDLC or PPP encapsulation send keepalive packets at 10-second intervals. The Frame Relay term for keepalives is LMI packets; the Junos OS supports both ANSI T1.617 Annex D LMIs and ITU Q933 Annex A LMIs. On ATM networks, OAM cells perform the same function. You configure OAM cells at the logical interface level; for more information, see Defining the ATM OAM F5 Loopback Cell Period.

To disable the sending of keepalives:

1. In configuration mode, go to the **[edit interfaces *interface-name*]** hierarchy level.

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

2. Include the **no-keepalives** statement at the **[edit interfaces *interface-name*]** hierarchy level.

   ```
   [edit interfaces interface-name]
   no-keepalives;
   ```

To disable the sending of keepalives on a physical interface configured with Cisco HDLC encapsulation for a translational cross-connection:

1. In configuration mode, go to the **[edit interfaces*interface-name*]** hierarchy level.

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

2. Include the **no-keepalives** statement with the **encapsulation cisco-hdlc-tcc** statement at the **[edit interfaces *interface-name*]** hierarchy level.

```
[edit interfaces interface-name]
encapsulation cisco-hdlc-tcc;
no-keepalives;
```

To disable the sending of keepalives on a physical interface configured with PPP encapsulation for a translational cross-connection:

1. In configuration mode, go to the **[edit interfaces *interface-name*]** hierarchy level.

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

2. Include the **no-keepalives** statement with the **encapsulation ppp-tcc** statement at the **[edit interfaces *interface-name*]** hierarchy level.

```
[edit interfaces interface-name]
encapsulation ppp-tcc;
no-keepalives;
```

For more information about translation cross-connections, see Circuit and Translational Cross-Connects Overview.

When you configure PPP over ATM or Multilink PPP over ATM encapsulation, you can enable or disable keepalives on the logical interface. For more information, see Configuring PPP over ATM2 Encapsulation.

To explicitly enable the sending of keepalives:

1. In configuration mode, go to the **[edit interfaces *interface-name*]** hierarchy level.

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

2. Include the **keepalives** statement at the **[edit interfaces *interface-name*]** hierarchy level.

```
[edit interfacesinterface-name]
keepalives;
```

To change one or more of the default keepalive values:

1. In configuration mode, go to the **[edit interfaces *interface-name*]** hierarchy level.

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

2. Include the **keepalives** statement with the appropriate option as **interval*seconds***, **down-count*number***, and the **up-count*number***..

```
[edit interfaces interface-name]
keepalives;
keepalives <interval seconds> <down-count number> <up-count number>;
```

On interfaces configured with Cisco HDLC or PPP encapsulation, you can include the following three keepalive statements; note that Frame Relay encapsulation is not affected by these statements:

- **interval *seconds***—The time in seconds between successive keepalive requests. The range is from 1 second through 32767 seconds, with a default of 10 seconds.
- **down-count *number***—The number of keepalive packets a destination must fail to receive before the network takes a link down. The range is from 1 through 255, with a default of 3.
- **up-count *number***—The number of keepalive packets a destination must receive to change a link's status from down to up. The range is from 1 through 255, with a default of 1.

> **CAUTION:** If interface keepalives are configured on an interface that does not support the **keepalives** configuration statement (for example, 10-Gigabit Ethernet), the link layer may go down when the PIC is restarted. Avoid configuring the keepalives on interfaces that do not support the **keepalives** configuration statement.

For information about Frame Relay keepalive settings, see Configuring Frame Relay Keepalives.

On MX Series routers with Modular Port Concentrators/Modular Interface Cards (MPCs/MICs), the Packet Forwarding Engine on an MPC/MIC processes and responds to Link Control Protocol (LCP) Echo-Request keepalive packets that the PPP subscriber (client) initiates and sends to the router. The mechanism by which LCP Echo-Request packets are processed by the Packet Forwarding Engine instead of by the Routing

Engine is referred to as *PPP fast keepalive* For more information about how PPP fast keepalive works on an MX Series router with MPCs/MICs, see the *Junos OS Subscriber Access Configuration Guide*.

SEE ALSO

Defining the ATM OAM F5 Loopback Cell Period

Disabling the Sending of PPPoE Keepalive Messages

Understanding How the Router Processes Subscriber-Initiated PPP Fast Keepalive Requests

keepalives

no-keepalives

Configuring Frame Relay Keepalives

Circuit and Translational Cross-Connects Overview

Configuring PPP over ATM2 Encapsulation Overview

Understanding Unidirectional Traffic Flow on Physical Interfaces

By default, physical interfaces are bidirectional; that is, they both transmit and receive traffic. You can configure unidirectional link mode on a 10-Gigabit Ethernet interface that creates two new physical interfaces that are unidirectional. The new transmit-only and receive-only interfaces operate independently, but both are subordinate to the original parent interface.

The unidirectional interfaces enable the configuration of a unidirectional link topology. Unidirectional links are useful for applications such as broadband video services where almost all traffic flow is in one direction, from the provider to the user. Unidirectional link mode conserves bandwidth by enabling it to be differentially dedicated to transmit and receive interfaces. In addition, unidirectional link mode conserves ports for such applications because the transmit-only and receive-only interfaces act independently. Each can be connected to different routers, for example, reducing the total number of ports required.

NOTE: Unidirectional link mode is currently supported on only the following hardware:

- 4–port 10–Gigabit Ethernet DPC on the MX960 router
- 10–Gigabit Ethernet IQ2 PIC and 10–Gigabit Ethernet IQ2E PIC on the T Series router

The transmit-only interface is always operationally up. The operational status of the receive-only interface depends only on local faults; it is independent of remote faults and of the status of the transmit-only interface.

On the parent interface, you can configure attributes common to both interfaces, such as clocking, framing, gigether-options, and sonet-options. On each of the unidirectional interfaces, you can configure encapsulation, MAC address, MTU size, and logical interfaces.

Unidirectional interfaces support IP and IPv6. Packet forwarding takes place by means of static routes and static ARP entries. which you can configure independently on both unidirectional interfaces.

Only transmit statistics are reported on the transmit-only interface (and shown as zero on the receive-only interface). Only receive statistics are reported on the receive-only interface (and shown as zero on the transmit-only interface). Both transmit and receive statistics are reported on the parent interface.

SEE ALSO

unidirectional

Enabling Unidirectional Traffic Flow on Physical Interfaces

By default, physical interfaces are bidirectional; that is, they both transmit and receive traffic. You can configure unidirectional link mode on a 10-Gigabit Ethernet interface that creates two new physical interfaces that are unidirectional. The new transmit-only and receive-only interfaces operate independently, but both are subordinate to the original parent interface.

To enable unidirectional link mode on a physical interface, perform the following steps:

1. In configuration mode, go to the **[edit interfaces *interface-name*]** hierarchy level:

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

2. Configure the **unidirectional** option to create two new, unidirectional (transmit-only and receive-only) physical interfaces subordinate to the original parent interface.

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

NOTE: Unidirectional link mode is currently supported on only the following hardware:

- 4–port 10–Gigabit Ethernet DPC on the MX960 router
- 10–Gigabit Ethernet IQ2 PIC and 10–Gigabit Ethernet IQ2E PIC on the T Series router

SEE ALSO

unidirectional

Physical Interface Damping Overview

IN THIS SECTION

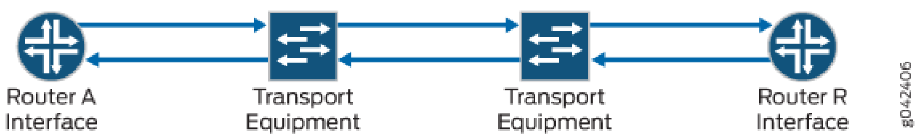
Physical interface damping limits the advertisement of the up and down transitions (flapping) on an interface. Each time a transition occurs, the interface state is changed, which generates an advertisement to the upper-level routing protocols. Damping helps reduce the number of these advertisements.

From the viewpoint of network deployment, physical interface flaps fall into the following categories:

- Nearly instantaneous multiple flaps of short duration (milliseconds).
- Periodic flaps of long duration (seconds).

Figure 6 on page 131 is used to describe these types of interface flaps and the damping configuration that you can use in each case.

Figure 6: Two Router Interfaces Connected Through Transport Equipment



> **NOTE:** We recommend that you use similar damping configurations on both ends of the physical interface. Configuring damping on one end and not having interface damping on the other end can result in undesired behavior.

The following sections describe the types of interface damping depending upon the transition time length.

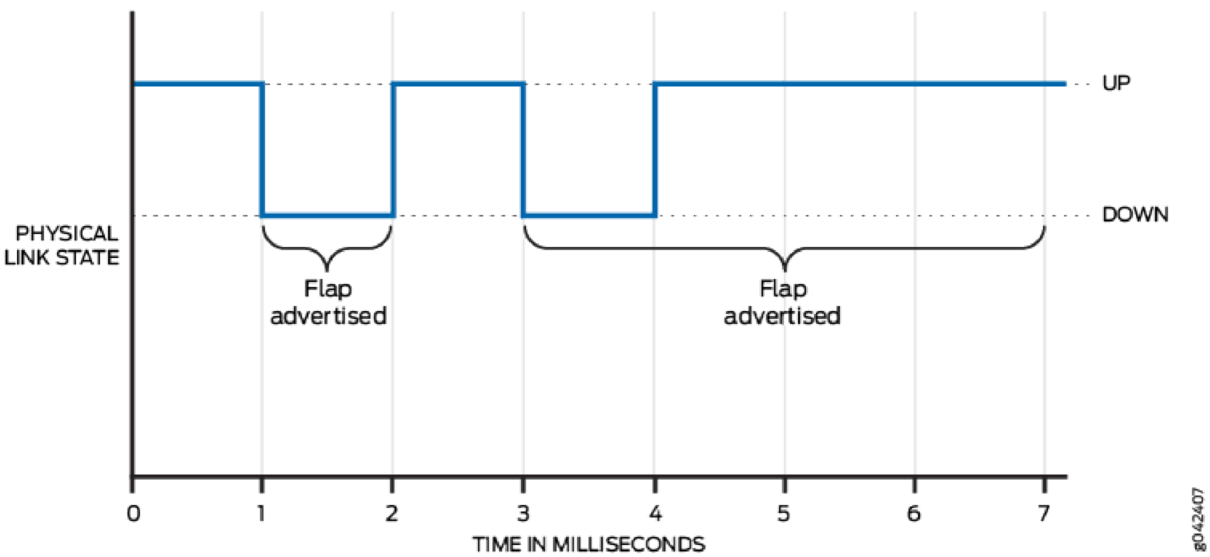
Damping Overview for Shorter Physical Interface Transitions

Figure 6 on page 131 shows two routers with two transport devices between them. If a redundant link between the two transport devices fails, link switching is performed. Link switching takes a number of milliseconds. As shown in Figure 7 on page 132, during switching, both router interfaces might encounter multiple flaps with an up-and-down duration of several milliseconds. These multiple flaps, if advertised to the upper-level routing protocols, might result in undesired route updates. This is why you might want to damp these interface flaps.

> **NOTE:** Damping is suitable only with routing protocols.

For shorter physical interface transitions, you configure interface damping with the **hold-time** statement on the interface. The hold timer enables interface damping by not advertising interface transitions until the hold timer duration has passed. When a hold-down timer is configured and the interface goes from up to down, the down hold-time timer is triggered. Every interface transition that occurs during the hold-time is ignored. When the timer expires and the interface state is still *down*, then the router begins to advertise the interface as being down. Similarly, when a hold-up timer is configured and an interface goes from down to up, the up hold-time timer is triggered. Every interface transition that occurs during the hold-time is ignored. When the timer expires and the interface state is still *up*, then the router begins to advertise the interface as being up.

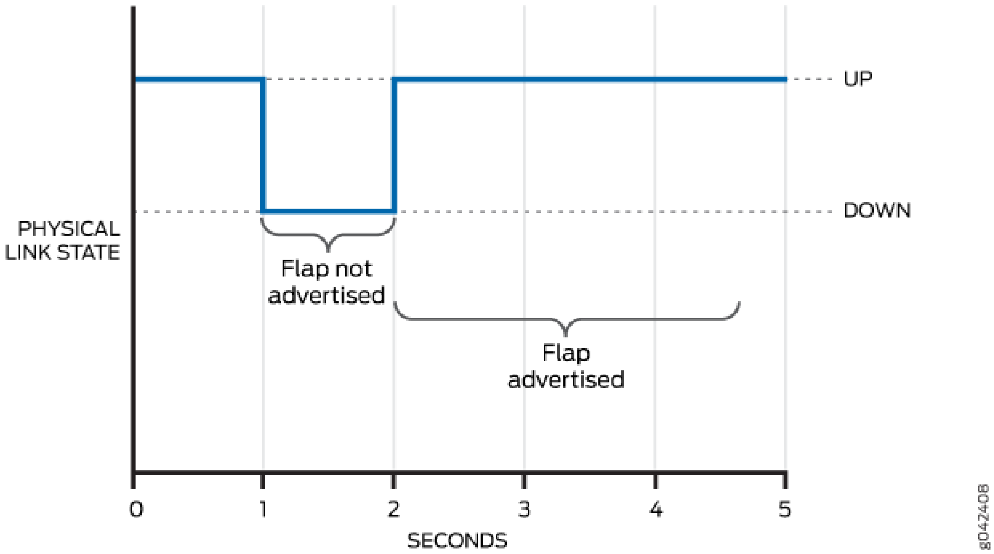
Figure 7: Multiple Flaps of Short Duration (Milliseconds)



Damping Overview for Longer Physical Interface Transitions

When the link between a router interface and the transport devices is not stable, this can lead to periodic flapping, as shown in Figure 8 on page 133. Flaps occur in the order of seconds or more, with an up-and-down flap duration in the order of a second or more. In this case, using the hold timer feature might not produce optimal results as it cannot suppress the relatively longer and repeated interface flaps. Increasing the hold time duration to seconds still allows the system to send route updates on the flapping interface, so fails to suppress periodically flapping interfaces on the system.

Figure 8: Periodic Flaps of Long Duration (Seconds)



For longer periodic interface flaps, you configure interface damping with the **damping** statement on the interface. This damping method uses an exponential back-off algorithm to suppress interface up-and-down event reporting to the upper-level protocols. Every time an interface goes down, a penalty is added to the interface penalty counter. If at some point the accumulated penalty exceeds the suppress level, the interface is placed in the suppress state, and further interface link up and down events are not reported to the upper-level protocols.

NOTE:

- Only PTX Series routers, T Series routers, MX960 routers, MX480 routers, MX240 routers, MX80 routers, and M10i routers support interface damping for longer periodic interface flaps on all the line cards.
- Penalty added on every interface flap is 1000.
- The system does not indicate whether an interface is down because of suppression or that is the actual state of the physical interface. Because of this, SNMP link traps and Operation, Administration, and Maintenance (OAM) protocols cannot differentiate the damped version of the link state from the real version. Therefore, the traps and protocols might not work as expected.
- You can verify suppression by viewing the information in the **Damping** field of the **show interface extensive** command output.

At all times, the interface penalty counter follows an exponential decay process. Figure 9 on page 135 and Figure 10 on page 136 show the decay process as it applies to recovery when the physical level link is down or up. As soon as the accumulated penalty reaches the lower boundary of the reuse level, the interface is marked as unsuppressed, and further changes in the interface link state are again reported to the upper-level protocols. You use the **max-suppress** option to configure the maximum time for restricting the accumulation of the penalty beyond the value of the maximum penalty. The value of the maximum penalty is calculated by the software. The maximum penalty corresponds to the time it would take max-suppress to decay and reach the reuse level. The penalty continues to decay after crossing the reuse level.

Figure 9 on page 135 and Figure 10 on page 136 show the accumulated penalty, and the decay over time as a curve. Whenever the penalty is below the reuse level and the physical level link changes state, state changes are advertised to the system and cause SNMP state changes.

Figure 9 on page 135 shows the penalty dropping below the reuse level when the physical link is down. The system is notified of a state change only after the physical level link transitions to up.

Figure 9: Physical-Level Link Is Down When the Penalty Falls Below the Reuse Level

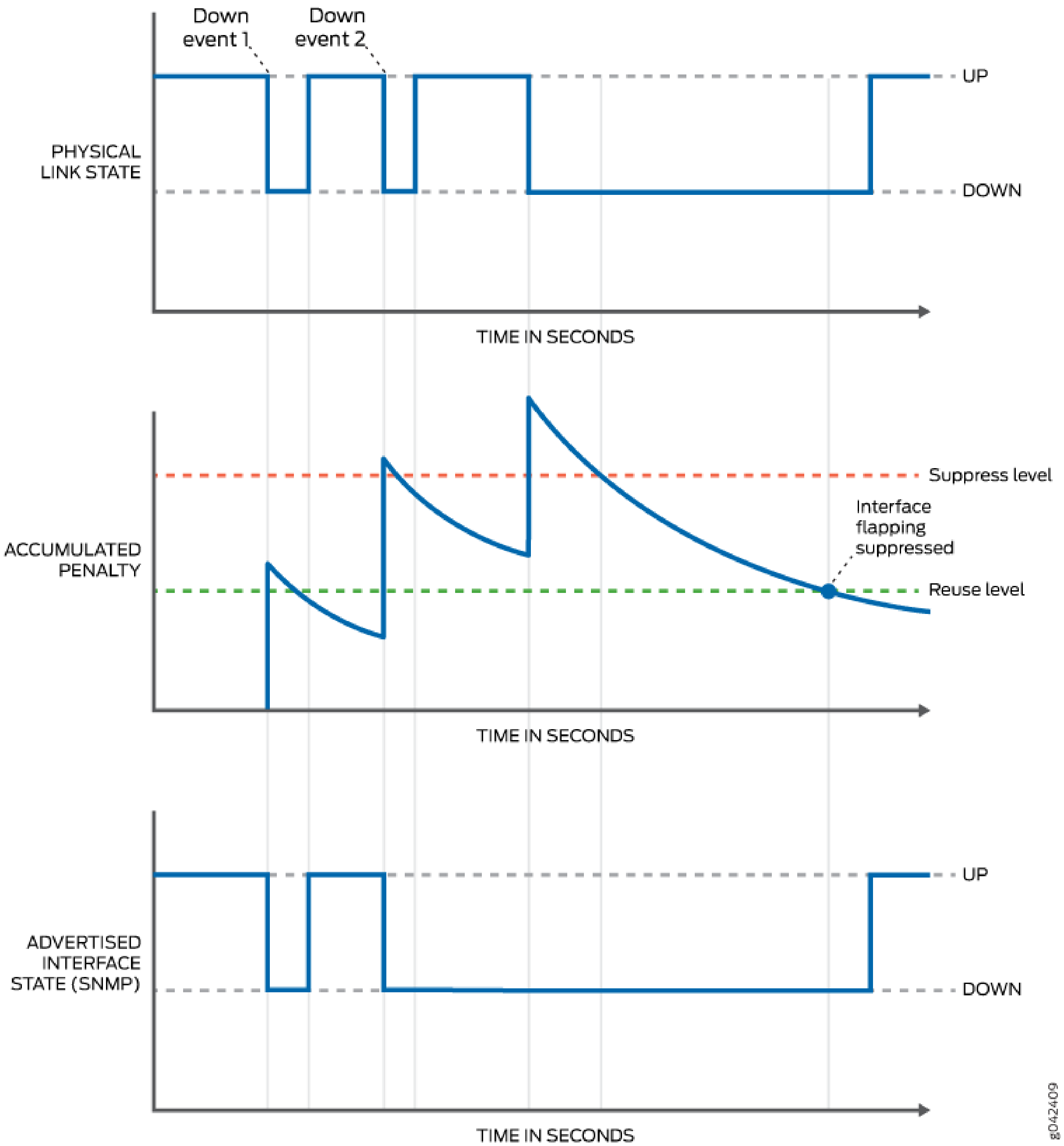
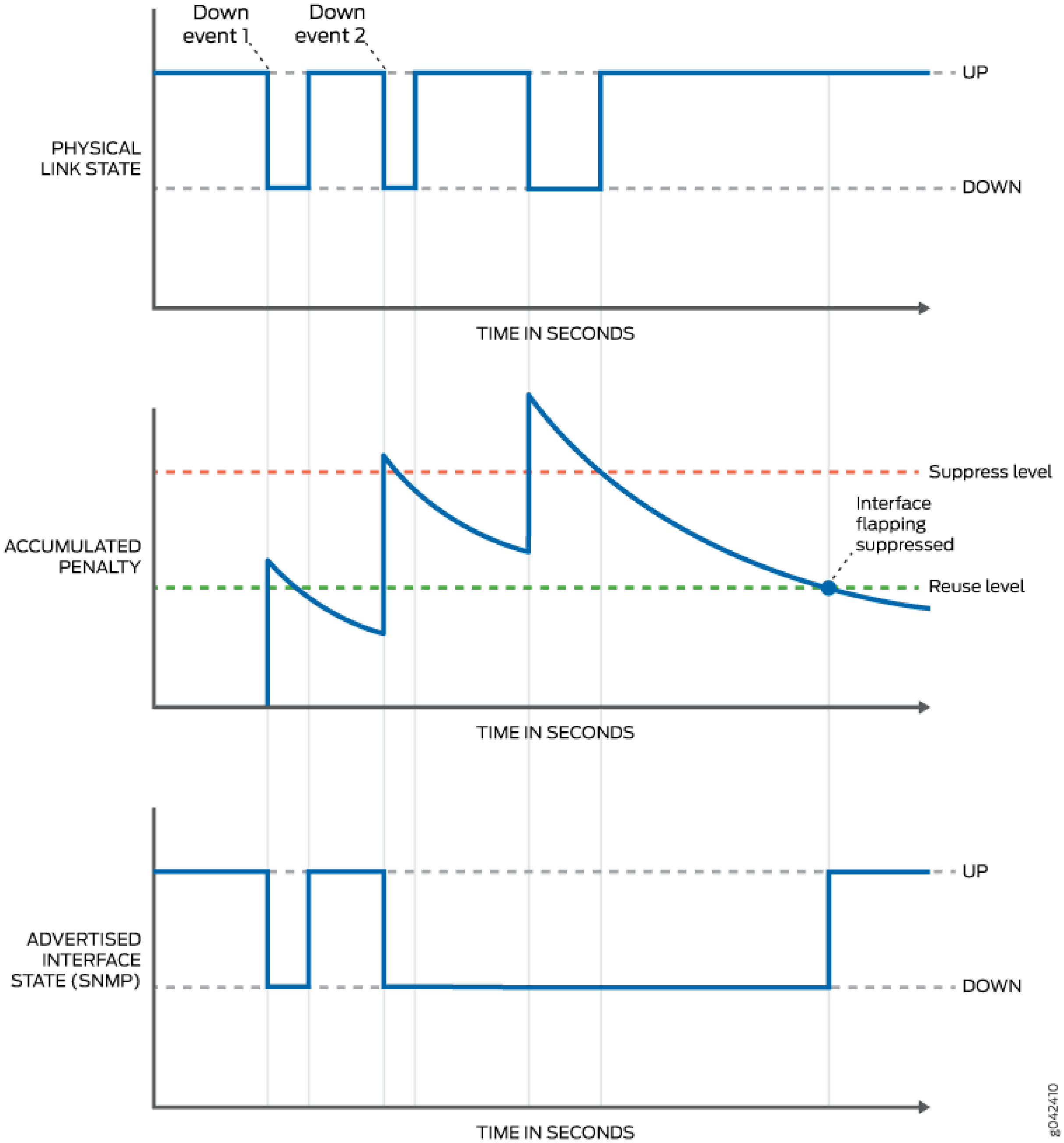


Figure 10 on page 136 shows the penalty dropping below the reuse level when the physical link is up. The system is notified of a state change immediately.

Figure 10: Physical-Level Link Is Up When the Penalty Falls Below the Reuse Level



SEE ALSO

Understanding Damping Parameters

damping | 305

hold-time

Damping Shorter Physical Interface Transitions

By default, when an interface changes from being up to being down, or from down to up, this transition is advertised immediately to the hardware and Junos OS. In some situations—for example, when an interface is connected to an add/drop multiplexer (ADM) or wavelength-division multiplexer (WDM), or to protect against SONET/SDH framer holes—you might want to damp interface transitions. This means not advertising the interface's transition until a certain period of time has passed, called the *hold-time*. When you have damped interface transitions and the interface goes from up to down, the down hold-time timer is triggered. Every interface transition that occurs during the hold-time is ignored. When the timer expires and the interface state is still *down*, then the router begins to advertise the interface as being down. Similarly, when an interface goes from down to up, the up hold-time timer is triggered. Every interface transition that occurs during the hold-time is ignored. When the timer expires and the interface state is still *up*, then the router begins to advertise the interface as being up. For information about physical interface damping, see "Physical Interface Damping Overview" on page 130.

This task applies to damping shorter physical interface transitions in milliseconds. To damp longer physical interface transitions in seconds, see "Damping Longer Physical Interface Transitions" on page 139.

To configure damping of shorter physical interface transitions:

1. Select the interface to damp, where the interface name is ***interface-type-fpc/pic/port***:

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

2. Configure the hold-time for link up and link down.

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

The hold time can be a value from 0 through 4,294,967,295 milliseconds. The default value is 0, which means that interface transitions are not damped. Junos OS advertises the transition within 100 milliseconds of the time value you specify.

For most Ethernet interfaces, hold timers are implemented using a one-second polling algorithm. For 1-port, 2-port, and 4-port Gigabit Ethernet interfaces with small form-factor pluggable transceivers (SFPs), hold timers are interrupt-driven.

NOTE: The **hold-time** option is not available for controller interfaces.

SEE ALSO

SONET/SDH Defect Hold Times for Damping Interface Transitions Overview

Configuring SONET/SDH Defect Triggers

hold-time

Damping Longer Physical Interface Transitions

Physical interface damping limits the advertisement of the up and down transitions (flapping) on an interface. An unstable link between a router Interface and the transport devices can lead to periodic flapping. Longer flaps occur with a period of about five seconds or more, with an up-and-down duration of one second. For these longer periodic interface flaps, you configure interface damping with the **damping** statement on the interface. This damping method uses an exponential back-off algorithm to suppress interface up and down event reporting to the upper-level protocols. Every time an interface goes down, a penalty is added to the interface penalty counter. If at some point the accumulated penalty exceeds the suppress level **max-suppress**, the interface is placed in the suppress state, and further interface state up and down transitions are not reported to the upper-level protocols.

> **NOTE:**
> - Only PTX Series routers, T Series routers, MX2010 routers, MX2020 routers, MX960 routers, MX480 routers, MX240 routers, MX80 routers, and M10i routers support interface damping for longer periodic interface flaps.
> - The system does not indicate whether an interface is down because of suppression or that is the actual state of the physical interface. Because of this, SNMP link traps and Operation, Administration, and Maintenance (OAM) protocols cannot differentiate the damped version of the link state from the real version. Therefore, the traps and protocols might not work as expected.
> - You can verify suppression by viewing the information in the **Damping** field of the **show interface extensive** command output.

You can view the damping parameters with the **show interfaces extensive** command.

To configure damping of longer physical interface transitions:

1. Select the interface to damp, where the interface name is ***interface-type-fpc/pic/port*** or an interface range:

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

2. Enable longer interface transition damping on a physical interface:

   ```
   [edit interfaces interface-name damping]
   user@host# set enable
   ```

3. (Optional) Set the maximum time in seconds that an interface can be suppressed no matter how unstable the interface has been.

   NOTE: Configure **max-suppress** to a value that is greater than the value of **half-life**; otherwise, the configuration is rejected.

   ```
   [edit interfaces interface-name damping]
   user@host# set max-suppress maximum-seconds
   ```

4. (Optional) Set the decay half-life in seconds, which is the interval after which the accumulated interface penalty counter is reduced by half if the interface remains stable.

   NOTE: Configure **max-suppress** to a value that is greater than the value of **half-life**; otherwise, the configuration is rejected.

   ```
   [edit interfaces interface-name damping]
   user@host# set half-life seconds
   ```

5. (Optional) Set the reuse threshold (no units). When the accumulated interface penalty counter falls below this value, the interface is no longer suppressed.

   ```
   [edit interfaces interface-name damping]
   user@host# set reuse number
   ```

6. (Optional) Set the suppression threshold (no units). When the accumulated interface penalty counter exceeds this value, the interface is suppressed.

   ```
   [edit interfaces interface-name damping]
   user@host# set suppress number
   ```

SEE ALSO

show interfaces extensive

Example: Configuring Physical Interface Damping

IN THIS SECTION

This example shows how to configure damping for a physical interface on a PTX Series Packet Transport Router.

Requirements

This example uses the following hardware and software components:

- One PTX Series Packet Transport Router
- One or more routers that provide input packets and receive output packets
- Junos OS Release 14.1 or later

Overview

Physical interface damping provides a smoothing of the up and down transitions (flapping) on an interface. Each time a transition occurs, the interface state is changed, which generates an advertisement to the upper-level routing protocols. Damping helps reduce the number of these advertisements.

From the viewpoint of network deployment, physical interface flaps fall into these categories:

- Nearly instantaneous multiple flaps of short duration (milliseconds). For shorter physical interface transitions, you configure interface damping with the **hold-time** statement on the interface. The hold timer enables interface damping by not advertising interface transitions until the hold timer duration has passed. When a hold-down timer is configured and the interface goes from up to down, the interface is not advertised to the rest of the system as being down until it has remained down for the hold-down timer period. Similarly, when a hold-up timer is configured and an interface goes from down to up, it is not advertised as being up until it has remained up for the hold-up timer period.

- Periodic flaps of long duration (seconds). For longer periodic interface flaps, you configure interface damping with the **damping** statement on the interface. This damping method uses an exponential back-off algorithm to suppress interface up and down event reporting to the upper-level protocols. Every time an interface goes down, a penalty is added to the interface penalty counter. If at some point the accumulated penalty exceeds the suppress level, the interface is placed in the suppress state, and further interface state up transitions are not reported to the upper-level protocols.

Configuration

CLI Quick Configuration

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

```
set interfaces xe-6/0/0 damping half-life 11 max-suppress 2222 reuse 3333 suppress 4444 enable
```

Step-by-Step Procedure

To configure damping on the PTX Series Packet Transport Router:

1. Set the half-life interval, maximum suppression, reuse, suppress values, and enable:

   ```
   [edit interface]
   user@router# set xe-6/0/0 damping half-life 11 max-suppress 2222 reuse 3333 suppress 4444 enable
   ```

2. Commit configuration:

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

Results

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

```
user@router# show interfaces
xe-6/0/0 {
  damping {
```

```
        half-life 11;
        max-suppress 2222;
        reuse 3333;
        suppress 4444;
        enable;
    }
```

Verification

IN THIS SECTION

To confirm that the configuration is working properly, perform this task:

Verifying Interface Damping on xe-6/0/0

Purpose

Verify that damping is enabled on the interface and that the damping parameter values are correctly set.

Action

From operational mode, run the **show interfaces extensive** command.

```
user@router# run show interfaces xe-6/0/0 extensive
```

```
Physical interface: xe-6/0/0, Enabled, Physical link is Up
  Interface index: 158, SNMP ifIndex: 535, Generation: 161
  Link-level type: Ethernet, MTU: 1514, LAN-PHY mode, Speed: 10Gbps, BPDU Error:
None, Loopback: None,
  Source filtering: Disabled, Flow control: Enabled
  Device flags   : Present Running
  Interface flags: SNMP-Traps Internal: 0x4000
  Link flags     : None
  CoS queues     : 8 supported, 8 maximum usable queues
  Hold-times     : Up 0 ms, Down 0 ms
  Damping       : half-life: 11 sec, max-suppress: 2222 sec, reuse: 3333, suppress:
 4444, state: unsuppressed
```

Meaning

Damping is enabled and configured successfully on the xe-6/0/0 interface.

SEE ALSO

damping | 305

Enabling or Disabling SNMP Notifications on Physical Interfaces

By default, Simple Network Management Protocol (SNMP) notifications are sent when the state of an interface or a connection changes. You can enable or disable these notification based on you requirements.

To explicitly enable sending SNMP notifications on the physical interface, perform the following steps:

1. In configuration mode, go to the **[edit interfaces *interface-name*]** hierarchy level:

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

2. Configure the **traps** option to enable sending of Simple Network Management Protocol (SNMP) notifications when the state of the connection changes.

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

To disable sending SNMP notifications on the physical interface, perform the following steps:

1. In configuration mode, go to the **[edit interfaces *interface-name*]** hierarchy level:

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

2. Configure the **no-traps** option to disable sending of Simple Network Management Protocol (SNMP) notifications when the state of the connection changes.

   ```
   [edit interfaces interface-name]
   user@host# set no-traps
   ```

SEE ALSO

traps

Configuring Accounting for the Physical Interface

IN THIS SECTION

Accounting Profiles Overview

Juniper Networks routers and switches can collect various kinds of data about traffic passing through the router and switch. You can set up one or more *accounting profiles* that specify some common characteristics of this data, including the following:

- The fields used in the accounting records
- The number of files that the router or switch retains before discarding, and the number of bytes per file
- The polling period that the system uses to record the data

You configure the profiles and define a unique name for each profile using statements at the **[edit accounting-options]** hierarchy level. There are two types of accounting profiles: interface profiles and filter profiles. You configure interface profiles by including the **interface-profile** statement at the **[edit accounting-options]** hierarchy level. You configure filter profiles by including the **filter-profile** statement at the **[edit accounting-options]** hierarchy level. For more information, see the *Network Management and Monitoring Guide*.

You apply filter profiles by including the **accounting-profile** statement at the **[edit firewall filter *filter-name*]** and **[edit firewall family *family* filter *filter-name*]** hierarchy levels. For more information, see the *Routing Policies, Firewall Filters, and Traffic Policers User Guide*.

Configuring Accounting for the Physical Interface

Before you begin

You must configure a profile to collect error and statistic information for input and output packets on a particular physical interface. An accounting profile specifies what statistics should be collected and written to a log file. For more information on how to configure an accounting-data log file, see the *Configuring Accounting-Data Log Files*.

An interface profile specifies the information collected and written to a log file. You can configure a profile to collect error and statistic information for input and output packets on a particular physical interface.

1. To configure which statistics should be collected for an interface, include the **fields** statement at the **[edit accounting-options interface-profile *profile-name*]** hierarchy level.

   ```
   [edit accounting-options interface-profile profile-name]
   user@host# set fieldsfield-name
   ```

2. Each accounting profile logs its statistics to a file in the **/var/log** directory. To configure which file to use, include the **file** statement at the **[edit accounting-options interface-profile *profile-name*]** hierarchy level.

   ```
   [edit accounting-options interface-profile profile-name]
   user@host# set file filename
   ```

   NOTE: You must specify a **file** statement for the interface profile that has already been configured at the **[edit accounting-options]** hierarchy level. For more information, see the Configuring Accounting-Data Log Files

3. Each interface with an accounting profile enabled has statistics collected once per interval time specified for the accounting profile. Statistics collection time is scheduled evenly over the configured interval. To configure the interval, include the interval statement at the [**edit accounting-options interface-profile *profile-name***] hierarchy level.

   ```
   [edit accounting-options interface-profile profile-name]
   user@host# set interval minutes
   ```

   NOTE: The minimum interval allowed is 1 minute. Configuring a low interval in an accounting profile for a large number of interfaces might cause serious performance degradation.

4. To configure the interfaces on which the accounting needs to be performed, apply the interface profile to a physical interface by including the **accounting-profile** statement at the **[edit interfaces interface-name]** hierarchy level.

```
[edit interfaces]
user@host# set interface-name accounting-profile profile-name
```

SEE ALSO

Configuring Accounting-Data Log Files

Displaying Accounting Profile for the Physical Interface

Purpose

To display the configured accounting profile a particular physical interface at the **[edit accounting-options interface-profile *profile-name*]** hierarchy level:

- interface-name—ge-1/0/1
- Interface profile —**if_profile**
- File name—**if_stats**
- Interval—15 minutes

Action

- Run the **show** command at the **[edit edit interfaces ge-1/0/1]** hierarchy level.

```
[edit interfaces ge-1/0/1]
accounting-profile if_profile;
```

- Run the **show** command at the **[edit accounting-options]** hierarchy level.

```
interface-profile if_profile {
    interval 15;
    file if_stats {
        fields {
            input-bytes;
            output-bytes;
            input-packets;
            output-packets;
            input-errors;
            output-errors;
        }
    }
}
```

Meaning

The configured accounting and its associated set options are displayed as expected.

Disabling a Physical Interface

IN THIS SECTION

Disabling a Physical Interface

You can disable a physical interface, marking it as being down, without removing the interface configuration statements from the configuration.

CAUTION: Dynamic subscribers and logical interfaces use physical interfaces for connection to the network. The Junos OS allows you to set the interface to disable and commit the change while dynamic subscribers and logical interfaces are still active. This action results in the loss of all subscriber connections on the interface. Use care when disabling interfaces.

To disable a physical interface:

1. In configuration mode, go to **[edit interfaces *interface-name*]** hierarchy level.

   ```
   [edit]
   user@host# edit interfaces ge-fpc/pic/port
   ```

2. Include the **disable** statement.

   ```
   [edit interfaces at-fpc/pic/port ]
   user@host# set disable
   ```

NOTE: On the router, when you use the **disable** statement at the **edit interfaces** hierarchy level, depending on the PIC type, the interface might or might not turn off the laser. Older PIC transceivers do not support turning off the laser, but newer Gigabit Ethernet PICs with SFP and XFP transceivers do support it and the laser will be turned off when the interface is disabled.

WARNING: Do not stare into the laser beam or view it directly with optical instruments even if the interface has been disabled.

Example: Disabling a Physical Interface

Sample interface configuration:

```
[edit interfaces]
user@host# show
ge-0/3/2 {
    unit 0 {
        description CE2-to-PE1;
        family inet {
            address 20.1.1.6/24;
        }
    }
}
```

Disabling the interface:

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

Verifying the interface configuration:

```
[edit interfaces ge-0/3/2]
user@host# show
disable; # Interface is marked as disabled.
    unit 0 {
    description CE2-to-PE1;
        family inet {
            address 20.1.1.6/24;
        }
```

```
}
```

Effect of Disabling Interfaces on T series PICs

The following table describes the effect of using the **set interfaces disable *interface_name*** statement on T series PICs.

Table 20: Effect of set interfaces disable <interface_name> on T series PICs

PIC Model Number	PIC Description	Type of PIC	Behaviour
PF-12XGE-SFPP	10-Gigabit Ethernet LAN/WAN PIC with SFP+ (T4000 Router)	5	Tx laser disabled
PF-24XGE-SFPP	10-Gigabit Ethernet LAN/WAN PIC with Oversubscription and SFP+ (T4000 Router)	5	Tx laser disabled
PF-1CGE-CFP	100-Gigabit Ethernet PIC with CFP (T4000 Router)	5	Tx laser disabled
PD-4XGE-XFP	10-Gigabit Ethernet, 4-port LAN/WAN XFP	4	Tx laser disabled
PD-5-10XGE-SFPP	10-Gigabit LAN/WAN with SFP+	4	Tx laser disabled
PD-1XLE-CFP	40-Gigabit with CFP	4	Tx laser disabled
PD-1CE-CFP-FPC4	100-Gigabit with CFP	4	Tx laser disabled
PD-TUNNEL	40-Gigabit Tunnel Services	4	NA
PD-4OC192-SON-XFP	OC192/STM64, 4-port XFP	4	Tx laser not disabled
PD-1OC768-SON-SR	OC768c/STM256, 1-port	4	Tx laser not disabled

RELATED DOCUMENTATION

disable

Logical Interface Properties

IN THIS SECTION

This topic discusses how to configure various logical interface properties with examples.

Logical Interface Properties Overview

For a physical interface device to function, you must configure at least one logical interface on that device. For each logical interface, you must specify the protocol family that the interface supports. You can also configure other logical interface properties. These vary by Physical Interface Card (PIC) and encapsulation type, but include the IP address of the interface, and whether the interface supports multicast traffic, data-link connection identifiers (DLCIs), virtual channel identifiers (VCIs) and virtual path identifiers (VPIs), and traffic shaping.

To configure logical interface properties, include the statements at the following hierarchy levels:

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

SEE ALSO

Logical Part of an Interface Name | 34

Specifying the Logical Interface Number

Each logical interface must have a logical unit number. The logical unit number corresponds to the logical unit part of the interface name. For more information, see "Interface Naming Overview" on page 26.

Point-to-Point Protocol (PPP), Cisco High-level Data Link Control (HDLC), and Ethernet circuit cross-connect (CCC) encapsulations support only a single logical interface, whose logical unit number must be 0. Frame Relay and ATM encapsulations support multiple logical interfaces, so you can configure one or more logical unit numbers.

You specify the logical unit number by including the **unit** statement:

```
unit logical-unit-number {
    ...
}
```

You can include this statement at the following hierarchy levels:

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

The range of number available for the logical unit number varies for different interface types. See *unit* for current range values.

Adding a Logical Unit Description to the Configuration

You can include a text description of each logical unit in the configuration file. Any descriptive text you include is displayed in the output of the **show interfaces** commands, and is also exposed in the **ifAlias** Management Information Base (MIB) object. It has no impact on the interface's configuration. To add a text description, include the **description** statement:

```
description text;
```

You can include this statement at the following hierarchy levels:

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

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

The description can be a single line of text. If the text contains spaces, enclose it in quotation marks.

> **NOTE:** You can configure the extended DHCP relay to include the interface description in the option 82 Agent Circuit ID suboption. See "*Using DHCP Relay Agent Option 82 Information*" in the *Junos OS Broadband Subscriber Management and Services Library*.

For information about describing physical interfaces, see "Configuring Interface Description" on page 89.

Configuring the Interface Bandwidth

By default, the Junos OS uses the physical interface's speed for the MIB-II object, **ifSpeed**. You can configure the logical unit to populate the **ifSpeed** variable by configuring a bandwidth value for the logical interface. The **bandwidth** statement sets an informational-only parameter; you cannot adjust the actual bandwidth of an interface with this statement.

> **NOTE:** We recommend that you be careful when setting this value. Any interface bandwidth value that you configure using the **bandwidth** statement affects how the interface cost is calculated for a dynamic routing protocol, such as OSPF. By default, the interface cost for a dynamic routing protocol is calculated using the following formula:
>
> ```
> cost = reference-bandwidth/bandwidth,
> ```
>
> where bandwidth is the physical interface speed. However, if you specify a value for bandwidth using the **bandwidth** statement, that value is used to calculate the interface cost, rather than the actual physical interface bandwidth.

To configure the bandwidth value for a logical interface, include the **bandwidth** statement:

```
bandwidth rate;
```

You can include this statement at the following hierarchy levels:

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

rate is the peak rate, in bps or cps. You can specify a value in bits per second either as a complete decimal number or as a decimal number followed by the abbreviation **k** (1000), **m** (1,000,000), or **g** (1,000,000,000). You can also specify a value in cells per second by entering a decimal number followed by the abbreviation **c**; values expressed in cells per second are converted to bits per second using the formula 1 cps = 384 bps. The value can be any positive integer. The **bandwidth** statement is valid for all logical interfaces, except multilink interfaces.

Configuring Interface Encapsulation on Logical Interfaces

IN THIS SECTION

Understanding Interface Encapsulation on Logical Interfaces

You can configure an encapsulation on a logical interface, which is the encapsulation used within certain packet types.

The following restrictions apply to logical interface encapsulation:

- With the atm-nlpid, atm-cisco-nlpid, and atm-vc-mux encapsulations, you can configure the inet family only.
- With the CCC circuit encapsulations, you cannot configure a family on the logical interface.
- A logical interface cannot have frame-relay-ccc encapsulation unless the physical device also has frame-relay-ccc encapsulation.
- A logical interface cannot have frame-relay-tcc encapsulation unless the physical device also has frame-relay-tcc encapsulation. In addition, you must assign this logical interface a DLCI from 512 through 1022 and configure it as point-to-point.
- A logical interface cannot have frame-relay-ether-type or frame-relay-ether-type-tcc encapsulation unless the physical interface has flexible-frame-relay encapsulation and is on an IQ or IQE PIC.
- For frame-relay-ether-type-tcc encapsulation, you must assign this logical interface a DLCI from 512 through 1022.
- For interfaces that carry IP version 6 (IPv6) traffic, you cannot configure ether-over-atm-llc encapsulation.

- When you use ether-over-atm-llc encapsulation, you cannot configure multipoint interfaces.
- A logical interface cannot have vlan-ccc or vlan-vpls encapsulation unless the physical device also has vlan-ccc or vlan-vpls encapsulation, respectively. In addition, you must assign this logical interface a VLAN ID from 512 through 1023; if the VLAN ID is 511 or lower, it is subject to the normal destination filter lookups in addition to source address filtering. For more information, see *Configuring VLAN and Extended VLAN Encapsulation*.
- You can create an ATM cell-relay circuit by configuring an entire ATM physical device or an individual virtual circuit (VC). When you configure an entire device, only cell-relay encapsulation is allowed on the logical interfaces. For more information, see *Configuring an ATM1 Cell-Relay Circuit Overview*.

Configuring the Encapsulation on a Logical Interface

Generally, you configure an interface's encapsulation at the **[edit interfaces *interface-name*]** hierarchy level. However, for some encapsulation types, such as Frame Relay, ATM, and Ethernet virtual local area network (VLAN) encapsulations, you can also configure the encapsulation type that is used inside the Frame Relay, ATM, or VLAN circuit itself.

To configure encapsulation on a logical interface:

1. In configuration mode, go to the **[edit interfaces *interface-name* unit *logical-unit-number*]** or **[edit logical-systems *logical-system-name* interfaces *interface-name* unit *logical-unit-number*]** hierarchy level.

```
[edit]
user@host# set interfaces at-fpc/pic/port unit logical-unit-number
```

2. Configure the encapsulation type as described in *encapsulation (Logical Interface)*.

```
[edit interfaces at-fpc/pic/port unit logical-unit-number]
user@host# set encapsulation encapsulation-type
```

Displaying the Encapsulation on a Logical Interface

Purpose

To display the configured encapsulation and its associated set options on a physical interface when the following are set at the **[edit interfaces *interface-name*]** or **[edit logical-systems *logical-system-name* interfaces *interface-name*]** hierarchy level:

- interface-name—at-1/1/0
- Encapsulation—**atm-ccc-cell-relay**
- Unit—120

Action

Run the **show** command at the **[edit interfaces *interface-name*]** hierarchy level.

```
[edit interfaces at-1/1/0]
user@host# show
encapsulation atm-ccc-cell-relay;
unit 120 {
   encapsulation atm-ccc-cell-relay;
}
```

Meaning

The configured encapsulation and its associated set options are displayed as expected.

RELATED DOCUMENTATION

encapsulation (Logical Interface)

Configuring VLAN and Extended VLAN Encapsulation

Configuring an ATM1 Cell-Relay Circuit Overview

Configuring Interface Encapsulation on PTX Series Packet Transport Routers

This topic describes how to configure interface encapsulation on PTX Series Packet Transport Routers. Use the **flexible-ethernet-services** configuration statement to configure different encapsulation for different logical interfaces under a physical interface. With flexible Ethernet services encapsulation, you can configure each logical interface encapsulation without range restrictions for VLAN IDs.

Supported encapsulations for physical interfaces include:

- **flexible-ethernet-services**
- **ethernet-ccc**
- **ethernet-tcc**

Supported encapsulations for logical interfaces include:

- **ethernet**
- **vlan-ccc**
- **vlan-tcc**

NOTE: PTX Series Packet Transport Routers do not support **extended-vlan-cc** and **extended-vlan-tcc** encapsulation on logical interfaces. Instead, you can configure a tag protocol ID (TPID) value of 0x9100 to achieve the same results.

To configure flexible Ethernet services encapsulation, include the **encapsulation flexible-ethernet-services** statement at the **[edit interfaces et-*fpc/pic/port*]** hierarchy level. For example:

```
interfaces {
  et-fpc/pic/port {
    vlan-tagging;
    encapsulation flexible-ethernet-services;
    unit 0 {
      vlan-id 1000;
      family inet {
        address 11.0.0.20/24;
      }
    }
    unit 1 {
      encapsulation vlan-ccc;
      vlan-id 1010;
    }
    unit 2 {
      encapsulation vlan-tcc;
      vlan-id 1020;
      family tcc {
        proxy {
          inet-address 11.0.2.160;
        }
        remote {
          inet-address 11.0.2.10;
        }
      }
    }
  }
}
```

Configuring a Point-to-Point Connection

By default, all interfaces are assumed to be point-to-point connections. You must ensure that the maximum transmission unit (MTU) sizes on both sides of the connection are the same.

For all interfaces except aggregated Ethernet, Fast Ethernet, and Gigabit Ethernet, you can explicitly configure an interface to be a point-to-point connection by including the **point-to-point** statement:

```
point-to-point;
```

You can include this statement at the following hierarchy levels:

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

Configuring a Multipoint Connection

By default, all interfaces are assumed to be point-to-point connections. To configure an interface to be a multipoint connection, include the **multipoint** statement:

```
multipoint;
```

You can include this statement at the following hierarchy levels:

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

Configuring Dynamic Profiles for PPP

A dynamic profile acts as a template that enables you to create, update, or remove a configuration that includes attributes for client access (for example, interface or protocol) or service (for example, IGMP). Using these profiles you can consolidate all of the common attributes of a client (and eventually a group of clients) and apply the attributes simultaneously.

After they are created, the profiles reside in a profile library on the router. You can then use the **dynamic-profile** statement to attach profiles to interfaces. To assign a dynamic profile to a PPP interface, you can include the **dynamic-profile** statement at the **[edit interfaces *interface-name* unit *logical-unit-number* ppp-options]** hierarchy level:

```
[edit interfaces interface-name unit logical-unit-number ppp-options]
dynamic-profile profile-name;
```

To monitor the configuration, issue the **show interfaces *interface-name*** command.

For information about dynamic profiles, see *Dynamic Profiles Overview* in the *Junos Subscriber Access Configuration Guide*.

For information about creating dynamic profiles, see *Configuring a Basic Dynamic Profile* in the *Junos Subscriber Access Configuration Guide*.

For information about assigning a dynamic profile to a PPP interface, see *Attaching Dynamic Profiles to Static PPP Subscriber Interfaces* in the *Junos Subscriber Access Configuration Guide*.

For information about using dynamic profiles to authenticate PPP subscribers, see *Configuring Dynamic Authentication for PPP Subscribers*.

> **NOTE:** Dynamic profiles for PPP subscribers are supported only on PPPoE interfaces for this release.

Configuring Accounting for the Logical Interface

IN THIS SECTION

Accounting Profiles Overview

Juniper Networks routers and switches can collect various kinds of data about traffic passing through the router and switch. You can set up one or more *accounting profiles* that specify some common characteristics of this data, including the following:

- The fields used in the accounting records
- The number of files that the router or switch retains before discarding, and the number of bytes per file
- The polling period that the system uses to record the data

You configure the profiles and define a unique name for each profile using statements at the **[edit accounting-options]** hierarchy level. There are two types of accounting profiles: interface profiles and filter profiles. You configure interface profiles by including the **interface-profile** statement at the **[edit**

accounting-options] hierarchy level. You configure filter profiles by including the **filter-profile** statement at the **[edit accounting-options]** hierarchy level. For more information, see the *Network Management and Monitoring Guide*.

You apply filter profiles by including the **accounting-profile** statement at the **[edit firewall filter *filter-name*]** and **[edit firewall family *family* filter *filter-name*]** hierarchy levels. For more information, see the *Routing Policies, Firewall Filters, and Traffic Policers User Guide*.

Configuring Accounting for the Logical Interface

Before you begin

You must configure a profile to collect error and statistic information for input and output packets on a particular logical interface. An accounting profile specifies what statistics should be collected and written to a log file. For more information on how to configure an accounting-data log file, see the *Configuring Accounting-Data Log Files*.

An interface profile specifies the information collected and written to a log file. You can configure a profile to collect error and statistic information for input and output packets on a particular logical interface.

1. To configure which statistics should be collected for an interface, include the **fields** statement at the **[edit accounting-options interface-profile *profile-name*]** hierarchy level.

   ```
   [edit accounting-options interface-profile profile-name]
   user@host# set fieldsfield-name
   ```

2. Each accounting profile logs its statistics to a file in the **/var/log** directory. To configure which file to use, include the **file** statement at the **[edit accounting-options interface-profile *profile-name*]** hierarchy level.

   ```
   [edit accounting-options interface-profile profile-name]
   user@host# set file filename
   ```

   > **NOTE:** You must specify a **file** statement for the interface profile that has already been configured at the **[edit accounting-options]** hierarchy level. For more information, see the Configuring Accounting-Data Log Files

3. Each interface with an accounting profile enabled has statistics collected once per interval time specified for the accounting profile. Statistics collection time is scheduled evenly over the configured interval. To configure the interval, include the interval statement at the [**edit accounting-options interface-profile *profile-name***] hierarchy level.

```
[edit accounting-options interface-profile profile-name]
user@host# set interval minutes
```

NOTE: The minimum interval allowed is 1 minute. Configuring a low interval in an accounting profile for a large number of interfaces might cause serious performance degradation.

4. To configure the interfaces on which the accounting needs to be performed, apply the interface profile to a logial interface by including the **accounting-profile** statement at the **[edit interfaces interface-name unit *logical-unit-number*]** hierarchy level.

```
[edit interfaces]
user@host# set interface-name unit logical-unit-numberaccounting-profile profile-name
```

SEE ALSO

Accounting Options Overview

Configuring Accounting-Data Log Files

Displaying Accounting Profile for the Logical Interface

Purpose

To display the configured accounting profile a particular logical interface at the **[edit accounting-options interface-profile *profile-name*]** hierarchy level:

- interface-name—ge-1/0/1
- Logical unit number—1
- Interface profile —**if_profile**
- File name—**if_stats**
- Interval—15 minutes

Action

- Run the **show** command at the **[edit interfaces ge-1/0/1 unit 1]** hierarchy level.

```
[edit interfaces ge-1/0/1 unit 1]
accounting-profile if_profile;
```

- Run the **show** command at the **[edit accounting-options]** hierarchy level.

```
interface-profile if_profile {
    interval 15;
    file if_stats {
        fields {
            input-bytes;
            output-bytes;
            input-packets;
            output-packets;
            input-errors;
            output-errors;
        }
    }
}
```

Meaning

The configured accounting and its associated set options are displayed as expected.

Enabling or Disabling SNMP Notifications on Logical Interfaces

By default, Simple Network Management Protocol (SNMP) notifications are sent when the state of an interface or a connection changes. To explicitly enable these notifications on the logical interface, include the **traps** statement; to disable these notifications on the logical interface, include the **no-traps** statement:

```
(traps | no-traps);
```

You can include these statements at the following hierarchy levels:

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

Disabling a Logical Interface

You can unconfigure a logical interface, effectively disabling that interface, without removing the logical interface configuration statements from the configuration. To do this, include the **disable** statement:

```
disable;
```

You can include this statement at the following hierarchy levels:

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

When an interface is disabled, a route (pointing to the reserved target "**REJECT**") with the IP address of the interface and a 32–bit subnet mask is installed in the routing table. See *Routing Protocols*.

Example: Disabling a Logical Interface

Sample interface configuration:

```
[edit interfaces]
user@host# show
et-2/1/1 {
   vlan-tagging;
   encapsulation flexible-ethernet-services;
   unit 0 {
      vlan-id 1000;
      family inet {
         address 11.0.0.20/24;
      }
   }
}
```

Disabling the interface:

```
[edit interfaces et-2/1/1 unit 0]
user@host# set disable
```

Verifying the interface configuration:

```
[edit interfaces et-2/1/1]
user@host# show
disable; # Interface is marked as disabled.
   unit 0 {
   vlan-id 1000;
```

```
        family inet {
            address 11.0.0.20/24;
        }
}
```

Protocol Family and Interface Address Properties

IN THIS SECTION

This section discusses on how to configure protocol family and interface address properties.

Configuring the Protocol Family

A protocol family is a group of logical properties within an interface configuration. Protocol families include all the protocols that make up a protocol suite. To use a protocol within a particular suite, you must configure the entire protocol family as a logical property for an interface.

Junos OS protocol families include the following common protocol suites:

- Inet—Supports IP protocol traffic, including OSPF, BGP, and Internet Control Message Protocol (ICMP).
- Inet6—Supports IPv6 protocol traffic, including RIP for IPv6 (RIPng), IS-IS, and BGP.
- ISO—Supports IS-IS traffic.
- MPLS—Supports MPLS.

In addition to the common protocol suites, JUNOS protocol families sometimes use the following protocol suites. For more information see, *family*.

To configure the logical interface's protocol family, include the **family** statement, specifying the selected family. To configure the protocol family, following are the minimum configuration tasks under the **[edit interfaces interface-name unit logical-unit-number family family]** hierarchy.

Table 21: Protocol Family Configuration Tasks

Task	Find Details Here
Configure MTU	"Configuring the Media MTU" on page 84
Configure the unit and family so that the interface can transmit and receive multicast traffic only	*Restricting Tunnels to Multicast Traffic*
Disable the sending of redirect messages by the router	*Protocol Redirect Messages*
Assign an address to an interface	"Configuring the Interface Address" on page 166

SEE ALSO

family

Configuring the Interface Address

You assign an address to an interface by specifying the address when configuring the protocol family. For the **inet** or **inet6** family, configure the interface IP address. For the **iso** family, configure one or more addresses for the loopback interface. For the **ccc**, **ethernet-switching**, **tcc**, **mpls**, **tnp**, and **vpls** families, you never configure an address.

> **NOTE:** The point-to-point (PPP) address is taken from the loopback interface address that has the primary attribute. When the loopback interface is configured as an unnumbered interface, it takes the primary address from the donor interface.

To assign an address to an interface, perform the following steps:

1. Configure the interface address at the **[edit interfaces *interface-name* unit *logical-unit-number* family *family*]** hierarchy level.

   - To configure an IPv4 address on routers and switches running Junos OS, use the **interface *interface-name* unit *number* family inet address *a.b.c.d/nn*** statement at the **[edit interfaces]** hierarchy level.

     You can also assign multiple IPv4 addresses on the same interface.

     ```
     [edit interfaces ]
     user@host# set interface-name unit logical-unit-number family inet address a.b.c.d/nn
     ```

     NOTE:

     - Juniper Networks routers and switches support **/31** destination prefixes when used in point-to-point Ethernet configurations; however, they are not supported by many other devices, such as hosts, hubs, routers, or switches. You must determine if the peer system also supports **/31** destination prefixes before configuration.
     - You can configure the same IPv4 address on multiple physical interfaces. When you assign the same IPv4 address to multiple physical interfaces, the operational behavior of those interfaces differs, depending on whether they are implicitly or explicitly point-to-point .
     - By default, all interfaces are assumed to be point-to-point (PPP) interfaces. For all interfaces except aggregated Ethernet, Fast Ethernet, and Gigabit Ethernet, you can explicitly configure an interface to be a point-to-point connection.
     - If you configure the same IP address on multiple interfaces in the same routing instance, Junos OS applies the configuration randomly on one of the interfaces. The other interfaces will remain without an IP address.

   - To configure an IPv6 address on routers and switches running Junos OS, use the **interface *interface-name* unit *number* family inet6 address *aaaa:bbbb:...:zzzz/nn*** statement at the **[edit interfaces]** hierarchy level.

     ```
     [edit interfaces ]
     user@host# set interface-name unit logical-unit-number family inet6 address aaaa:bbbb:...:zzzz/nn
     ```

NOTE:

- You represent IP version 6 (IPv6) addresses in hexadecimal notation using a colon-separated list of 16-bit values. The double colon (::) represents all bits set to 0.
- You must manually configure the router or switch advertisement and advertise the default prefix for autoconfiguration to work on a specific interface.

2. [Optional] Set the broadcast address on the network or subnet .

   [edit interfaces *interface-name* unit *logical-unit-number* family *family* address *address*],
   user@host# **set broadcast *address***

   NOTE: The broadcast address must have a host portion of either all ones or all zeros. You cannot specify the addresses **0.0.0.0** or **255.255.255.255**

3. [Optional] specify the remote address of the connection for the encrypted, PPP-encapsulated, and tunnel interfaces.

   [edit logical-systems *logical-system-name* interfaces *interface-name* unit *logical-unit-number* family *family* address *address*]
   user@host# **set destination *address***

4. [Optional] For interfaces that carry IP version 6 (IPv6) traffic, configure the host to assign iteslf a unique 64-Bit IP Version 6 interface identifier (EUI-64).

   [edit logical-systems *logical-system-name* interfaces *interface-name* unit *logical-unit-number* family *family* address *address*]
   user@host# **set eui-64**

Configuring Default, Primary, and Preferred Addresses and Interfaces

IN THIS SECTION

Default, Primary, and Preferred Addresses and Interfaces

The router has a default address and a primary interface, and interfaces have primary and preferred addresses.

The *default address* of the router is used as the source address on unnumbered interfaces. The routing protocol process tries to pick the default address as the router ID, which is used by protocols, including OSPF and internal BGP (IBGP).

The *primary interface* for the router is the interface that packets go out when no interface name is specified and when the destination address does not imply a particular outgoing interface.

An interface's *primary address* is used by default as the local address for broadcast and multicast packets sourced locally and sent out the interface. An interface's *preferred address* is the default local address used for packets sourced by the local router to destinations on the subnet.

The default address of the router is chosen using the following sequence:

1. The primary address on the loopback interface **lo0** that is not **127.0.0.1** is used.
2. The primary address on the primary interface is used.

Configuring the Primary Interface for the Router

The *primary interface* for the router has the following characteristics:

- It is the interface that packets go out when you type a command such as ping 255.255.255.255—that is, a command that does not include an interface name (there is no interface ***type*-0/0/0.0** qualifier) and where the destination address does not imply any particular outgoing interface.
- It is the interface on which multicast applications running locally on the router, such as Session Announcement Protocol (SAP), do group joins by default.

- It is the interface from which the default local address is derived for packets sourced out an unnumbered interface if there are no non-127 addresses configured on the loopback interface, lo0.

By default, the multicast-capable interface with the lowest-index address is chosen as the primary interface. If there is no such interface, the point-to-point interface with the lowest index address is chosen. Otherwise, any interface with an address could be picked. In practice, this means that, on the router, the **fxp0** or **em0** interface is picked by default.

To configure a different interface to be the primary interface, include the **primary** statement:

```
primary;
```

You can include this statement at the following hierarchy levels:

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

Configuring the Primary Address for an Interface

The *primary address* on an interface is the address that is used by default as the local address for broadcast and multicast packets sourced locally and sent out the interface. For example, the local address in the packets sent by a **ping interface so-0/0/0.0 255.255.255.255** command is the primary address on interface **so-0/0/0.0**. The primary address flag also can be useful for selecting the local address used for packets sent out unnumbered interfaces when multiple non-127 addresses are configured on the loopback interface, **lo0**. By default, the primary address on an interface is selected as the numerically lowest local address configured on the interface.

To set a different primary address, include the **primary** statement:

```
primary;
```

You can include this statement at the following hierarchy levels:

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

Configuring the Preferred Address for an Interface

The *preferred address* on an interface is the default local address used for packets sourced by the local router to destinations on the subnet. By default, the numerically lowest local address is chosen. For example, if the addresses **172.16.1.1/12**, **172.16.1.2/12**, and **172.16.1.3/12** are configured on the same interface, the preferred address on the subnet (by default, **172.16.1.1**) would be used as a local address when you issue a **ping 172.16.1.5** command.

To set a different preferred address for the subnet, include the **preferred** statement:

```
preferred;
```

You can include this statement at the following hierarchy levels:

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

Operational Behavior of Interfaces When the Same IPv4 Address Is Assigned to Them

You can configure the same IPv4 address on multiple physical interfaces. When you assign the same IPv4 address to multiple physical interfaces, the operational behavior of those interfaces differs, depending on whether they are (implicitly) point-to-point or not.

> **NOTE:** For all interfaces except aggregated Ethernet, Fast Ethernet, and Gigabit Ethernet, you can explicitly configure an interface to be a point-to-point connection.

If you configure the same IP address on multiple interfaces in the same routing instance, Junos OS applies the configuration randomly on one of the interfaces. The other interfaces will remain without an IP address.

The following examples show the sample configuration of assigning the same IPv4 address to implicitly and explicitly point-to-point interfaces, and their corresponding **show interfaces terse** command outputs to see their operational status.

Configuring same IPv4 address on two non-p2p interfaces:

```
[edit]
user@host# show
ge-0/1/0 {
    unit 0 {
        family inet {
            address 200.1.1.1/24;
        }
```

```
    }
}
```

```
ge-3/0/1 {
    unit 0 {
        family inet {
            address 200.1.1.1/24;
        }
    }
}
```

The sample output shown below for the above configuration reveals that only **ge-0/1/0.0** was assigned the same IPv4 address **200.1.1.1/24** and its **link** state was **up**, while **ge-3/0/1.0** was not assigned the IPv4 address, though its **link** state was up, which means that it will be operational only when it gets a unique IPv4 address other than **200.1.1.1/24**.

`user@host>` **show interfaces terse ge***

```
Interface               Admin Link Proto    Local                 Remote
        ge-0/1/0                up    up
        ge-0/1/0.0              up    up   inet     200.1.1.1/24
                                   multiservice
        ge-0/1/1                up    down
        ge-3/0/0                up    down
        ge-3/0/1                up    up
        ge-3/0/1.0              up    up   inet
                                   multiservice
```

Configuring same IPv4 address on (implicit) p2p interfaces:

```
[edit]
user@host# show
so-0/0/0 {
    unit 0 {
        family inet {
            address 200.1.1.1/24;
        }
    }
}
so-0/0/3 {
    unit 0 {
        family inet {
            address 200.1.1.1/24;
```

```
    }
  }
}
```

The sample output shown below for the above configuration reveals that both **so-0/0/0.0** and **so-0/0/3.0** were assigned the same IPv4 address **200.1.1.1/24** and that their **link** states were down. The interfaces are down due to an issue with the link and not because same IPv4 address is assigned to both the interfaces. It is expected that not more than one of the interface is up at any given time (following a redundancy scheme outside of the JUNOS devices scope) as both being up may cause adverse effects.

`user@host>` **show interfaces terse so***

```
Interface               Admin Link Proto    Local                 Remote
so-0/0/0                up    up
so-0/0/0.0              up    down inet     200.1.1.1/24
so-0/0/1                up    up
so-0/0/2                up    down
so-0/0/3                up    up
so-0/0/3.0              up    down inet     200.1.1.1/24
so-1/1/0                up    down
so-1/1/1                up    down
so-1/1/2                up    up
so-1/1/3                up    up
so-2/0/0                up    up
so-2/0/1                up    up
so-2/0/2                up    up
so-2/0/3                up    down
```

Configuring same IPv4 address in multiple instances of a non-p2p interface

```
[edit interfaces]
user@host# show
ge-0/0/1 {
  vlan-tagging;
    unit 0 {
    vlan-id 1;
      family inet {
      address 1.1.1.1/24;
    }
  }
  unit 1{
    vlan-id 2;
      family inet {
```

```
            address 1.1.1.1/24;
        }
    }
}
```

On a non-P2P interface, you cannot configure the same local address on different units of different interfaces. In this case, commit error will be thrown and the configuration will not be successful.

Configuring same IPv4 address in multiple instances of the same p2p interface

```
[edit interfaces]
user@host# show
gr-0/0/10 {
    unit 0 {
        tunnel {
            source 1.1.1.1;
            destination 1.1.1.2;
        }
        family inet {
            mtu 1500;
            address 1.2.2.2/24;
        }
    }
    unit 1{
        family inet {
            address 1.2.2.2/24;
        }
    }
}
```

The sample output shown below for the above configuration reveals that only one interfaces gets successfully configured on P2P interfaces, when you try to configure same IPv4 address for multiple instance of different interfaces.

`user@host>` **show interfaces terse | match 1.2.2.2**

```
Interface                Admin Link Proto    Local      Remote
gr-0/0/10.0              up    up   inet     1.2.2.2/24
```

Configuring IPCP Options for Interfaces with PPP Encapsulation

For interfaces with PPP encapsulation, you can configure IPCP to negotiate IP address assignments and to pass network-related information such as Windows Name Service (WINS) and Domain Name System (DNS) servers, as defined in RFC 1877, *PPP Internet Protocol Control Protocol Extensions for Name Server Addresses*.

When you enable a PPP interface, you can configure an IP address, enable the interface to negotiate an IP address assignment from the remote end, or allow the interface to be unnumbered. You can also assign a destination profile to the remote end. The destination profile includes PPP properties, such as primary and secondary DNS and NetBIOS Name Servers (NBNSs). These options are described in the following sections:

> **NOTE:** The Junos OS does not request name servers from the remote end; the software does, however, send name servers to the remote end if requested.

Before you begin

You must configure the PPP encapsulation on the interface before configuring the IPCP option. On the logical interface, the following PPP encapsulation types are supported:

- **atm-mlppp-llc**
- **atm-ppp-llc**
- **atm-ppp-vc-mux**
- **multilink-ppp**

For more information about PPP encapsulation, see "Configuring Interface Encapsulation on Logical Interfaces" on page 154 and *Configuring ATM Interface Encapsulation*

- To configure an IP address for the interface, include the **address** statement in the configuration. For more information, see "Configuring the Interface Address" on page 166.

  If you include the **address** statement in the configuration, you cannot include the **negotiate-address** or **unnumbered-address** statement in the configuration.

  When you include the **address** statement in the interface configuration, you can assign PPP properties to the remote end.

  > **NOTE:** The option to negotiate an IP address is not allowed in MLFR and MFR encapsulations.

- To enable the interface to obtain an IP address from the remote end, include the **negotiate-address** statement at the **[edit interfaces *interface-name* unit *logical-unit-number* family inet]** hierarchy level.

```
[edit interfaces interface-name unit logical-unit-number family inet]
user@host# set negotiate-address
```

  > **NOTE:** If you include the **negotiate-address** statement in the configuration, you cannot include the **address** or **unnumbered-address** statement in the configuration.

- To configure an interface to be unnumbered, include the **unnumbered-address** and **destination** statements in the configuration.

```
[edit interfaces interface-name unit logical-unit-number family inet]
user@host# set unnumbered-address interface-name
user@host# set destination address
```

  > **NOTE:**
  > - The **unnumbered-address** statement enables the local address to be derived from the specified interface. The interface name must include a logical unit number and must have a configured address (see "Configuring the Interface Address" on page 166). Specify the IP address of the remote interface with the **destination** statement.
  > - If you include the **unnumbered-address** statement in the configuration, you cannot include the **address** or **negotiate-address** statement in the interface configuration.

- To assign PPP properties to the remote end include the **destination-profile** statement:

```
[edit interfaces interface-name unit logical-unit-number family inet address address]
user@host# set destination-profile name
```

```
[edit interfaces interface-name unit logical-unit-number family inet unnumbered-address interface-name]
user@host# set destination-profile name
```

NOTE:

- You can assign PPP properties to the remote end, after you include the **address** or **unnumbered-address** statement in the interface configuration.
- You define the profile at the **[edit access group-profile name ppp]** hierarchy level. For more information, see *Example: PPP MP for L2TP*

SEE ALSO

Example: PPP MP for L2TP

Configuring an Unnumbered Interface

IN THIS SECTION

This topic includes the following information:

Overview of Unnumbered Interfaces

When you need to conserve IP addresses, you can configure unnumbered interfaces. Setting up an unnumbered interface enables IP processing on the interface without assigning an explicit IP address to the interface. For IPv6, in which conserving addresses is not a major concern, you can configure unnumbered interfaces to share the same subnet across multiple interfaces. IPv6 unnumbered interfaces are only supported on Ethernet interfaces. The statements you use to configure an unnumbered interface depend on the type of interface you are configuring: a point-to-point interface or an Ethernet interface:

Configuring an Unnumbered Point-to-Point Interface

1. In configuration mode, go to the **[edit interfaces *interface-name* unit *logical-unit-number*]** hierarchy level.

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

2. To configure an unnumbered point-to-point interface, configure the protocol family, but do not include the **address** statement.

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

> **NOTE:**
> - For interfaces with PPP encapsulation, you can configure an unnumbered interface by including the **unnumbered-interface** statement in the configuration. For more information, see "Configuring IPCP Options for Interfaces with PPP Encapsulation" on page 175.
> - When configuring unnumbered interfaces, you must ensure that a source address is configured on some interface in the router. This address is the default address. We recommend that you do this by assigning an address to the loopback interface (**lo0**), as described in "Loopback Interface Configuration" on page 233. If you configure an address (other than a martian) on the **lo0** interface, that address is always the default address, which is preferable because the loopback interface is independent of any physical interfaces and therefore is always accessible.

Configuring an Unnumbered Ethernet or Demux Interface

1. In configuration mode, go to the **[edit interfaces *interface-name* unit *logical-unit-number* family *family-name*]** hierarchy level.

```
[edit ]
user@host# edit interfaces interface-name unit logical-unit-number family family-name
```

2. To configure an unnumbered Ethernet or demultiplexing interface, include the **unnumbered-address** statement in the configuration.

```
[edit interfaces interface-name unit logical-unit-number family family-name]
user@host# set unnumbered-address interface-name
```

3. (Optional) To specify the unnumbered Ethernet interface as the next-hop interface for a configured static route, include the **qualified-next-hop** statement at the **[edit routing-options static route *destination-prefix*]** hierarchy level. This feature enables you to specify independent preferences and metrics for static routes on a next-hop basis.

```
[edit routing-options static route destination-prefix]
user@host# set qualified-next-hop (address | interface-name)
```

NOTE:

- The **unnumbered-address** statement currently supports configuration of unnumbered demux interfaces only for the IPv4 address family. You can configure unnumbered Ethernet interfaces for both IPv4 and IPv6 address families.
- The interface that you configure to be unnumbered *borrows* an assigned IP address from another interface, and is referred to as the *borrower interface*. The interface from which the IP address is borrowed is referred to as the *donor interface*. In the **unnumbered-address** statement, ***interface-name*** specifies the donor interface. For an unnumbered Ethernet interface, the donor interface can be an Ethernet, ATM, SONET, or loopback interface that has a logical unit number and configured IP address and is not itself an unnumbered interface. For an unnumbered IP demultiplexing interface, the donor interface can be an Ethernet or loopback interface that has a logical unit number and configured IP address and is not itself an unnumbered interface. In addition, for either Ethernet or demux, the donor interface and the borrower interface must be members of the same routing instance and the same logical system.
- When you configure an unnumbered Ethernet or demux interface, the IP address of the donor interface becomes the source address in packets generated by the unnumbered interface.
- You can configure a host route that points to an unnumbered Ethernet or demux interface. For information about host routes, see the *MPLS Applications User Guide*.

Configuring a Preferred Source Address for Unnumbered Ethernet or Demux Interfaces

When a loopback interface with multiple secondary IP addresses is configured as the donor interface for an unnumbered Ethernet or demux interface, you can optionally specify any one of the loopback interface's secondary addresses as the preferred source address for the unnumbered Ethernet or demux interface. This feature enables you to use an IP address other than the primary IP address on some of the unnumbered Ethernet or demux interfaces in your network.

1. In configuration mode, go to the **[edit interfaces *interface-name* unit *logical-unit-number* family *family-name*]** hierarchy level.

```
[edit ]
user@host# edit interfaces interface-name unit logical-unit-number family family-name
```

2. To configure a secondary address on a loopback donor interface as the preferred source address for an unnumbered Ethernet or demux interface, include the **preferred-source-address** option in the **unnumbered-address** statement:

```
[edit interfaces interface-name unit logical-unit-number family family-name]
user@host# set unnumbered-address interface-name <preferred-source-address address
```

NOTE:
The following considerations apply when you configure a preferred source address on an unnumbered Ethernet or demux interface:

- The **unnumbered-address** statement currently supports the configuration of a preferred source address only for the IPv4 address family for demux interfaces, and for IPv4 and IPv6 address families for Ethernet interfaces.
- If you do not specify the preferred source address, the router uses the default primary IP address of the donor interface.
- You cannot delete an address on a donor loopback interface while it is being used as the preferred source address for an unnumbered Ethernet or demux interface.

Restrictions for Configuring Unnumbered Ethernet Interfaces

The following restrictions apply when you configure unnumbered Ethernet interfaces:

- The **unnumbered-address** statement currently supports the configuration of unnumbered Ethernet interfaces for IPv4 and IPv6 address families.
- You cannot assign an IP address to an Ethernet interface that is already configured as an unnumbered interface.
- The donor interface for an unnumbered Ethernet interface must have one or more configured IP addresses.
- The donor interface for an unnumbered Ethernet interfaced cannot be configured as unnumbered.
- An unnumbered Ethernet interface does not support configuration of the following **address** statement options: **arp**, **broadcast**, **primary**, **preferred**, and **vrrp-group**. For information about these options, see "Configuring the Interface Address" on page 166.
- Running IGMP and PIM are supported only on unnumbered Ethernet interfaces that directly face the host and have no downstream PIM neighbors. IGMP and PIM are not supported on unnumbered Ethernet interfaces that act as upstream interfaces in a PIM topology.
- Running OSPF and IS-IS on unnumbered Ethernet interfaces is not supported. However, you can run OSPF over unnumbered Ethernet interfaces configured as a Point-to-Point connection.

  For link-state distribution using an interior gateway protocol (IGP), ensure that OSPF is enabled on the donor interface for an unnumbered interface configuration, so the donor IP address is reachable to establish OSPF sessions.

NOTE: If you configure the same address on multiple interfaces in the same routing instance, Junos OS uses only the first configuration, the remaining address configurations are ignored and can leave interfaces without an address. Interfaces that do not have an assigned address cannot be used as a donor interface for an unnumbered Ethernet interface.

For example, in the following configuration the address configuration of interface xe-0/0/1.0 is ignored:

```
interfaces {
  xe-0/0/0 {
    unit 0 {
      family inet {
        address 192.168.1.1/24;
      }
    }
  }
  xe-0/0/1 {
    unit 0 {
     family inet {
       address 192.168.1.1/24;
     }
  }
}
```

For more information on configuring the same address on multiple interfaces, see "Configuring the Interface Address" on page 166.

Displaying the Unnumbered Ethernet Interface Configuration

Purpose

To display the configured unnumbered interface at the **[edit interfaces *interface-name* unit *logical-unit-number*]** hierarchy level:

- Unnumbered interface —ge-1/0/0
- Donor interface —ge-0/0/0
- Donor interface address —4.4.4.1/24

The unnumbered interface "borrows" an IP address from the donor interface.

Action

- Run the **show** command at the **[edit]** hierarchy level.

```
interfaces {
    ge-0/0/0 {
        unit 0 {
            family inet {
                address 4.4.4.1/24;
            }
        }
    }
    ge-1/0/0 {
        unit 0 {
            family inet {
                unnumbered-address ge-0/0/0.0;
            }
        }
    }
}
```

Meaning

The sample configuration that is described works correctly on M and T Series routers. For unnumbered interfaces on MX Series routers, you must additionally configure static routes on an unnumbered Ethernet interface by including the **qualified-next-hop** statement at the **[edit routing-options static route destination-prefix]** hierarchy level to specify the unnumbered Ethernet interface as the next-hop interface for a configured static route.

Displaying the Configured Preferred Source Address for an Unnumbered Ethernet Interface

Purpose

To display the configuration of preferred source address for an unnumbered interface at the **[edit interfaces *interface-name* unit *logical-unit-number* family inet]** hierarchy level:

- Unnumbered interface —ge-4/0/0
- Donor interface —lo0
- Donor interface primary address—2.2.2.1/32
- Donor interface secondary address—3.3.3.1/32

Action

- Run the **show** command at the **[edit]** hierarchy level.

```
interfaces {
    lo0 {
        unit 0 {
```

```
            family inet {
                address 2.2.2.1/32;
                address 3.3.3.1/32;
            }
        }
    }
}
interfaces {
    ge-4/0/0 {
        unit 0 {
            family inet {
                unnumbered-address lo0.0 preferred-source-address 3.3.3.1;
            }
        }
    }
}
```

Meaning

The loopback interface **lo0** is the donor interface from which unnumbered Ethernet interface **ge-4/0/0** “borrows” an IP address.

The example shows one of the loopback interface’s secondary addresses, 3.3.3.1, as the preferred source address for the unnumbered Ethernet interface.

Displaying the Configuration for Unnumbered Ethernet Interface as the Next Hop for a Static Route

Purpose

To display the unnumbered interface configured as the next hop for the static route at the **[edit interfaces *interface-name* unit *logical-unit-number* family inet]** hierarchy level:

- Unnumbered interface —ge-0/0/0
- Donor interface —lo0
- Donor interface primary address—5.5.5.1/32
- Donor interface secondary address—6.6.6.1/32
- Static route—7.7.7.1/32

Action

- Run the **show** command at the **[edit]** hierarchy level.

```
interfaces {
  ge-0/0/0 {
    unit 0 {
      family inet {
        unnumbered-address lo0.0;
      }
    }
  }
}
lo0 {
    unit 0 {
      family inet {
        address 5.5.5.1/32;
        address 6.6.6.1/32;
      }
    }
  }
```

- The following configuration enables the kernel to install a static route to address 7.7.7.1/32 with a next hop through unnumbered interface ge-0/0/0.0.

```
static {
route 7.7.7.1/32 {
  qualified-next-hop ge-0/0/0.0;
}
}
```

Meaning

In this example, **ge-0/0/0** is the unnumbered interface and a loopback interface, **lo0**, is the donor interface from which **ge-0/0/0** "borrows" an IP address. The example also configures a static route to **7.7.7.1/32** with a next hop through unnumbered interface **ge-0/0/0.0**.

Setting the Protocol MTU

When you initially configure an interface, the protocol maximum transmission unit (MTU) is calculated automatically. If you subsequently change the media MTU, the protocol MTU on existing address families automatically changes.

For a list of default protocol MTU values, see "Media MTU Sizes by Interface Type" on page 74.

To modify the MTU for a particular protocol family, include the **mtu** statement:

```
mtu bytes;
```

You can include this statement at the following hierarchy levels:

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

If you increase the size of the protocol MTU, you must ensure that the size of the media MTU is equal to or greater than the sum of the protocol MTU and the encapsulation overhead. For a list of encapsulation overhead values, see “Encapsulation Overhead by Interface Encapsulation Type” on page 88. If you reduce the media MTU size, but there are already one or more address families configured and active on the interface, you must also reduce the protocol MTU size. (You configure the media MTU by including the **mtu** statement at the **[edit interfaces *interface-name*]** hierarchy level.)

> **NOTE:** Changing the media MTU or protocol MTU causes an interface to be deleted and added again.

The maximum number of data-link connection identifiers (DLCIs) is determined by the MTU on the interface. If you have keepalives enabled, the maximum number of DLCIs is 1000, with the MTU set to 5012.

The actual frames transmitted also contain cyclic redundancy check (CRC) bits, which are not part of the MTU. For example, the default protocol MTU for a Gigabit Ethernet interface is 1500 bytes, but the largest possible frame size is actually 1504 bytes; you need to consider the extra bits in calculations of MTUs for interoperability.

SEE ALSO

Media MTU Overview | 73

Disabling the Removal of Address and Control Bytes

For Point-to-Point Protocol (PPP) CCC-encapsulated interfaces, the address and control bytes are removed by default before the packet is encapsulated into a tunnel.

You can disable the removal of address and control bytes. To do this, include the **keep-address-and-control** statement:

```
keep-address-and-control;
```

You can include this statement at the following hierarchy levels:

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

SEE ALSO

keep-address-and-control | **354**

Disabling the Transmission of Redirect Messages on an Interface

By default, the interface sends protocol redirect messages. To disable the sending of these messages on an interface, include the **no-redirects** statement:

```
no-redirects;
```

You can include this statement at the following hierarchy levels:

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

To disable the sending of protocol redirect messages for the entire router or switch, include the **no-redirects** statement at the **[edit system]** hierarchy level.

SEE ALSO

no-redirects

Applying a Filter to an Interface

IN THIS SECTION

Defining Interface Groups in Firewall Filters

When applying a firewall filter, you can define an interface to be part of an *interface group*. Packets received on that interface are tagged as being part of the group. You can then match these packets using the **interface-group** match statement, as described in the *Routing Policies, Firewall Filters, and Traffic Policers User Guide*.

To define the interface to be part of an interface group, include the **group** statement:

```
group filter-group-number;
```

You can include this statement at the following hierarchy levels:

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

> **NOTE:** The number 0 is not a valid interface group number.

Filter-Based Forwarding on the Output Interface

If port-mirrored packets are to be distributed to multiple monitoring or collection interfaces, based on patterns in packet headers, it is helpful to configure a filter-based forwarding (FBF) filter on the port-mirroring egress interface.

When an FBF filter is installed as an output filter, a packet that is forwarded to the filter has already undergone at least one route lookup. After the packet is classified at the egress interface by the FBF filter, it is redirected to another routing table for additional route lookup. To avoid packet looping inside the Packet Forwarding Engine, the route lookup in the latter routing table (designated by an FBF routing instance) must result in a different next hop from any next hop specified in a table that has already been applied to the packet.

If an input interface is configured for FBF, the source lookup is disabled for those packets headings to a different routing instance, since the routing table is not set up to handle the source lookup.

For more information about FBF configuration, see the *Junos OS Routing Protocols Library*. For more information about port mirroring, see the *Junos OS Services Interfaces Library for Routing Devices*.

Applying a Filter to an Interface

To apply firewall filters to an interface, include the **filter** statement:

```
filter {
    group filter-group-number;
    input filter-name;
    input-list [ filter-names ];
    output filter-name;
    output-list [ filter-names ];
}
```

To apply a single filter, include the **input** statement:

```
filter {
    input filter-name;
}
```

To apply a list of filters to evaluate packets received on an interface, include the **input-list** statement.

```
filter {
    input-list [ filter-names ];
}
```

Up to 16 filter names can be included in an input list.

To apply a list of filters to evaluate packets transmitted on an interface, include the **output-list** statement.

```
filter {
    output-list [ filter-names ];
}
```

When you apply filters using the **input-list** statement or the **output-list** statement, a new filter is created with the name *<interface-name>.<unit-direction>*. This filter is exclusively interface-specific.

You can include these statements at the following hierarchy levels:

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

In the **family** statement, the protocol family can be **ccc**, **inet**, **inet6**, **mpls**, or **vpls**.

In the **group** statement, specify the interface group number to associate with the filter.

In the **input** statement, list the name of one firewall filter to be evaluated when packets are received on the interface.

In the **input-list** statement, list the names of filters to evaluate when packets are received on the interface. You can include up to 16 filter names.

In the **output** statement, list the name of one firewall filter to be evaluated when packets are transmitted on the interface.

> **NOTE:** Output filters do not work for broadcast and multicast traffic, including VPLS traffic (except in MX Series routers with MPC/MIC interfaces), as shown in “Applying a Filter to an Interface” on page 188.

> **NOTE:** MPLS family firewall filters applied on the output interface are not supported on the PTX10003 router due to product limitation.

> **NOTE:** On an MX Series router, you cannot apply as an output filter, a firewall filter configured at the **[edit firewall filter family ccc]** hierarchy level. Firewall filters configured for the **family ccc** statement can be applied only as input filters.

In the **output-list** statement, list the names of filters to evaluate when packets are transmitted on the interface. You can include up to 16 filter names.

You can use the same filter one or more times. On M Series routers (except the M320 and M120 routers), if you apply a firewall filter or policer to multiple interfaces, the filter or policer acts on the sum of traffic entering or exiting those interfaces.

On T Series, M120, and M320 routers, interfaces are distributed among multiple packet forwarding components. Therefore, on these routers, if you apply a firewall filter or policer to multiple interfaces, the filter or policer acts on the traffic stream entering or exiting each interface, regardless of the sum of traffic on the multiple interfaces.

For more information on Understanding Ethernet Frame Statistics, see the *MX Series Layer 2 Configuration Guide*.

If you apply the filter to the interface **lo0**, it is applied to packets received or transmitted by the Routing Engine. You cannot apply MPLS filters to the management interface (**fxp0** or **em0**) or the loopback interface (**lo0**).

Filters applied at the **[set interfaces lo0 unit 0 family any filter input]** hierarchy level are not installed on T4000 Type 5 FPCs.

For more information about firewall filters, see the *Routing Policies, Firewall Filters, and Traffic Policers User Guide*. For more information about MPLS filters, see the *MPLS Applications User Guide*.

Example: Input Filter for VPLS Traffic

For M Series and T Series routers only, apply an input filter to VPLS traffic. Output filters do not work for broadcast and multicast traffic, including VPLS traffic. Note that on MX Series routers with MPC/MIC interfaces, the VPLS filters on the egress is applicable to broadcast, multicast, and unknown unicast traffic.

```
[edit interfaces]
fe-2/2/3 {
    vlan-tagging;
    encapsulation vlan-vpls;
    unit 601 {
        encapsulation vlan-vpls;
        vlan-id 601;
        family vpls {
            filter {
                input filter1;  # Works for multicast destination MAC address
                output filter1;  # Does not work for multicast destination MAC address
            }
        }
    }
}
[edit firewall]
family vpls {
    filter filter1 {
        term 1 {
            from {
                destination-mac-address {
                    01:00:0c:cc:cc:cd/48;
                }
            }
            then {
                discard;
            }
        }
        term 2 {
            then {
                accept;
            }
        }
    }
}
```

Example: Filter-Based Forwarding at the Output Interface

The following example illustrates the configuration of filter-based forwarding at the output interface. In this example, the packet flow follows this path:

1. A packet arrives at interface **fe-1/2/0.0** with source and destination addresses **10.50.200.1** and **10.50.100.1** respectively.
2. The route lookup in routing table **inet.0** points to the egress interface **so-0/0/3.0**.
3. The output filter installed at **so-0/0/3.0** redirects the packet to routing table **fbf.inet.0**.
4. The packet matches the entry **10.50.100.0/25** in the **fbf.inet.0** table, and finally leaves the router from interface **so-2/0/0.0**.

```
[edit interfaces]
so-0/0/3 {
    unit 0 {
        family inet {
            filter {
                output fbf;
            }
            address 10.50.10.2/25;
        }
    }
}
fe-1/2/0 {
    unit 0 {
        family inet {
            address 10.50.50.2/25;
        }
    }
}
so-2/0/0 {
    unit 0 {
        family inet {
            address 10.50.20.2/25;
        }
    }
}
[edit firewall]
filter fbf {
    term 0 {
        from {
            source-address {
```

```
            10.50.200.0/25;
          }
        }
        then routing-instance fbf;
      }
      term d {
        then count d;
      }
    }
    [edit routing-instances]
    fbf {
      instance-type forwarding;
      routing-options {
        static {
          route 10.50.100.0/25 next-hop so-2/0/0.0;
        }
      }
    }
    [edit routing-options]
    interface-routes {
      rib-group inet fbf-group;
    }
    static {
      route 10.50.100.0/25 next-hop 10.50.10.1;
    }
    rib-groups {
      fbf-group {
        import-rib [inet.0 fbf.inet.0];
      }
    }
```

Enabling Source Class and Destination Class Usage

IN THIS SECTION

Source Class and Destination Class Usage

For interfaces that carry IPv4, IPv6, MPLS, or peer AS billing traffic, you can maintain packet counts based on the entry and exit points for traffic passing through your network. Entry and exit points are identified by source and destination prefixes grouped into disjoint sets defined as *source classes* and *destination classes*. You can define classes based on a variety of parameters, such as routing neighbors, autonomous systems, and route filters.

Source class usage (SCU) counts packets sent to customers by performing lookup on the IP source address. SCU makes it possible to track traffic originating from specific prefixes on the provider core and destined for specific prefixes on the customer edge. You must enable SCU accounting on both the inbound and outbound physical interfaces, and the route for the source of the packet must be in located in the forwarding table.

> **NOTE:** SCU and DCU accounting do not work with directly connected interface routes. Source class usage does not count packets coming from sources with direct routes in the forwarding table because of software architecture limitations.

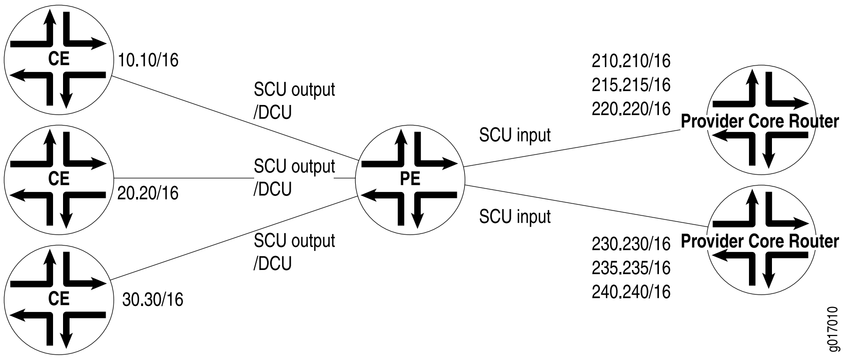
Destination class usage (DCU) counts packets from customers by performing lookup of the IP destination address. DCU makes it possible to track traffic originating from the customer edge and destined for specific prefixes on the provider core router.

> **NOTE:** We recommend that you stop the network traffic on an interface before you modify the DCU or SCU configuration for that interface. Modifying the DCU or SCU configuration without stopping the traffic might corrupt the DCU or SCU statistics. Before you restart the traffic after modifying the configuration, enter the **clear interfaces statistics** command.

Figure 11 on page 195 illustrates an Internet service provider (ISP) network. In this topology, you can use DCU to count packets customers send to specific prefixes. For example, you can have three counters, one per customer, that count the packets destined for prefix **210.210/16** and **220.220/16**.

You can use SCU to count packets the provider sends from specific prefixes. For example, you can count the packets sent from prefix **210.210/16** and **215.215/16** and transmitted on a specific output interface.

Figure 11: Prefix Accounting with Source and Destination Classes



You can configure up to 126 source classes and 126 destination classes. For each interface on which you enable destination class usage and source class usage, the Junos OS maintains an interface-specific counter for each corresponding class up to the 126 class limit.

> **NOTE:** For transit packets exiting the router through the tunnel, forwarding path features, such as RPF, forwarding table filtering, source class usage, and destination class usage are not supported on the interfaces you configure as the output interface for tunnel traffic. For firewall filtering, you must allow the output tunnel packets through the firewall filter applied to input traffic on the interface that is the next-hop interface towards the tunnel destination.

> **NOTE:**
>
> Performing DCU accounting when an output service is enabled produces inconsistent behavior in the following configuration:
>
> - Both SCU input and DCU are configured on the packet input interface.
> - SCU output is configured on the packet output interface.
> - Interface services is enabled on the output interface.
>
> For an incoming packet with source and destination prefixes matching the SCU and DCU classes respectively configured in the router, both SCU and DCU counters will be incremented. This behavior is not harmful or negative. However, it is inconsistent with non-serviced packets, in that only the SCU count will be incremented (because the SCU class ID will override the DCU class ID in this case).

To enable packet counting on an interface, include the **accounting** statement:

```
accounting {
```

```
        destination-class-usage;
        source-class-usage {
            direction;
        }
    }
```

direction can be one of the following:

- **input**—Configure at least one expected ingress point.
- **output**—Configure at least one expected egress point.
- **input output**—On a single interface, configure at least one expected ingress point and one expected egress point.

You can include these statements at the following hierarchy levels:

- **[edit interfaces *interface-name* unit *logical-unit-number* family (inet | inet6 | mpls)]**
- **[edit logical-systems *logical-system-name* interfaces *interface-name* unit *logical-unit-number* family (inet | inet6 | mpls)]**

For SCU to work, you must configure at least one input interface and at least one output interface.

The ability to count a single packet for both SCU and DCU accounting depends on the underlying physical interface.

- For traffic over MPC/MIC interfaces , a single incoming packet is counted for both SCU and DCU accounting if both SCU and DCU are configured. To ensure the outgoing packet is counted, include the **source-class-usage output** statements in the configuration of the outgoing interface.
- For traffic over DPC interfaces, an incoming packet is counted only once, and SCU takes priority over DCU. This means that when a packet arrives on an interface on which you include the **source-class-usage input** and **destination-class-usage** statements in the configuration, and when the source and destination both match accounting prefixes, the Junos OS associates the packet with the source class only.

For traffic over MPC interfaces , SCU and DCU accounting is performed after output filters are evaluated. If a packet matches a firewall filter match condition, the packet is included in SCU or DCU accounting except in the case where the action of the matched term is **discard**.

On T Series, M120, and M320 routers, the source class and destination classes are not carried across the router fabric. The implications of this are as follows:

- On T Series, M120, and M320 routers, SCU and DCU accounting is performed before the packet enters the fabric.
- On M7i, M10i, M120, and M320 routers, on MX Series routers with non-MPC, and on T Series routers, SCU and DCU accounting is performed before output filters are evaluated. Consequently, if a packet

matches a firewall filter match condition, the packet is included in SCU or DCU accounting; the packet is counted for any term action (including the **discard** action).

- On M120, M320, and T Series routers, the **destination-class** and **source-class** statements are supported at the **[edit firewall family *family-name* filter *filter-name* term *term-name* from]** hierarchy level only for the filter applied to the forwarding table. On M7i, M10i, and MX Series routers, these statements are supported.

Once you enable accounting on an interface, the Junos OS maintains packet counters for that interface, with separate counters for **inet**, **inet6**, and **mpls** protocol families. You must then configure the source class and destination class attributes in policy action statements, which must be included in forwarding-table export policies.

> **NOTE:** When configuring policy action statements, you can configure only one source class for each matching route. In other words, more than one source class cannot be applied to the same route.

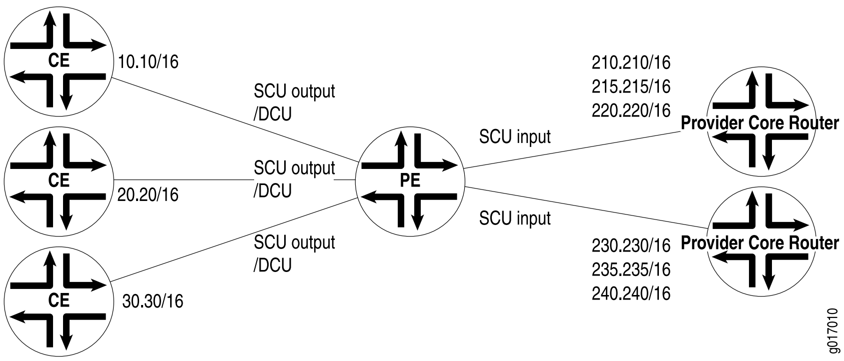
In Junos OS Release 9.3 and later, you can configure SCU accounting for Layer 3 VPNs configured with the **vrf-table-label** statement. Include the **source-class-usage** statement at the **[edit routing-instances *routing-instance-name* vrf-table-label]** hierarchy level. The **source-class-usage** statement at this hierarchy level is supported only for the virtual routing and forwarding (VRF) instance type.

> **NOTE:** DCU counters cannot be enabled on the label-switched interface (LSI) that is created dynamically when the **vrf-table-label** statement is configured within a VRF. For more information, see the *Junos OS VPNs Library for Routing Devices*.

For a complete discussion about source and destination class accounting profiles, see the *Network Management and Monitoring Guide*. For more information about MPLS, see the *MPLS Applications User Guide*.

Enabling Source Class and Destination Class Usage

Figure 12: Prefix Accounting with Source and Destination Classes



Configure DCU and SCU output on one interface:

```
[edit]
interfaces {
    so-6/1/0 {
        unit 0 {
            family inet {
                accounting {
                    destination-class-usage;
                    source-class-usage {
                        output;
                    }
                }
            }
        }
    }
}
```

1. **Complete SCU Configuration**

   Source routers A and B use loopback addresses as the prefixes to be monitored. Most of the configuration tasks and actual monitoring occur on transit Router SCU.

   The loopback address on Router A contains the origin of the prefix that is to be assigned to source class A on Router SCU. However, no SCU processing happens on this router. Therefore, configure Router A for basic OSPF routing and include your loopback interface and interface **so-0/0/2** in the OSPF process.

2. **Router A**

```
[edit]
interfaces {
    so-0/0/2 {
        unit 0 {
            family inet {
                address 10.255.50.2/24;
            }
        }
    }
    lo0 {
        unit 0 {
            family inet {
                address 10.255.192.10/32;
            }
        }
    }
}
protocols {
    ospf {
        area 0.0.0.0 {
            interface so-0/0/2.0;
            interface lo0.0;
        }
    }
}
```

3. **Router SCU**

   Last, apply the policy to the forwarding table.

   Router SCU handles the bulk of the activity in this example. On Router SCU, enable source class usage on the inbound and outbound interfaces at the **[edit interfaces *interface-name* unit *unit-number* family inet accounting]** hierarchy level. Make sure you specify the expected traffic: input, output, or, in this case, both.

   Next, configure a route filter policy statement that matches the prefixes of the loopback addresses from routers A and B. Include statements in the policy that classify packets from Router A in one group named **scu-class-a** and packets from Router B in a second class named **scu-class-b**. Notice the efficient use of a single policy containing multiple terms.

```
[edit]
interfaces {
```

```
    so-0/0/1 {
        unit 0 {
            family inet {
                accounting {
                    source-class-usage {
                        input;
                        output;
                    }
                }
                address 10.255.50.1/24;
            }
        }
    }
    so-0/0/3 {
        unit 0 {
            family inet {
                accounting {
                    source-class-usage {
                        input;
                        output;
                    }
                }
                address 10.255.10.3/24;
            }
        }
    }
    lo0 {
        unit 0 {
            family inet {
                address 10.255.6.111/32;
            }
        }
    }
}
protocols {
    ospf {
        area 0.0.0.0 {
            interface so-0/0/1.0;
            interface so-0/0/3.0;
        }
    }
}
routing-options {
    forwarding-table {
```

```
        export scu-policy;
    }
}
policy-options {
    policy-statement scu-policy {
        term 0 {
            from {
                route-filter 10.255.192.0/24 orlonger;
            }
            then source-class scu-class-a;
        }
        term 1 {
            from {
                route-filter 10.255.165.0/24 orlonger;
            }
            then source-class scu-class-b;
        }
    }
}
```

4. **Router B**

   Just as Router A provides a source prefix, Router B's loopback address matches the prefix assigned to **scu-class-b** on Router SCU. Again, no SCU processing happens on this router, so configure Router B for basic OSPF routing and include your loopback interface and interface **so-0/0/4** in the OSPF process.

```
interfaces {
    so-0/0/4 {
        unit 0 {
            family inet {
                address 10.255.10.4/24;
            }
        }
    }
    lo0 {
        unit 0 {
            family inet {
                address 10.255.165.226/32;
            }
        }
    }
}
protocols {
```

```
    ospf {
        area 0.0.0.0 {
            interface so-0/0/4.0;
            interface lo0.0;
        }
    }
}
```

5. **Enabling Packet Counting for Layer 3 VPNs**

   You can use SCU and DCU to count packets on Layer 3 VPNs. To enable packet counting for Layer 3 VPN implementations at the egress point of the MPLS tunnel, you must configure a virtual loopback tunnel interface (**vt**) on the PE router, map the virtual routing and forwarding (VRF) instance type to the virtual loopback tunnel interface, and send the traffic received from the VPN out the source class output interface, as shown in the following example:

   Configure a virtual loopback tunnel interface on a provider edge router equipped with a tunnel PIC:

```
[edit interfaces]
vt-0/3/0 {
    unit 0 {
        family inet {
            accounting {
                source-class-usage {
                    input;
                }
            }
        }
    }
}
```

6. Map the VRF instance type to the virtual loopback tunnel interface.

   In Junos OS Release 9.3 and later, you can configure SCU accounting for Layer 3 VPNs configured with the **vrf-table-label** statement. Include the **source-class-usage** statement at the **[edit routing-instances *routing-instance-name* vrf-table-label]** hierarchy level. The **source-class-usage** statement at this hierarchy level is supported only for the virtual routing and forwarding (VRF) instance type. DCU is not supported when the **vrf-table-label** statement is configured. For more information, see the *Junos OS VPNs Library for Routing Devices*.

```
[edit routing-instances]
VPN-A {
```

```
    instance-type vrf;
    interface at-2/1/1.0;
    interface vt-0/3/0.0;
    route-distinguisher 10.255.14.225:100;
    vrf-import import-policy-A;
    vrf-export export-policy-A;
    protocols {
        bgp {
            group to-r4 {
                local-address 10.27.253.1;
                peer-as 400;
                neighbor 10.27.253.2;
            }
        }
    }
}
```

7. Send traffic received from the VPN out the source class output interface:

```
[edit interfaces]
at-2/1/0 {
    unit 0 {
        family inet {
            accounting {
                source-class-usage {
                    output;
                }
            }
        }
    }
}
```

For more information about VPNs, see the *Junos OS VPNs Library for Routing Devices*. For more information about virtual loopback tunnel interfaces, see the *Junos OS Services Interfaces Library for Routing Devices*.

SEE ALSO

accounting | 286

destination-classes

family

forward-and-send-to-re | 335

source-classes

targeted-broadcast

unit

Understanding Targeted Broadcast

Targeted broadcast is a process of flooding a target subnet with Layer 3 broadcast IP packets originating from a different subnet. The intent of targeted broadcast is to flood the target subnet with the broadcast packets on a LAN interface without broadcasting to the entire network. Targeted broadcast is configured with various options on the egress interface of the router or switch and the IP packets are broadcast only on the LAN (egress) interface. Targeted broadcast helps you implement remote administration tasks such as backups and wake-on LAN (WOL) on a LAN interface, and supports virtual routing and forwarding (VRF) instances.

Regular Layer 3 broadcast IP packets originating from a subnet are broadcast within the same subnet. When these IP packets reach a different subnet, they are forwarded to the Routing Engine (to be forwarded to other applications). Because of this, remote administration tasks such as backups cannot be performed on a particular subnet through another subnet. As a workaround you can enable targeted broadcast, to forward broadcast packets that originate from a different subnet.

Layer 3 broadcast IP packets have a destination IP address that is a valid broadcast address for the target subnet. These IP packets traverse the network in the same way as unicast IP packets until they reach the destination subnet. In the destination subnet, if the receiving router has targeted broadcast enabled on the egress interface, the IP packets are forwarded to an egress interface and the Routing Engine or to an egress interface only. The IP packets are then translated into broadcast IP packets which flood the target subnet only through the LAN interface (if there is no LAN interface, the packets are discarded), and all hosts on the target subnet receive the IP packets. If targeted broadcast is not enabled on the receiving router, the IP packets are treated as regular Layer 3 broadcast IP packets and are forwarded to the Routing Engine. If targeted broadcast is enabled without any options, the IP packets are forwarded to the Routing Engine.

Targeted broadcast can be configured to forward the IP packets only to an egress interface, which is helpful when the router is flooded with packets to process, or to both an egress interface and the Routing Engine.

NOTE: Targeted broadcast does not work when the targeted broadcast option **forward-and-send-to-re** and the traffic sampling option **sampling** are configured on the same egress interface of an M320 router, a T640 router, or an MX960 router. To overcome this scenario, you must either disable one of the these options or enable the **sampling** option with the targeted broadcast option **forward-only** on the egress interface. For information about traffic sampling, see *Configuring Traffic Sampling*.

NOTE: Any firewall filter that is configured on the Routing Engine loopback interface (lo0) cannot be applied to IP packets that are forwarded to the Routing Engine as a result of a targeted broadcast. This is because broadcast packets are forwarded as flood next hop and not as local next hop traffic, and you can only apply a firewall filter to local next hop routes for traffic directed towards the Routing Engine.

SEE ALSO

targeted-broadcast

Configuring Targeted Broadcast

IN THIS SECTION

The following sections explain how to configure targeted broadcast on an egress interface and its options:

Configuring Targeted Broadcast and Its Options

You can configure targeted broadcast on an egress interface with different options. You can either allow the IP packets destined for a Layer 3 broadcast address to be forwarded on the egress interface and to send a copy of the IP packets to the Routing Engine or you can allow the IP packets to be forwarded on the egress interface only. Note that the packets are broadcast only if the egress interface is a LAN interface.

To configure targeted broadcast and its options:

1. Configure the physical interface.

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

2. Configure the logical unit number at the **[edit interfaces *interface-name*** hierarchy level.

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

3. Configure the protocol family as inet at the **[edit interfaces *interface-name* unit *interface-unit-number*** hierarchy level.

   ```
   [edit interfaces interface-name unit interface--unit-number]
   user@host# set family inet
   ```

4. Configure targeted broadcast at the **[edit interfaces *interface-name* unit *interface-unit-number* family inet** hierarchy level

   ```
   [edit interfaces interface-name unit interface--unit-number family inet]
   user@host# set targeted-broadcast
   ```

5. Specify one of the following options as per requirement:

   - To allow IP packets destined for a Layer 3 broadcast address to be forwarded on the egress interface and to send a copy of the IP packets to the Routing Engine.

     ```
     [edit interfaces interface-name unit interface-unit-number family inet targeted-broadcast]
     user@host# set forward-and-send-to-re
     ```

   - To allow IP packets to be forwarded on the egress interface only.

```
[edit interfaces interface-name unit interface-unit-number family inet targeted-broadcast]
user@host# set forward-only
```

NOTE: Targeted broadcast does not work when the targeted broadcast option **forward-and-send-to-re** and the traffic sampling option **sampling** are configured on the same egress interface of an M320 router, a T640 router, or an MX960 router. To overcome this scenario, you must either disable one of the these options or enable the **sampling** option with the targeted broadcast option **forward-only** on the egress interface. For information about traffic sampling, see *Configuring Traffic Sampling*.

Display Targeted Broadcast Configuration Options

IN THIS SECTION

The following topics display targeted broadcast configuration with its various options:

Forward IP Packets On the Egress Interface and To the Routing Engine

Purpose

Display the configuration when targeted broadcast is configured on the egress interface to forward the IP packets on the egress interface and to send a copy of the IP packets to the Routing Engine.

Action

To display the configuration run the **show** command at the **[edit interfaces *interface-name* unit *interface-unit-number* family inet]** where the interface name is ge-2/0/0, the unit value is set to 0, the protocol family is set to inet.

```
[edit interfaces interface-name unit interface-unit-number family inet]
user@host#show
targeted-broadcast {
   forward-and-send-to-re;
}
```

Forward IP Packets On the Egress Interface Only

Purpose

Display the configuration when targeted broadcast is configured on the egress interface to forward the IP packets on the egress interface only.

Action

To display the configuration run the **show** command at the **[edit interfaces *interface-name* unit *interface-unit-number* family inet]** where the interface name is ge-2/0/0, the unit value is set to 0, the protocol family is set to inet.

```
[edit interfaces interface-name unit interface-unit-number family inet]
user@host#show
targeted-broadcast {
    forward-only;
}
```

SEE ALSO

targeted-broadcast

2

CHAPTER

Other Interfaces

Discard Interfaces

IN THIS SECTION

The discardinterface *dsc* is not a physical interface, but a virtual interface that discards packets.

The following sections explain discard interfaces in detail.

Discard Interfaces Overview

IN THIS SECTION

Discard Interfaces

You can configure the **inet** family protocol on the discard interface, which allows you to apply an output filter to the interface. If you apply an output filter to the interface, the action specified by the filter is executed before the traffic is discarded.

After you configure a discard interface, you must then configure a local policy to forward attacking traffic to the discard interface.

Benefits of Discard Interfaces

- The discard interface allows you to identify the ingress point of a denial-of-service (DoS) attack. When your network is under attack, the target host IP address is identified, and the local policy forwards attacking packets to the discard interface. When traffic is routed out of the discard interface, the traffic is silently discarded.

Guidelines to Follow When Configuring a Discard Interface

Keep the following guidelines in mind when configuring the discard interface:

- Only the logical interface unit 0 is supported.
- The **filter** and **address** statements are optional.
- Although you can configure an input filter and a filter group, these configuration statements have no effect because traffic is not transmitted from the discard interface.
- The discard interface does not support class of service (CoS).

Configuring Discard Interfaces

IN THIS SECTION

The discard (dsc) interface is a virtual interface that silently discards packets as they arrive. It is especially useful if the network is under a denial-of-service (DoS) attack, because you can configure a policy to drop millions of requests from being sent to a given target address, or set of addresses.

In addition, with a discard interface, you can configure filters for counting, logging, and sampling the traffic (which you cannot do with **discard static routes**).

Note that a discard interface can have only one logical unit (unit 0), but you can configure multiple IP addresses on that unit.

In M and MX series routers, the discard interface is supported for **inet**, **mpls**, and **vpls** traffic families. Starting in Junos release 20.1, for MX Series routers, the discard interface is also supported for the **inet6** family.

The following sections explain how to configure a discard interface:

Configuring and Usage of Discard Interface

To configure a discard interface:

1. In configuration mode, go to the **[edit interfaces]** hierarchy level.

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

2. Configure the discard interface. Note that you must use 'dsc' to configure discard interface and ensure that there is no discard interface already configured.

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

3. Configure the logical interface and the protocol family.

   ```
   [edit interfaces dsc]
   user@host# edit unit 0 family family
   ```

4. If appropriate, apply an output filter to the discard interface.

   Input filters have no impact in this context.

   ```
   [edit interfaces dsc unit 0 family family]
   user@host# set filter output filter-name
   ```

5. Commit the configuration and go to the top of the hierarchy level.

   ```
   [edit interfaces dsc unit 0 family family]
   user@host# commit
   user@host# top
   ```

Configure an Output Filter with Output policy

You must configure an output policy to set up the community on the routes injected into the network.

To configure an output policy.

1. In configuration mode, go to the **[edit policy-options]** hierarchy level.

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

2. Configure a routing policy.

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

3. Configure a policy term with a name.

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

4. Configure the list of prefix-lists of routes to match with a name.

```
[edit policy-options policy-statement statement-name term term-variable]
user@host# set from prefix-list name
```

5. Configure the action that is to be taken when the if and to conditions match with the **then** statement. In this case, configure the BGP community properties (set, add, and delete) associated with a route.

```
[edit policy-options policy-statement statement-name term term-variable]
user@host# set then community (set | add | delete) community-name
```

6. Commit the configuration and go to the top of the hierarchy level.

```
[edit interfaces dsc unit 0 family family]
user@host# commit
user@host# top
```

SEE ALSO

Example: Forwarding Packets to the Discard Interface

IP Demultiplexing Interfaces

IN THIS SECTION

Demultiplexing (demux) interfaces are logical interfaces that share a common, underlying interface. You can create logical subscriber interfaces using static or dynamic demultiplexing interfaces. In addition, you can use IP demultiplexing interfaces or VLAN demultiplexing interfaces when creating logical subscriber interfaces.

Demultiplexing Interface Overview

IN THIS SECTION

Demux interfaces support only Gigabit Ethernet, Fast Ethernet, 10-Gigabit Ethernet, or aggregated Ethernet underlying interfaces.

Demux interfaces are supported on M120 or MX Series routers only.

NOTE: You can also configure demux interfaces dynamically. For information about how to configure dynamic IP demux or dynamic VLAN demux interfaces, see *Configuring Dynamic Subscriber Interfaces Using IP Demux Interfaces in Dynamic Profiles* or *Configuring Dynamic Subscriber Interfaces Using VLAN Demux Interfaces in Dynamic Profiles*.

To configure static demux interfaces, see "Configuring a VLAN Demultiplexing Interface" on page 223 and "Configuring an IP Demultiplexing Interface" on page 218.

IP Demux Interface Overview

IP demux interfaces use the IP source address or IP destination address to demultiplex received packets when the subscriber is not uniquely identified by a Layer 2 circuit.

To determine which IP demux interface to use, the destination or source prefix is matched against the destination or source address of packets that the underlying interface receives. The underlying interface family type must match the demux interface prefix type.

VLAN Demux Interface Overview

VLAN demux interfaces use the VLAN ID to demultiplex received packets when the subscriber is not uniquely identified. A VLAN demux interface uses an underlying logical interface to receive packets.

To determine which VLAN demux interface to use, the VLAN ID is matched against that which the underlying interface receives.

NOTE: VLAN demux subscriber interfaces over aggregated Ethernet physical interfaces are supported only for MX Series routers that have only Trio MPCs installed. If the router has other MPCs in addition to Trio MPCs, theCLI accepts the configuration but errors are reported when the subscriber interfaces are brought up.

Guidelines to Remember When Configuring A Demux Interface

IN THIS SECTION

Keep the following guidelines in mind when configuring the demux interface:

- Demux interfaces are supported on M120 or MX Series routers only.
- Only demux0 is supported. If you configure another demux interface, such as demux1, the configuration commit fails.
- You can configure only one **demux0** interface per chassis, but you can define logical demux interfaces on top of it (for example, **demux0.1**, **demux0.2**, and so on).
- If the address in a received packet does not match any demux prefix, the packet is logically received on the underlying interface. For this reason, the underlying interface is often referred to as the *primary* interface.

Points to Remember When Configuring an IP Demux Interface

In addition to the guidelines in "Guidelines to Remember When Configuring A Demux Interface" on page 215, the following guidelines are to be noted when configuring an IP demux interface:

- You must associate demux interfaces with an underlying logical interface.

> **NOTE:** IP demux interfaces currently support only Gigabit Ethernet, Fast Ethernet, 10-Gigabit Ethernet, and aggregated Ethernet underlying interfaces.

- The demux underlying interface must reside on the same logical system as the demux interfaces that you configure over it.
- IP demux interfaces currently supports the Internet Protocol version 4 (IPv4) suite inet and Internet Protocol version 6 (IPv6) suite inet6 family types.
- You can configure more than one demux prefix for a given demux unit. However, you cannot configure the exact same demux prefix on two different demux units with the same underlying interface.
- You can configure overlapping demux prefixes on two different demux units with the same underlying prefix. However, under this configuration, best match rules apply (in other words, the most specific prefix wins).

Points to Remember When Configuring a VLAN Demux Interface

In addition to the guidelines in "Guidelines to Remember When Configuring A Demux Interface" on page 215, the following guidelines are to be noted when configuring a VLAN demux interface:

- You must associate VLAN demux interfaces with an underlying logical interface.

> **NOTE:** VLAN demux interfaces currently support only Gigabit Ethernet, Fast Ethernet, 10-Gigabit Ethernet, and aggregated Ethernet underlying interfaces.

- The demux underlying interface must reside on the same logical system as the demux interfaces that you configure over it.
- VLAN demux interfaces currently supports the Internet Protocol version 4 (IPv4) suite inet and Internet Protocol version 6 (IPv6) suite inet6 family types.

MAC Address Validation on Static Demux Interfaces

IN THIS SECTION

MAC address validation enables the router to validate that received packets contain a trusted IP source and an Ethernet MAC source address.

MAC address validation is supported on static demux interfaces on MX Series routers only.

There are two types of MAC address validation that you can configure:

Loose

Forwards packets when both the IP source address and the MAC source address match one of the trusted address tuples.

Drops packets when the IP source address matches one of the trusted tuples, but the MAC address does not support the MAC address of the tuple

Continues to forward packets when the source address of the incoming packet does not match any of the trusted IP addresses.

Strict

Forwards packets when both the IP source address and the MAC source address match one of the trusted address tuples.

Drops packets when the MAC address does not match the tuple's MAC source address, or when IP source address of the incoming packet does not match any of the trusted IP addresses.

SEE ALSO

Associating VLAN IDs to VLAN Demux Interfaces

Binding VLAN IDs to Logical Interfaces

Subscriber Interfaces and Demultiplexing Overview

Configuring an IP Demultiplexing Interface

Demultiplexing (demux) interfaces are logical interfaces that share a common, underlying interface. You can configure IP demultiplexing interfaces or VLAN demultiplexing interfaces.

To configure an IP demux interface, you must configure the demux prefixes that are used by the underlying interface and then configure the IP demultiplexing interface as explained in the following tasks:

Configuring an IP Demux Underlying Interface

An IP demux interface uses an underlying logical interface to receive packets. To determine which IP demux interface to use, the destination or source prefix is matched against the destination or source address of packets that the underlying interface receives. The underlying interface family type must match the demux interface prefix type.

> **NOTE:** IP demux interfaces currently support only Gigabit Ethernet, Fast Ethernet, 10-Gigabit Ethernet, and aggregated Ethernet underlying interfaces.

To configure a logical interface as an IP demux underlying interface with demux source:

1. In configuration mode, go to the **[edit interfaces]** hierarchy level:

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

2. Configure the interface as fe-*x*/*y*/*z* and the logical interface with the **unit** statement. Note that IP demux interfaces currently support only Gigabit Ethernet, Fast Ethernet, 10-Gigabit Ethernet, and aggregated Ethernet underlying interfaces. In this procedure, we show a Fast Ethernet interface as an example.

```
[edit interfaces]
```

```
user@host# edit fe-x/y/z unit logical-unit-number
```

3. Configure the logical demux source family type on the IP demux underlying interface as inet or inet6, or both.

```
[edit interfaces fe-x/y/z unit logical-unit-number]
user@host# set demux-source (inet | inet6)
```

or

```
[edit interfaces fe-x/y/z unit logical-unit-number]
user@host# set demux-source [inet inet6]
```

4. (Optional) To improve datapath performance for DHCPv4 subscribers, specify that only subscribers with 32-bit prefixes are allowed to come up on the interface.

```
[edit interfaces fe-x/y/z unit logial-unit-number]
user@host# set host-prefix-only
```

> **NOTE:** This step requires that you specify the **demux-source** as only **inet**. A commit error occurs if you specify only **inet6** or both **inet** and **inet6**.

5. Save the configuration and move to top of the hierarchy level.

```
[edit interfaces fe-x/y/z unit logial-unit-number]
user@host# commit
user@host# top
```

To configure a logical interface as an IP demux underlying interface with demux destination:

1. In configuration mode, go to the **[edit interfaces]** hierarchy level:

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

2. Configure the interface as fe-*x/y/z* and the logical interface with the **unit** statement. Note that IP demux interfaces currently support only Gigabit Ethernet, Fast Ethernet, 10-Gigabit Ethernet, and aggregated Ethernet underlying interfaces.

```
[edit interfaces]
user@host# edit fe-x/y/z unit logical-unit-number unit logical-unit-number
```

3. Configure the logical demux destination family type on the IP demux underlying interface as inet or inet6.

```
[edit interfaces fe-x/y/z unit logical-unit-number]
user@host# set demux-destination (inet | inet6)
```

4. Save the configuration and move to top of the hierarchy level.

```
[edit interfaces fe-x/y/z unit logial-unit-number]
user@host# commit
user@host# top
```

Configuring the IP Demux Interface

You can configure one or more logical demux source prefixes or destination prefixes after specifying an underlying interface for the static demux interface to use. This underlying interface must reside on the same logical system as the demux interface.

You configure demux prefixes for use by the underlying interface. The demux prefixes can represent individual hosts or networks. For a given demux interface unit, you can configure either demux source or demux destination prefixes but not both.

You can choose not to configure a demux source or demux destination prefix. This type of configuration results in a transmit-only interface.

To configure the IP demux interface with source prefix:

1. In configuration mode, go to the **[edit interfaces]** hierarchy level:

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

2. Configure the interface as a logical demux interface (for example, demux0 interface) and configure the logical interface with the **unit** statement.

NOTE: You can configure only one demux0 interface per chassis, but you can define logical demux interfaces on top of it (for example, demux0.1, demux0.2, and so on).

```
[edit interfaces]
user@host# edit demux0 unit logical-unit-number
```

3. Configure the underlying interface on which the demux interface is running under the **demux-options** statement.

```
[edit interfaces demux0 unit logical-unit-number]
user@host# set demux-options underlying-interface interface-name
```

4. Configure the protocol family.

```
[edit interfaces demux0 unit logical-unit-number]
user@host# edit family family
```

5. Configure one or more logical demux source prefixes (IP address). The prefixes are matched against the source address of packets that the underlying interface receives. When a match occurs, the packet is processed as if it was received on the demux interface.

```
[edit interfaces demux0 unit logical-unit-number family family]
user@host# set demux-source source-prefix
```

6. Save the configuration and move to top of the hierarchy level.

```
[edit interfaces demux0 unit logical-unit-number family family]
user@host# commit
user@host# top
```

To configure the IP demux interface with destination prefix:

1. In configuration mode, go to the **[edit interfaces]** hierarchy level:

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

2. Configure the interface as a logical demux interface (for example, demux0 interface) and configure the logical interface with the **unit** statement.

    NOTE: You can configure only one demux0 interface per chassis, but you can define logical demux interfaces on top of it (for example, demux0.1, demux0.2, and so on).

    ```
    [edit interfaces]
    user@host# edit demux0 unit logical-unit-number
    ```

3. Configure the underlying interface on which the demux interface is running under the **demux-options** statement.

    ```
    [edit interfaces demux0 unit logical-unit-number]
    user@host# set demux-options underlying-interface interface-name
    ```

4. Configure the protocol family.

    ```
    [edit interfaces demux0 unit logical-unit-number]
    user@host# edit family family
    ```

5. Configure one or more logical demux destination prefixes. The prefixes are matched against the destination address of packets that the underlying interface receives. When a match occurs, the packet is processed as if it was received on the demux interface.

    ```
    [edit interfaces demux0 unit logical-unit-number family family]
    user@host# set demux-destination destination-prefix
    ```

6. Save the configuration and move to top of the hierarchy level.

    ```
    [edit interfaces demux0 unit logical-unit-number family family]
    user@host# commit
    user@host# top
    ```

Configuring MAC Address Validation on Static IP Demux Interfaces

MAC address validation enables the router to validate that received packets contain a trusted IP source and an Ethernet MAC source address.

To configure MAC address validation for an IP demux interface:

1. In configuration mode, go to the **[edit interfaces demux0 unit *logical-unit-number*]** hierarchy level:

```
[edit]
user@host# edit interfaces demux0 unit logical-unit-number
```

2. Configure the protocol family for the interface.

```
[edit interfaces demux0 unit logical-unit-number]
user@host# edit family family
```

3. Configure the **mac-validate** statement to validate source MAC address with loose or strict options.

```
[edit interfaces demux0 unit logical-unit-number family family]
user@host# set mac-validate (loose | strict)
```

4. Save the configuration and move to top of the hierarchy level.

```
[edit interfaces demux0 unit logical-unit-number family family]
user@host# commit
user@host# top
```

Configuring a VLAN Demultiplexing Interface

Demultiplexing (demux) interfaces are logical interfaces that share a common, underlying interface. You can configure IP demultiplexing interfaces or VLAN demultiplexing interfaces.

To configure a VLAN demux interface, you must configure the demux prefixes that are used by the underlying interface and then configure the VLAN demultiplexing interface as explained by the following tasks:

1. Configuring a VLAN Demux Underlying Interface | 224
2. Configuring the VLAN Demux Interface | 226

Configuring a VLAN Demux Underlying Interface

A VLAN demux interface uses an underlying logical interface to receive packets. To determine which VLAN demux interface to use, the VLAN ID is matched against that which the underlying interface receives.

> **NOTE:** VLAN demux interfaces currently support only Gigabit Ethernet, Fast Ethernet, 10-Gigabit Ethernet, and aggregated Ethernet underlying interfaces.
>
> VLAN demux subscriber interfaces over aggregated Ethernet physical interfaces are supported only for MX Series routers that have only Trio MPCs installed. If the router has other MPCs in addition to Trio MPCs, the CLI accepts the configuration but errors are reported when the subscriber interfaces are brought up

To configure a logical interface as a VLAN demux underlying interface with demux source:

1. In configuration mode, go to the **[edit interfaces]** hierarchy level:

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

2. Configure the interface as fe-x/y/z and the logical interface with the **unit** option.

   ```
   [edit interfaces]
   user@host# edit fe-x/y/z unit logical-unit-number unit logical-unit-number
   ```

3. Configure the VLAN ID. The VLAN ID is used to determine which VLAN demux interface to use, that is the VLAN ID is matched against that which the underlying interface receives.

   ```
   [edit interfaces fe-x/y/z unit logical-unit-number]
   user@host# set vlan-id number
   ```

4. Configure the logical demux source family type on the VLAN demux underlying interface as inet or inet6.

   ```
   [edit interfaces fe-x/y/z unit logical-unit-number]
   ```

```
user@host# set demux-source (inet | inet6)
```

5. Save the configuration and move to top of the hierarchy level.

```
[edit interfaces fe-x/y/z unit logial-unit-number]
user@host# commit
user@host# top
```

To configure a logical interface as a VLAN demux underlying interface with demux destination:

1. In configuration mode, go to the **[edit interfaces]** hierarchy level:

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

2. Configure the interface as fe-x/y/z and the logical interface with the **unit** option.

```
[edit interfaces]
user@host# edit fe-x/y/z unit logical-unit-number unit logical-unit-number
```

3. Configure the VLAN ID. The VLAN ID is used to determine which VLAN demux interface to use, that is the VLAN ID is matched against that which the underlying interface receives.

```
[edit interfaces fe-x/y/z unit logical-unit-number]
user@host# set vlan-id number
```

4. Configure the logical demux destination family type on the VLAN demux underlying interface as inet or inet6.

```
[edit interfaces fe-x/y/z unit logical-unit-number]
user@host# set demux-destination (inet | inet6)
```

5. Save the configuration and move to top of the hierarchy level.

```
[edit interfaces fe-x/y/z unit logial-unit-number]
user@host# commit
user@host# top
```

Configuring the VLAN Demux Interface

You can configure one or more logical demux source prefixes or destination prefixes after specifying an underlying interface for the static demux interface to use. This underlying interface must reside on the same logical system as the demux interface.

You configure demux prefixes for use by the underlying interface. The demux prefixes can represent individual hosts or networks. For a given demux interface unit, you can configure either demux source prefix or demux destination prefixes but not both.

You can choose not to configure a demux source prefix or a demux destination prefix. This type of configuration results in a transmit-only interface

To configure VLAN demux interface with demux source prefix:

1. In configuration mode, go to the **[edit interfaces]** hierarchy level:

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

2. Configure the interface as a logical demux interface (for example, demux0 interface) and configure the logical interface with the **unit** statement.

   > **NOTE:** You can configure only one demux0 interface per chassis, but you can define logical demux interfaces on top of it (for example, demux0.1, demux0.2, and so on).

   ```
   [edit interfaces]
   user@host# edit demux0 unit logical-unit-number
   ```

3. Configure the underlying interface on which the demux interface is running under the **demux-options** statement.

   ```
   [edit interfaces demux0 unit logical-unit-number]
   user@host# set demux-options underlying-interface interface-name
   ```

4. Configure the protocol family for the interface.

   ```
   [edit interfaces demux0 unit logical-unit-number]
   user@host# edit family family
   ```

5. Configure one or more logical demux source prefixes. The prefixes are matched against the source address of packets that the underlying interface receives. When a match occurs, the packet is processed as if it was received on the demux interface.

   [edit interfaces demux0 unit *logical-unit-number* family *family*]
   user@host# **set demux-source *source-prefix***

6. Save the configuration and move to top of the hierarchy level.

   [edit interfaces demux0 unit *logical-unit-number*]
   user@host# **commit**
   user@host# **top**

To configure VLAN demux interface with demux destination prefix:

1. In configuration mode, go to the **[edit interfaces]** hierarchy level:

   [edit]
   user@host# **edit interfaces**

2. Configure the interface as a logical demux interface (for example, demux0 interface) and configure the logical interface with the **unit** statement.

   NOTE: You can configure only one demux0 interface per chassis, but you can define logical demux interfaces on top of it (for example, demux0.1, demux0.2, and so on).

   [edit interfaces]
   user@host# **edit demux0 unit *logical-unit-number***

3. Configure the underlying interface on which the demux interface is running under the **demux-options** statement.

   [edit interfaces demux0 unit *logical-unit-number*]
   user@host# **set demux-options underlying-interface *interface-name***

4. Configure the protocol family for the interface.

```
[edit interfaces demux0 unit logical-unit-number]
user@host# edit family family
```

5. Configure one or more logical demux destination prefixes. The prefixes are matched against the destination address of packets that the underlying interface receives. When a match occurs, the packet is processed as if it was received on the demux interface.

```
[edit interfaces demux0 unit logical-unit-number family family]
user@host# set demux-destination destination-prefix
```

6. Save the configuration and move to top of the hierarchy level.

```
[edit interfaces demux0 unit logical-unit-number]
user@host# commit
user@host# top
```

Configuring MAC Address Validation on Static VLAN Demux Interfaces

MAC address validation enables the router to validate that received packets contain a trusted IP source and an Ethernet MAC source address.

To configure MAC address validation for a VLAN demux interface:

1. In configuration mode, go to the **[edit interfaces demux0 unit *logical-unit-number*]** hierarchy level:

```
[edit]
user@host# edit interfaces demux0 unit logical-unit-number
```

2. Configure the protocol family for the interface.

```
[edit interfaces demux0 unit logical-unit-number]
user@host# edit family family
```

3. Configure the **mac-validate** statement to validate source MAC address with loose or strict options.

```
[edit interfaces demux0 unit logical-unit-number family family]
user@host# set mac-validate (loose | strict)
```

4. Save the configuration and move to top of the hierarchy level.

   [edit interfaces demux0 unit *logical-unit-number* family *family*]
   user@host# **commit**
   user@host# **top**

Verifying a Demux Interface Configuration

Purpose

Check the configuration of a demux interface and its underlying interface when the following are configured:

- Two VLANs are configured, where each VLAN consists of two IP demux interfaces.
- One VLAN demultiplexes based on the source address
- The other VLAN demultiplexes based on the destination address.

Action

From configuration mode on the MX Series router, run the **show interfaces fe-0/0/0** and **show interfaces demux0** configuration mode commands.

user@host> show interfaces fe-0/0/0

```
vlan-tagging;
unit 100 {
 vlan-id 100;
 demux-source inet; # Enable demux of inet prefixes
 family inet {
  address 10.1.1.1/24;
  filter {
   input vlan1-primary-in-filter;
   output vlan1-primary-out-filter;
  }
  mac-validate loose;
 }
}
unit 200 {
 vlan-id 200;
 demux-destination inet; # Enable demux of inet using destination addresses
 family inet {
  address 20.1.1.1/24;
 }
}
```

```
unit 300 {
 vlan-id 300;
 demux-source inet; # Enable demux of inet using source addresses
 family inet {
  address 20.1.2.1/24;
 }
}
```

user@host> show interfaces demux0

```
unit 101 {
 description vlan1-sub1;
 demux-options {
  underlying-interface fe-0/0/0.100;
 }
 family inet {
  demux-source 10.1.1.0/24;
  filter {
   input vlan1-sub1-in-filter;
   output vlan1-sub1-out-filter;
  }
 mac-validate loose;
 }
}
unit 102 {
 description vlan1-sub2;
 demux-options {
  underlying-interface fe-0/0/0.100;
 }
 family inet {
  demux-source {
   10.1.0.0/16;
   10.2.1.0/24;
  }
  filter {
   input vlan1-sub2-in-filter;
   output vlan1-sub2-out-filter;
  }
  mac-validate loose;
 }
}
unit 202 {
```

```
 description vlan2-sub2;
 demux-options {
  underlying-interface fe-0/0/0.200;
 }
 family inet {
  demux-destination 100.1.2.0/24;
 }
}
unit 302 {
 description vlan2-sub2;
 demux-options {
  underlying-interface fe-0/0/0.300;
 }
 family inet {
  demux-source 100.1.2.0/24;
 }
}
```

Loopback Interfaces

IN THIS SECTION

This topic discusses about the use of loopback interface, step-by-step procedure on how to configure loopback interfaces with examples.

Understanding the Loopback Interface

The Internet Protocol (IP) specifies a loopback network with the (IPv4) address **127.0.0.0/8**. Most IP implementations support a loopback interface (**lo0**) to represent the loopback facility. Any traffic that a computer program sends on the loopback network is addressed to the same computer. The most commonly used IP address on the loopback network is **127.0.0.1** for IPv4 and **::1** for IPv6. The standard domain name for the address is **localhost**.

A network device also includes an internal loopback address (**lo0.16384**). The internal loopback address is a particular instance of the loopback address with the logical unit number 16384.

The loopback interface is used to identify the device. While any interface address can be used to determine if the device is online, the loopback address is the preferred method. Whereas interfaces might be removed or addresses changed based on network topology changes, the loopback address never changes.

When you ping an individual interface address, the results do not always indicate the health of the device. For example, a subnet mismatch in the configuration of two endpoints on a point-to-point link makes the link appear to be inoperable. Pinging the interface to determine whether the device is online provides a misleading result. An interface might be unavailable because of a problem unrelated to the device's configuration or operation. You can use the loopback interface to address these issues.

Benefits of Loopback Interface

- As the loopback address never changes, it is the best way to identify a device in the network.
- The loopback interface is always up and it is reachable as long as the route to that IP address is available in the IP routing table. Hence you can use the loopback interface for diagnostics and troubleshooting purposes.
- Protocols such as OSPF use the loopback address to determine protocol-specific properties for the device or network. Further, some commands such as **ping mpls** require a loopback address to function correctly.
- You can apply stateless firewall filters to the loopback address to filter packets originating from, or destined for, the Routing Engine.
- Junos OS creates the loopback interface for the internal routing instance, which prevents any filter on **lo0.0** from disrupting internal traffic.

SEE ALSO

Understanding Interfaces

Loopback Interface Configuration

IN THIS SECTION

Configuring the Loopback Interface

When specifying the loopback address, do not include a destination prefix. Also, in most cases, do not specify a loopback address on any unit other than unit 0.

> **NOTE:** For Layer 3 virtual private networks (VPNs), you can configure multiple logical units for the loopback interface. This allows you to configure a logical loopback interface for each virtual routing and forwarding (VRF) routing instance. For more information, see the *Junos OS VPNs Library for Routing Devices*.
>
> For some applications, such as SSL for Junos XML protocol, the address for the interface **lo0.0** must be **127.0.0.1**.

You can configure loopback interfaces using a subnetwork address for both inet and inet6 address families. Many protocols require a subnetwork address as their source address. Configuring a subnetwork loopback address as a donor interface enables these protocols to run on unnumbered interfaces.

If you configure the loopback interface, it is automatically used for unnumbered interfaces. If you do not configure the loopback interface, the router chooses the first interface to come online as the default. If you configure more than one address on the loopback interface, we recommend that you configure one to be the primary address to ensure that it is selected for use with unnumbered interfaces. By default, the primary address is used as the source address when packets originate from the interface.

On the router, you can configure the physical loopback interface, **lo0**, and one or more addresses on the interface. You can configure more than just **unit 0** for **lo0**, but each additional unit needs to be applied somewhere other than the main instance.

1. To configure the physical loopback interface, include the following statements at the **[edit interfaces]** hierarchy level:

```
[edit interfaces]
lo0 {
    unit 0 {
        family inet {
            address loopback-address;
            address <loopback-address2>;
            ...
        }
        family inet6 {
            address loopback-address;
        }
    }
}
```

Example: Configuring Two Addresses on the Loopback Interface with Host Routes

To configure two addresses on the loopback interface with host routes:

```
[edit]
user@host# edit interfaces lo0 unit 0 family inet
[edit interfaces lo0 unit 0 family inet]
user@host# set address 172.16.0.1
[edit interfaces lo0 unit 0 family inet]
user@host# set address 10.0.0.1
[edit interfaces lo0 unit 0 family inet]
user@host# top
[edit]
user@host# show
interfaces {
    lo0 {
        unit 0 {
            family inet {
                10.0.0.1/32;
                127.0.0.1/32;
                172.16.0.1/32;
            }
        }
    }
}
```

Example: Configuring Two Addresses on the Loopback Interface with Subnetwork Routes

To configure two addresses on the loopback interface with subnetwork routes:

```
[edit]
user@host# edit interfaces lo0 unit 0 family inet
[edit interfaces lo0 unit 0 family inet]
user@host# set address 192.16.0.1/24
[edit interfaces lo0 unit 0 family inet]
user@host# set address 10.2.0.1/16
[edit interfaces lo0 unit 0 family inet]
user@host# top
[edit]
user@host# show
interfaces {
    lo0 {
        unit 0 {
            family inet {
                10.2.0.1/16;
                127.0.0.1/32;
                192.16.0.1/24;
            }
        }
    }
}
```

Example: Configuring an IPv4 and an IPv6 Address on the Loopback Interface with Subnetwork Routes

To configure an IPv4 and an IPv6 address on the loopback interface with subnetwork routes:

```
[edit]
user@host# edit interfaces lo0 unit 0 family inet
[edit interfaces lo0 unit 0 family inet]
user@host# set address 192.16.0.1/24
[edit interfaces lo0 unit 0 family inet]
user@host# up
[edit interfaces lo0 unit 0 family]
user@host# edit interfaces lo0 unit 0 family inet6
[edit interfaces lo0 unit 0 family inet6]
user@host# set address 3ffe::1:200:f8ff:fe75:50df/64
```

```
[edit interfaces lo0 unit 0 family inet6]
user@host# top
[edit]
user@host# show
interfaces {
    lo0 {
        unit 0 {
            family inet {
                127.0.0.1/32;
                192.16.0.1/24;
            }
            family inet6 {
                3ffe::1:200:f8ff:fe75:50df/64;
            }
        }
    }
}
```

RELATED DOCUMENTATION

Junos OS VPNs Library for Routing Devices

Serial Interfaces

IN THIS SECTION

This topic discusses about the serial interfaces, and how to configure serial line protocol, serial clocking mode, serial signal handling, serial DTR circuit, serial signal polarities, serial loopback capability, and serial line encoding.

Serial Interfaces Overview

Devices that communicate over a serial interface are divided into two classes: data terminal equipment (DTE) and data circuit-terminating equipment (DCE). Juniper Networks Serial Physical Interface Cards (PICs) have two ports per PIC and support full-duplex data transmission. These PICs support DTE mode only. On the Serial PIC, you can configure three types of serial interfaces, which are based on the following standards:

- EIA-530—An Electronics Industries Alliance (EIA) standard for the interconnection of DTE and DCE using serial binary data interchange with control information exchanged on separate control circuits.
- V.35—An ITU-T standard describing a synchronous, physical layer protocol used for communications between a network access device and a packet network. V.35 is most commonly used in the United States and in Europe.
- X.21—An ITU-T standard for serial communications over synchronous digital lines. The X.21 protocol is used primarily in Europe and Japan.

There are no serial interface-specific logical properties. For information about general logical properties that you can configure, see *Configuring Logical Interface Properties*. This support on serial interfaces is the same as the existing LFI and MLPPP support on T1 and E1 interfaces.

Benefits of Serial Interfaces

- Serial interface are a simple, cost-effective way to connect transmitting and receiving devices or ICs. A serial interface requires fewer conducting wires (often only one) than other interfaces, which eases implementation.
- Serial interfaces support long-distance communication.

Configuring the Serial Line Protocol

IN THIS SECTION

Configuring the Serial Line Protocol

By default, serial interfaces use the EIA-530 line protocol. You can configure each port on the PIC independently to use one of the following line protocols:

- EIA-530
- V.35
- X.21

To configure the serial line protocol:

1. Include the **line-protocol** statement, specifying the **eia530**, **v.35**, or **x.21** option:

   ```
   line-protocol protocol;
   ```

   You can include these statements at the following hierarchy levels:

   - **[edit interfaces se-*pim*/0/*port* serial-options]**
   - **[edit interfaces se-*fpc*/*pic*/*port* serial-options]**

For more information about serial interfaces, see the following sections:

Serial Interface Default Settings

IN THIS SECTION

Serial Interface Default Settings

IN THIS SECTION

EIA-530 Interface Default Settings

If you do not include the **line-protocol** statement or if you explicitly configure the default EIA-530 line protocol, the default settings are as follows:

```
dce-options | dte-options {
    cts normal;
    dcd normal;
    dsr normal;
    dtr normal;
    rts normal;
    tm normal;
}
clock-rate 16.384mhz;
clocking-mode loop;
cts-polarity positive;
dcd-polarity positive;
dsr-polarity positive;
dtr-circuit balanced;
dtr-polarity positive;
encoding nrz;
rts-polarity positive;
tm-polarity positive;
```

NOTE: On M Series routers, you can set the DCE clocking mode for EIA-530 interfaces and commit. An error message is not displayed and the CLI is not blocked.

You can include the line-protocol statement at the following hierarchy levels:

- **[edit interfaces se-*pim*/0/*port* serial-options]**
- **[edit interfaces se-*fpc*/*pic*/*port* serial-options]**

V.35 Interface Default Settings

If you include the **line-protocol v.35** statement, the default settings are as follows:

```
dce-options | dte-options {
    cts normal;
    dcd normal;
    dsr normal;
    dtr normal;
    rts normal;
}
```

```
clock-rate 16.384mhz;
clocking-mode loop;
cts-polarity positive;
dcd-polarity positive;
dsr-polarity positive;
dtr-circuit balanced;
dtr-polarity positive;
encoding nrz;
rts-polarity positive;
```

You can include the line-protocol statement at the following hierarchy levels:

- **[edit interfaces se-*pim*/0/*port* serial-options]**
- **[edit interfaces se-*fpc*/*pic*/*port* serial-options]**

X.21 Interface Default Settings

If you include the **line-protocol x.21** statement, the default settings are as follows:

```
dce-options | dte-options {
   control-signal normal;
   indication normal;
}
clock-rate 16.384mhz;
clocking-mode loop;
control-polarity positive;
encoding nrz;
indication-polarity positive;
```

You can include the line-protocol statement at the following hierarchy levels:

- **[edit interfaces se-*pim*/0/*port* serial-options]**
- **[edit interfaces se-*fpc*/*pic*/*port* serial-options]**

Invalid Serial Interface Statements

IN THIS SECTION

The following sections show the invalid configuration statements for each type of serial interface. If you include the following statements in the configuration, an error message indicates the location of the error and the configuration is not activated.

Invalid EIA-530 Interface Statements

If you do not include the **line-protocol** statement or if you explicitly configure the default EIA-530 line protocol, the following statements are invalid:

```
dce-options | dte-options {
    control-signal (assert | de-assert | normal);
    indication (ignore | normal | require);
}
control-polarity (negative | positive);
indication-polarity (negative | positive);
```

You can include the line-protocol statement at the following hierarchy levels:

- **[edit interfaces se-*pim*/0/*port* serial-options]**
- **[edit interfaces se-*fpc*/*pic*/*port* serial-options]**

Invalid V.35 interface Statements

If you include the **line-protocol v.35** statement, the following statements are invalid:

```
dce-options | dte-options {
    control-signal (assert | de-assert | normal);
    indication (ignore | normal | require);
    tm (ignore | normal | require);
}
control-polarity (negative | positive);
indication-polarity (negative | positive);
loopback (dce-local | dce-remote);
tm-polarity (negative | positive);
```

You can include the line-protocol statement at the following hierarchy levels:

- **[edit interfaces se-*pim*/0/*port* serial-options]**
- **[edit interfaces se-*fpc*/*pic*/*port* serial-options]**

Invalid X.21 Interface Statements

If you include the **line-protocol x.21** statement, the following statements are invalid:

```
dce-options | dte-options {
```

```
    cts (ignore | normal | require);
    dcd (ignore | normal | require);
    dsr (ignore | normal | require);
    dtr (assert | de-assert | normal);
    rts (assert | de-assert | normal);
    tm (ignore | normal | require);
}
clocking-mode (dce | internal);
cts-polarity (negative | positive);
dce-polarity (negative | positive);
dsr-polarity (negative | positive);
dtr-circuit (balanced | unbalanced);
dtr-polarity (negative | positive);
loopback (dce-local | dce-remote);
rts-polarity (negative | positive);
tm-polarity (negative | positive);
```

You can include the **line-protocol** statement at the following hierarchy levels:

- **[edit interfaces se-*pim*/0/*port* serial-options]**
- **[edit interfaces se-*fpc*/*pic*/*port* serial-options]**

Configuring the Serial Clocking Mode

IN THIS SECTION

Configuring the Serial Clocking Mode

By default, serial interfaces use loop clocking mode. For EIA-530 and V.35 interfaces, you can configure each port on the PIC independently to use loop, DCE, or internal clocking mode. For X.21 interfaces, only loop clocking mode is supported.

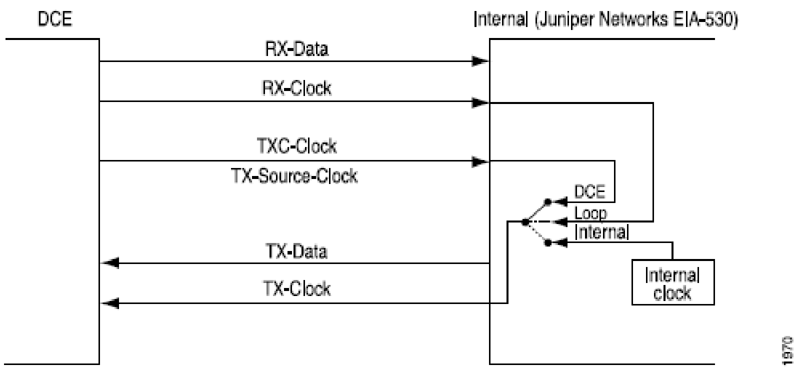
The three clocking modes work as follows:

- Loop clocking mode—Uses the DCE's RX clock to clock data from the DCE to the DTE.
- DCE clocking mode—Uses the TXC clock, which is generated by the DCE specifically to be used by the DTE as the DTE's transmit clock.
- Internal clocking mode—Also known as line timing, uses an internally generated clock. You can configure the speed of this clock by including the **clock-rate** statement at the **[edit interfaces se-*pim*/0/*port* serial-options]** or **[edit interfaces se-*fpc*/*pic*/*port* dte-options]** hierarchy levels. For more information about the DTE clock rate, see "Configuring the DTE Clock Rate" on page 244.

Note that DCE clocking mode and loop clocking mode use external clocks generated by the DCE.

Figure 13 on page 243 shows the clock sources of loop, DCE, and internal clocking modes.

Figure 13: Serial Interface Clocking Mode



To configure the clocking mode of a serial interface, include the **clocking-mode** statement:

```
clocking-mode (dce | internal | loop);
```

You can include this statement at the following hierarchy levels:

- **[edit interfaces se-*pim*/0/*port* serial-options]**
- **[edit interfaces se-*fpc*/*pic*/*port* serial-options]**

Inverting the Serial Interface Transmit Clock

When an externally timed clocking mode (DCE or loop) is used, long cables might introduce a phase shift of the DTE-transmitted clock and data. At high speeds, this phase shift might cause errors. Inverting the transmit clock corrects the phase shift, thereby reducing error rates.

By default, the transmit clock is not inverted. To invert the transmit clock, include the **transmit-clock invert** statement:

```
transmit-clock invert;
```

You can include this statement at the following hierarchy levels:

- **[edit interfaces se-*pim*/0/*port* serial-options]**
- **[edit interfaces se-*fpc*/*pic*/*port* serial-options]**

Configuring the DTE Clock Rate

By default, the serial interface has a clock rate of 16.384 MHz. For EIA-530 and V.35 interfaces with internal clocking mode configured, you can configure the clock rate.

To configure the clock rate, include the **clock-rate** statement:

```
clock-rate rate;
```

You can include this statement at the following hierarchy levels:

- **[edit interfaces se-*pim*/0/*port* serial-options]**
- **[edit interfaces se-*fpc*/*pic*/*port* serial-options]**

You can configure the following interface speeds:

- 2.048 MHz
- 2.341 MHz
- 2.731 MHz
- 3.277 MHz
- 4.096 MHz
- 5.461 MHz
- 8.192 MHz
- 16.384 MHz

Although the serial interface is intended for use at the default rate of 16.384 MHz, you might need to use a slower rate if any of the following conditions prevail:

- The interconnecting cable is too long for effective operation.
- The interconnecting cable is exposed to an extraneous noise source that might cause an unwanted voltage in excess of +1 volt measured differentially between the signal conductor and circuit common at the load end of the cable, with a 50-ohm resistor substituted for the generator.
- You need to minimize interference with other signals.
- You need to invert signals.

For detailed information about the relationship between signaling rate and interface cable distance, see the following standards:

- EIA-422-A, *Electrical Characteristics of Balanced Voltage Digital Interface Circuits*
- EIA-423-A, *Electrical Characteristics of Unbalanced Voltage Digital Interface Circuits*

Configuring the Serial Signal Handling

By default, normal signal handling is enabled for all signals. For each signal, the **normal** option applies to the normal signal handling for that signal, as defined by the following standards:

- TIA/EIA Standard 530
- ITU-T Recommendation V.35
- ITU-T Recommendation X.21

Table 22 on page 245 shows the serial interface modes that support each signal type.

Table 22: Signal Handling by Serial Interface Type

Signal	Serial Interfaces
From-DCE signals	
Clear to send (CTS)	EIA-530 and V.35
Data carrier detect (DCD)	EIA-530 and V.35
Data set ready (DSR)	EIA-530 and V.35
Indication	X.21 only
Test mode (TM)	EIA-530 only
To-DCE signals	

Table 22: Signal Handling by Serial Interface Type *(continued)*

Signal	Serial Interfaces
Control signal	X.21 only
Data transfer ready (DTR)	EIA-530 and V.35
Request to send (RTS)	EIA-530 and V.35

You configure serial interface signal characteristics by including the **dce-options** or **dte-options** statement:

```
dce-options |dte-options {
    control-signal (assert | de-assert | normal);
    cts (ignore | normal | require);
    dcd (ignore | normal | require);
    dsr (ignore | normal | require);
    dtr signal-handling-option;
    ignore-all;
    indication (ignore | normal | require);
    rts (assert | de-assert | normal);
    tm (ignore | normal | require);
}
```

You can include these statements at the following hierarchy levels:

- **[edit interfaces se-*pim*/0/*port* serial-options]**
- **[edit interfaces se-*fpc*/*pic*/*port* serial-options]**

For EIA-530 and V.35 interfaces, configure to-DCE signals by including the **dtr** and **rts** statements, specifying the **assert**, **de-assert**, or **normal** option:

```
dtr (assert | de-assert | normal);
rts (assert | de-assert | normal);
```

For X.21 interfaces, configure to-DCE signals by including the **control-signal** statement, specifying the **assert**, **de-assert**, or **normal** option:

```
control-signal (assert | de-assert | normal);
```

Assertion is when the positive side of a given signal is at potential high-level output voltage (Voh), while the negative side of the same signal is at potential low-level output voltage (Vol). *Deassertion* is when the positive side of a given signal is at potential Vol, while the negative side of the same signal is at potential Voh.

For the DTR signal, you can configure normal signal handling using the signal for automatic resynchronization by including the **dtr** statement, and specifying the **auto-synchronize** option:

```
dtr {
  auto-synchronize {
    duration milliseconds;
    interval seconds;
  }
}
```

The pulse duration of resynchronization can be from 1 through 1000 milliseconds. The offset interval for resynchronization can be from 1 through 31 seconds.

For EIA-530 and V.35 interfaces, configure from-DCE signals by including the **cts**, **dcd**, and **dsr** statements, specifying the **ignore**, **normal**, or **require** option:

```
cts (ignore | normal | require);
dcd (ignore | normal | require);
dsr (ignore | normal | require);
```

For X.21 interfaces, configure from-DCE signals by including the **indication** statement, specifying the **ignore**, **normal**, or **require** option:

```
indication (ignore | normal | require);
```

For EIA-530 interfaces only, you can configure from-DCE test-mode (TM) signaling by including the **tm** statement, specifying the **ignore**, **normal**, or **require** option:

```
tm (ignore | normal | require);
```

To specify that the from-DCE signal must be asserted, include the **require** option in the configuration. To specify that the from-DCE signal must be ignored, include the **ignore** option in the configuration.

NOTE: For V.35 and X.21 interfaces, you cannot include the **tm** statement in the configuration.

For X.21 interfaces, you cannot include the **cts**, **dcd**, **dsr**, **dtr**, and **rts** statements in the configuration.

For EIA-530 and V.35 interfaces, you cannot include the **control-signal** and **indication** statements in the configuration.

For a complete list of serial options statements that are not supported by each serial interface mode, see "Invalid Serial Interface Statements" on page 240.

To return to the default normal signal handling, delete the **require**, **ignore**, **assert**, **de-assert**, or **auto-synchronize** statement from the configuration, as shown in the following example:

```
[edit]
user@host# delete interfaces se-fpc/pic/port dte-options control-leads cts require
```

To explicitly configure normal signal handling, include the **control-signal** statement with the **normal** option:

```
control-signal normal;
```

You can configure the serial interface to ignore all control leads by including the **ignore-all** statement:

```
ignore-all;
```

You can include the **ignore-all** statement in the configuration only if you do not explicitly enable other signal handling options at the **[edit interfaces se-*pim*/0/*port* serial-options dce-options]** or **[edit interfaces se-*fpc*/*pic*/*port* serial-options dte-options]** hierarchy levels.

You can include the **control-signal**, **cts**, **dcd**, **dsr**, **dtr**, **indication**, **rts**, and **tm** statements at the following hierarchy levels:

- **[edit interfaces se-*pim*/0/*port* serial-options dte-options]**
- **[edit interfaces se-*fpc*/*pic*/*port* serial-options dte-options]**

Configuring the Serial DTR Circuit

A balanced circuit has two currents that are equal in magnitude and opposite in phase. An unbalanced circuit has one current and a ground; if a pair of terminals is unbalanced, one side is connected to electrical ground and the other carries the signal. By default, the DTR circuit is balanced.

For EIA-530 and V.35 interfaces, configure the DTR circuit by including the **dtr-circuit** statement:

```
dtr-circuit (balanced | unbalanced);
```

You can include this statement at the following hierarchy levels:

- **[edit interfaces se-*pim*/0/*port* serial-options]**
- **[edit interfaces se-*fpc*/*pic*/*port* serial-options]**

Configuring Serial Signal Polarities

Serial interfaces use a differential protocol signaling technique. Of the two serial signals associated with a circuit, the one referred to as the A signal is denoted with a plus sign, and the one referred to as the B signal is denoted with a minus sign; for example, DTR+ and DTR–. If DTR is low, then DTR+ is negative with respect to DTR–. If DTR is high, then DTR+ is positive with respect to DTR–.

By default, all signal polarities are positive. You can reverse this polarity on a Juniper Networks serial interface. You might need to do this if signals are miswired as a result of reversed polarities.

For EIA-530 and V.35 interfaces, configure signal polarities by including the **cts-polarity**, **dcd-polarity**, **dsr-polarity**, **dtr-polarity**, **rts-polarity**, and **tm-polarity** statements:

```
cts-polarity (negative | positive);
dcd-polarity (negative | positive);
dsr-polarity (negative | positive);
dtr-polarity (negative | positive);
rts-polarity (negative | positive);
tm-polarity (negative | positive);
```

You can include these statements at the following hierarchy levels:

- **[edit interfaces se-*pim*/0/*port* serial-options]**
- **[edit interfaces se-*fpc*/*pic*/*port* serial-options]**

For X.21 interfaces, configure signal polarities by including the **control-polarity** and **indication-polarity** statements:

```
control-polarity (negative | positive);
indication-polarity (negative | positive);
```

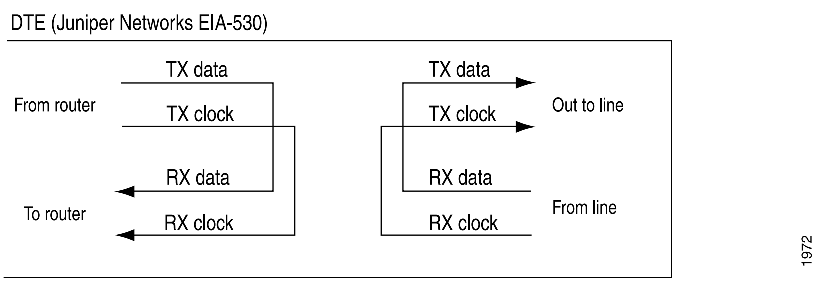
You can include these statements at the following hierarchy levels:

- **[edit interfaces se-*pim*/0/*port* serial-options]**
- **[edit interfaces se-*fpc*/*pic*/*port* serial-options]**

Configuring Serial Loopback Capability

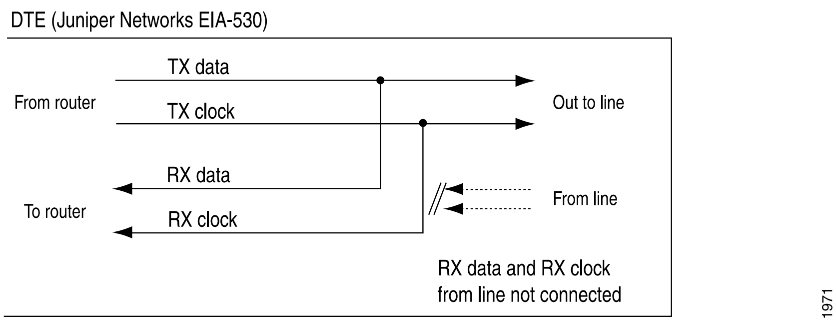
From the router, remote line interface unit (LIU) loopback loops the TX (transmit) data and TX clock back to the router as RX (receive) data and RX clock. From the line, LIU loopback loops the RX data and RX clock back out the line as TX data and TX clock, as shown in Figure 14 on page 250.

Figure 14: Serial Interface LIU Loopback



DCE local and DCE remote control the EIA-530 interface-specific signals for enabling local and remote loopback on the link partner DCE. Local loopback is shown in Figure 15 on page 251.

Figure 15: Serial Interface Local Loopback



For EIA-530 interfaces, you can configure DCE local, DCE remote, local, and remote (LIU) loopback capability.

For V.35, you can configure remote LIU and local loopback capability. DCE local and DCE remote loopbacks are not supported on V.35 and X.21 interfaces. Local and remote loopbacks are not supported on X.21 interfaces.

To configure the loopback capability on a serial interface, include the **loopback** statement, specifying the **dce-local**, **dce-remote**, **local**, or **remote** option:

```
loopback mode;
```

You can include this statement at the following hierarchy levels:

- **[edit interfaces se-*pim*/0/*port* serial-options]**
- **[edit interfaces se-*fpc*/*pic*/*port* serial-options]**

To disable the loopback capability, remove the **loopback** statement from the configuration:

```
[edit]
user@host# delete interfaces se-fpc/pic/port serial-options loopback
```

You can determine whether there is an internal or external problem by checking the error counters in the output of the **show interface se-*fpc*/*pic*/*port* extensive** command:

```
user@host> show interfaces se-fpc/pic/port extensive
```

To Configure Serial Loopback Capability:

1. To determine the source of a problem, loop the packets on the local router, the local DCE, the remote DCE, and the remote line interface unit (LIU).

2. To do this, include the **no-keepalives** and **encapsulation cisco-hdlc** statements at the **[edit interfaces se-*fpc*/*pic*/*port*]** hierarchy level, and the **loopback local** option at the **[edit interfaces se-*pim*/0/*port* serial-options]** or **[edit interfaces se-*fpc*/*pic*/*port* serial-options]** hierarchy level. With this configuration, the link stays up, so you can loop ping packets to a remote router. The **loopback local** statement causes the interface to loop within the PIC just before the data reaches the transceiver.

```
[edit interfaces]
se-1/0/0 {
    no-keepalives;
    encapsulation cisco-hdlc;
    serial-options {
        loopback local;
    }
    unit 0 {
        family inet {
            address 10.100.100.1/24;
        }
    }
}
```

Configuring Serial Line Encoding

By default, serial interfaces use non-return to zero (NRZ) line encoding. You can configure non-return to zero inverted (NRZI) line encoding if necessary.

To have the interface use NRZI line encoding, include the **encoding** statement, specifying the **nrzi** option:

```
encoding nrzi;
```

To explicitly configure the default NRZ line encoding, include the **encoding** statement, specifying the **nrz** option:

```
encoding nrz;
```

You can include this statement at the following hierarchy levels:

- **[edit interfaces se-*pim*/0/*port* serial-options]**
- **[edit interfaces se-*fpc*/*pic*/*port* serial-options]**

When setting the line encoding parameter, you must set the same value for paired ports. Ports 0 and 1 must share the same value.

3

CHAPTER

Monitor and Troubleshooting Interfaces

Monitoring Interfaces

IN THIS SECTION

This topic discusses about tracing operations of individual router interface, interface process, and pppd process.

Tracing Interface Operations Overview

You can trace the operations of individual router interfaces and those of the interface process (dcd). For a general discussion of tracing and of the precedence of multiple tracing operations, see the *Junos OS Administration Library*.

For information about the operations of Virtual Router Resolution Protocol (VRRP)-enabled interfaces, see the *High Availability User Guide*.

SEE ALSO

Tracing Operations of an Individual Router Interface | 254

Tracing Operations of the Interface Process | 255

Tracing Operations of an Individual Router Interface

To trace the operations of individual router interfaces, perform the following steps:

1. In configuration mode, go to the **[edit interfaces *interface-name*]** hierarchy level:

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

2. Configure the **traceoptions** option.

```
[edit interfaces interface-name]
user@host# edit traceoptions
```

3. Configure the tracing flag.

```
[edit interfaces interface-name traceoptions]
user@host# set flag flag-option
```

You can specify the following interface tracing flags:

- **all**—Trace all interface operations.
- **event**—Trace all interface events.
- **ipc**—Trace all interface interprocess communication (IPC) messages.
- **media**—Trace all interface media changes.

The interfaces **traceoptions** statement does not support a trace file. The logging is done by the kernel, so the tracing information is placed in the system **syslog** files.

For more information about trace operations, see "Tracing Operations of the Interface Process" on page 255.

SEE ALSO

traceoptions

Tracing Operations of the Interface Process

To trace the operations of the router or switch interface process, dcd, perform the following steps:

1. In configuration mode, go to the **[edit interfaces]** hierarchy level:

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

2. Configure the **traceoptions** statement.

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

3. Configure the **no-remote-trace** option to disable remote tracing.

```
[edit interfaces traceoptions]
user@host# set no-remote-trace
```

4. Configure the **file *filename*** option.

```
[edit interfaces traceoptions]
user@host# edit file
```

5. Configure the **files *number*** option, **match *regular-expression*** option, **size *size*** option, and **world-readable | no-world-readable** option.

```
[edit interfaces traceoptions file]
user@host# set files number
user@host# set match regular-expression
user@host# set size size
user@host# set word-readable | no-world-readable
```

6. Configure the tracing flag.

```
[edit interfaces traceoptions]
user@host# set flag flag-option
```

7. Configure the **disable** option in **flag *flag-option*** statement to disable the tracing operation. You can use this option to disable a single operation when you have defined a broad group of tracing operations, such as **all**.

```
[edit interfaces traceoptions]
user@host# set flag flag-option disable
```

You can specify the following flags in the **interfaces traceoptions** statement:

- **all**—Enable all configuration logging.
- **change-events**—Log changes that produce configuration events.
- **gres-events**—Log the events related to GRES.
- **resource-usage**—Log the resource usage for different states.
- **config-states**—Log the configuration state machine changes.
- **kernel**—Log configuration IPC messages to kernel.
- **kernel-detail**—Log details of configuration messages to kernel.
- **select-events**—Log the events on select state machine.

By default, interface process operations are placed in the file named dcd and three 1-MB files of tracing information are maintained.

For general information about tracing, see the tracing and logging information in the *Junos OS Administration Library*.

SEE ALSO

Tracing Interface Operations Overview | 254
Tracing Operations of an Individual Router Interface | 254
traceoptions

Tracing Operations of the pppd Process

You can trace the operations of the router’s pppd process.

To trace the router’s pppd process:

1. In configuration mode, go to the **[edit protocols ppp]** hierarchy level.

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

2. Include the **traceoptions** statement.

```
[edit protocols ppp]
```

```
traceoptions {
    file filename <files number> <match regular-expression> <size size> <world-readable | no-world-readable>;
    flag flag;
    level severity-level;
    no-remote-trace;
}
```

- To specify more than one tracing operation, include multiple **flag** statements.

  You can specify the following flags in the **traceoptions** statement:

  - **access**—Trace access code
  - **address-pool**—Trace address pool code
  - **all**—Trace all areas of code
  - **auth**—Trace authentication code
  - **chap**—Trace challenge handshake authentication protocol code
  - **ci**—Trace CI code
  - **config**—Trace configuration code
  - **ifdb**—Trace interface database code
  - **lcp**—Trace LCP state machine code
  - **memory**—Trace memory management code
  - **message**—Trace message processing code
  - **mlppp**—Trace multilink point-to-point protocol code
  - **ncp**—Trace NCP state machine code
  - **pap**—Trace password authentication protocol code
  - **ppp**—Trace PPP protocol processing code
  - **radius**—Trace RADIUS processing code
  - **redundancy**—Trace redundancy code
  - **rtsock**—Trace routing socket code
  - **session**—Trace session management code
  - **signal**—Trace signal handling code

- **timer**—Trace timer code
- **ui**—Trace user interface code

SEE ALSO

traceoptions | 395

Troubleshooting Interfaces

IN THIS SECTION

This topic discusses about various troubleshooting scenarios.

Troubleshooting: em0 Management Interface Link is Down

Problem

Description: Ethernet Link Down alarm is raised when you run the **show chassis alarm** operational mode command on a T640 router, a T1600 router, T4000 router, or a TX Matrix Plus router.

Diagnosis

Perform the following tests to check if the em0 management interface is down on the master Routing Engine or the backup Routing Engine:

1. Run the **show chassis alarms** command.

show chassis alarms

```
user@host0> show chassis alarms
1 alarms currently active
Alarm time Class Description
2011-10-19 11:13:02 MYT Major Host 1 em0 : Ethernet Link Down
```

Is the alarm **Ethernet Link Down** displayed against the em0 interface of the master Routing Engine (Host 0)?

- Yes:Contact JTAC for further assistance.
- No: Continue to the next diagnostic test.

2. Run the **show interfaces em0** and the **show interfaces em0 terse** operational mode commands.

   show interfaces em0

```
user@host> show interfaces em0
Physical interface: em0, Enabled, Physical link is Up
Interface index: 1, SNMP ifIndex: 1
Type: Ethernet, Link-level type: Ethernet, MTU: 1514, Speed: 100mbps
Device flags : Present Running
Interface flags: SNMP-Traps
...
```

   show interfaces em0 terse

```
user@host> show interfaces em0 terse
Interface Admin Link Proto Local Remote
em0 up up
em0.0 up up inet 10.100.100.1/30
```

   Is the em0 interface on the master Routing Engine **up**?

   - Yes:Continue to resolution.
   - No: Contact JTAC for further assistance

Resolution

To Resolve This Issue

From the aforementioned diagnosis, we ascertain that the chassis alarm has been raised for the em0 management interface in the backup Routing Engine (Host 1) and not for the master Routing Engine (Host 0).

Implement one of the following solutions on the backup Routing Engine to resolve this issue:

- Disable the em0 interface in the backup Routing Engine:

  1. In configuration mode, go to the **[edit groups re1]** hierarchy level.

     ```
     user@host1# edit groups re1
     ```

  2. Disable the em0 interface.

     ```
     [edit groups re1]
     user@host1# set interfaces em0 disable
     ```

- Ignore the alarm:

  1. In configuration mode, go to the **[edit chassis]** hierarchy level.

     ```
     user@host1# edit chassis
     ```

  2. Ignore the **Ethernet link down** alarm on the management interface by setting the **management-ethernet link-down** alarm option to **ignore**.

     ```
     [edit chassis]
     user@host1# set alarm management-ethernet link-down ignore
     ```

SEE ALSO

Supported Routing Engines by Router

show chassis alarms

Troubleshooting: fxp0 Management Interface Link is Down

Problem

Description: Ethernet Link Down alarm is raised when you run the **show chassis alarm** operational mode command on an M Series router, an MX Series router, a T320 router, a T640 router, a T1600 router, or on a TX Matrix router.

Diagnosis

Perform the following tests to check if the fxp0 interface is down on the master Routing Engine or the backup Routing Engine:

1. Run the **show chassis alarms** command.

   show chassis alarms

```
user@host0> show chassis alarms
1 alarms currently active
Alarm time Class Description
2011-10-19 11:13:02 MYT Major Host 1 fxp0 : Ethernet Link Down
```

   Is the alarm **Ethernet Link Down** displayed against the fxp0 interface of the master Routing Engine (Host 0)?

   - Yes:Contact JTAC for further assistance.
   - No: Continue to the next diagnostic test.

2. Run the **show interfaces fxp0** and the **show interfaces fxp0 terse** operational mode commands.

   show interfaces fxp0

```
user@host> show interfaces fxp0
Physical interface: fxp0, Enabled, Physical link is Up
Interface index: 1, SNMP ifIndex: 1
Type: Ethernet, Link-level type: Ethernet, MTU: 1514, Speed: 100mbps
Device flags : Present Running
Interface flags: SNMP-Traps
...
```

show interfaces fxp0 terse

```
user@host> show interfaces fxp0 terse
Interface Admin Link Proto Local Remote
fxp0 up up
fxp0.0 up up inet 10.100.100.1/30
```

Is the fxp0 interface on the master Routing Engine **up**?

- Yes:Continue to resolution.
- No: Contact JTAC for further assistance

Resolution

To Resolve This Issue

From the diagnosis, we ascertain that the chassis alarm has been raised for the fxp0 management interface in the backup Routing Engine (Host 1) and not for the master Routing Engine (Host 0).

Implement one of the following solutions on the backup Routing Engine to avoid this issue:

- Disable the fxp0 interface in the backup Routing Engine:
  1. In configuration mode, go to the **[edit groups re1]** hierarchy level.

     ```
     user@host1# edit groups re1
     ```

  2. Disable the fxp0 interface.

     ```
     [edit groups re1]
     user@host1# set interfaces fxp0 disable
     ```

- Ignore the alarm:
  1. In configuration mode, go to the **[edit chassis]** hierarchy level.

     ```
     user@host1# edit chassis
     ```

  2. Ignore the **Ethernet link down** alarm on the management interface by setting the **management-ethernet link-down** alarm option to **ignore**.

```
[edit chassis]
user@host1# set alarm management-ethernet link-down ignore
```

SEE ALSO

Supported Routing Engines by Router

show chassis alarms

Troubleshooting: Faulty Ethernet Physical Interface on an M Series, an MX Series, or a T Series Router

You can follow the basic troubleshooting checklist as explained in the following topics from one through five to troubleshoot an Ethernet physical interface on an M Series, MX Series, or a T Series router.

Checking the Cable Connection

Problem

Description: Packets are not received or transmitted over the Ethernet physical interface.

Diagnosis

1. Is the correct cable connected to the correct port?

    - Yes:Continue to "Checking the Physical Link Status of the Interface" on page 265.

    - No: See "Resolving Cabling Issue" on page 265.

Resolution

Resolving Cabling Issue

Perform one or more of the following steps to resolve the cabling issue:

1. Connect the cable properly on the local and remote ends without any loose connections.

2. Swap the Ethernet cable for a known good cable if the existing cable is damaged.

3. Connect a single-mode fiber cable to a single-mode interface only and a multimode fiber cable to a multimode interface only. To check fiber optic cable integrity, see "Checking Fiber Optic Cable Integrity" on page 265.

4. Connect the correct small form-factor pluggable transceiver (SFP) on both sides of the cable.

Checking Fiber Optic Cable Integrity

To check the integrity of fiber optic cable with an external cable diagnostic testing tool:

> **NOTE:** A single-mode fiber cable must be connected to a single-mode interface and a multi-mode fiber cable must be connected to a multi-mode interface.

1. Measure the received light level at the receiver (R_X) port to see whether the received light level is within the receiver specification of the Ethernet interface.

2. Measure transmitted light level at the transmitter (T_X) port to see whether the transmitted light level is within the transmitter specification of the Ethernet interface.

Checking the Physical Link Status of the Interface

Problem

Description: Unable to transmit and receive packets on the Ethernet interface even though the cable connection is correct.

Solution

To display the physical link status of the interface, run the **show interface *interface-name* media** operational mode command. For example, on the ge-5/0/1 interface.

```
user@host> show interfaces ge-5/0/1 media
Physical interface: ge-5/0/1, Enabled, Physical link is Up
  Interface index: 317, SNMP ifIndex: 1602
  Link-level type: Ethernet, MTU: 1514, Speed: 1000mbps, BPDU Error: None,
```

```
MAC-REWRITE Error: None, Loopback: Disabled,
  Source filtering: Disabled, Flow control: Enabled, Auto-negotiation: Enabled,
Remote fault: Online, Speed-negotiation: Disabled,
  Auto-MDIX: Enabled
  Device flags   : Present Running
  Interface flags: SNMP-Traps Internal: 0x4000
  Link flags     : None
  CoS queues     : 8 supported, 8 maximum usable queues
  Current address: 2c:6b:f5:4c:26:73, Hardware address: 2c:6b:f5:4c:26:73
  Last flapped   : 2012-11-30 01:25:37 UTC (03:46:55 ago)
  Input rate     : 880 bps (1 pps)
  Output rate    : 312 bps (0 pps)
  Active alarms  : None
  Active defects : None
  MAC statistics:
   Input bytes: 901296, Input packets: 9799, Output bytes: 976587, Output packets:
 10451
  Filter statistics:
    Filtered packets: 68, Padded packets: 0, Output packet errors: 0
  Autonegotiation information:
    Negotiation status: Complete
    Link partner:
        Link mode: Full-duplex, Flow control: Symmetric/Asymmetric, Remote fault:
 OK
    Local resolution:
        Flow control: Symmetric, Remote fault: Link OK
  Interface transmit statistics: Disabled
```

For information about **show interfaces *interface-name* media**, see *show interfaces* .

Diagnosis

1. Are there any connectivity problems such as input errors and packet loss even though the **Enabled** field displays **Physical link is Up** status and the **Active alarms and Active defect** field displays **None**?

   - Yes:Go to "Checking the Interface Statistics in Detail" on page 267.

   - No: Continue to the next diagnostic test.

2. Does the **Enabled** field display **Physical link is Down** status and the **Active alarms and Active defect** field display **Link**?

   - Yes:The interface is either not connected correctly or is not receiving a valid signal. Go to "Resolving Cabling Issue" on page 265.

- No: Continue.

Checking the Interface Statistics in Detail

Problem

Description: The physical interface is not working even though the **Enabled** field displays **Physical link is Up** status and the **Active alarms and Active defect** field displays **None**.

Solution

To display the interface statistics in detail, run the **show interface *interface-name* extensive** operational command. For example, on ge-5/0/1 interface.

```
user@host> show interfaces ge-5/0/1 extensive
Physical interface: ge-5/0/1, Enabled, Physical link is Up
  Interface index: 317, SNMP ifIndex: 1602, Generation: 322
  Link-level type: Ethernet, MTU: 1514, Speed: 1000mbps, BPDU Error: None,
MAC-REWRITE Error: None, Loopback: Disabled,
  Source filtering: Disabled, Flow control: Enabled, Auto-negotiation: Enabled,
Remote fault: Online, Speed-negotiation: Disabled,
  Auto-MDIX: Enabled
  Device flags   : Present Running
  Interface flags: SNMP-Traps Internal: 0x4000
  Link flags     : None
  CoS queues     : 8 supported, 8 maximum usable queues
  Hold-times     : Up 0 ms, Down 0 ms
  Current address: 2c:6b:f5:4c:26:73, Hardware address: 2c:6b:f5:4c:26:73
  Last flapped   : 2012-11-30 01:25:37 UTC (04:38:32 ago)
  Statistics last cleared: Never
  Traffic statistics:
   Input  bytes  :              806283                    0 bps
   Output bytes  :             1153215                  424 bps
   Input  packets:               10818                    0 pps
   Output packets:               11536                    0 pps
   IPv6 transit statistics:
    Input  bytes  :                  0
    Output bytes  :                  0
    Input  packets:                  0
    Output packets:                  0
  Label-switched interface (LSI) traffic statistics:
   Input  bytes  :                   0                    0 bps
   Input  packets:                   0                    0 pps
  Dropped traffic statistics due to STP State:
   Input  bytes  :                   0
```

```
    Output bytes  :                    0
    Input  packets:                    0
    Output packets:                    0
   Input errors:
     Errors: 0, Drops: 0, Framing errors: 0, Runts: 0, Policed discards: 233060,
 L3 incompletes: 0, L2 channel errors: 0,
     L2 mismatch timeouts: 0, FIFO errors: 0, Resource errors: 0
   Output errors:
     Carrier transitions: 11, Errors: 0, Drops: 0, Collisions: 0, Aged packets: 0,
  FIFO errors: 0, HS link CRC errors: 0,
     MTU errors: 0, Resource errors: 0
   Egress queues: 8 supported, 4 in use
   Queue counters:       Queued packets  Transmitted packets      Dropped packets
     0 best-effort                3216                 3216                    0
     1 expedited-fo                  0                    0                    0
     2 assured-forw                  0                    0                    0
     3 network-cont               8320                 8320                    0
   Queue number:         Mapped forwarding classes
     0                   best-effort
     1                   expedited-forwarding
     2                   assured-forwarding
     3                   network-control
   Active alarms  : None
   Active defects : None
   MAC statistics:                      Receive         Transmit
     Total octets                       1007655          1082219
     Total packets                        10886            11536
     Unicast packets                       4350             4184
     Broadcast packets                       32               77
     Multicast packets                     6504             7275
     CRC/Align errors                         0                0
     FIFO errors                              0                0
     MAC control frames                       0                0
     MAC pause frames                         0                0
     Oversized frames                         0
     Jabber frames                            0
     Fragment frames                          0
     VLAN tagged frames                       0
     Code violations                          0
   Filter statistics:
     Input packet count                   10886
     Input packet rejects                    68
     Input DA rejects                        68
     Input SA rejects                         0
```

```
    Output packet count                                   11536
    Output packet pad count                                   0
    Output packet error count                                 0
    CAM destination filters: 0, CAM source filters: 0
  Autonegotiation information:
    Negotiation status: Complete
    Link partner:
        Link mode: Full-duplex, Flow control: Symmetric/Asymmetric, Remote fault:
 OK
    Local resolution:
        Flow control: Symmetric, Remote fault: Link OK
  Packet Forwarding Engine configuration:
    Destination slot: 5
  CoS information:
    Direction : Output
    CoS transmit queue               Bandwidth               Buffer Priority
Limit
                              %            bps     %            usec
    0 best-effort            95      950000000    95               0       low
none
    3 network-control         5       50000000     5               0       low
none
  Interface transmit statistics: Disabled
```

For information about **show interfaces *interface-name* detail**, see *show interfaces* .

Diagnosis

1. Does the **Policed discards**, **L2 channel errors**, **Input DA rejects**, or the **Input SA rejects** field display any errors?

   For information about the errors, see*show interfaces* .

   - Yes:Resolve the errors as needed. Resolving these errors is beyond the scope of this topic.
   - No: Continue with "Performing the Loopback Diagnostic Test" on page 269.

Performing the Loopback Diagnostic Test

Problem

Description: The interface cable is connected correctly and there are no alarms or errors associated with the Ethernet physical interface yet the interface is not working.

Solution

To check whether the Ethernet port or PIC is faulty, you must perform the internal loopback test and hardware loopback test.

To perform a internal loopback diagnostic test on an Ethernet interface, for example on ge-5/0/1 interface:

1. In configuration mode, go to the **[edit interfaces *ge-5/0/1*]** hierarchy level.

```
[edit]
user@host# edit interface ge-5/0/1
```

2. Set the **gigether-options** option as loopback, commit the configuration and quit configuration mode.

```
[edit interfaces ge-5/0/1
user@host# set gigether-options loopback
user@host# commit
user@host# quit
```

3. In operational mode, execute the **show interfaces *ge-5/0/1* media** command.

```
user@host> show interfaces ge-5/0/1 media
Physical interface: ge-5/0/1, Enabled, Physical link is Up
  Interface index: 317, SNMP ifIndex: 1602
  Link-level type: Ethernet, MTU: 1514, Speed: 1000mbps, BPDU Error: None,
MAC-REWRITE Error: None, Loopback: Enabled,
  Source filtering: Disabled, Flow control: Enabled, Auto-negotiation: Enabled,
 Remote fault: Online, Speed-negotiation: Disabled,
  Auto-MDIX: Enabled
  Device flags   : Present Running
  Interface flags: SNMP-Traps Internal: 0x4000
  Link flags     : None
  CoS queues     : 8 supported, 8 maximum usable queues
  Current address: 2c:6b:f5:4c:26:73, Hardware address: 2c:6b:f5:4c:26:73
  Last flapped   : 2012-11-30 01:25:37 UTC (03:46:55 ago)
  Input rate     : 880 bps (1 pps)
  Output rate    : 312 bps (0 pps)
  Active alarms  : None
  Active defects : None
  MAC statistics:
    Input bytes: 901296, Input packets: 9799, Output bytes: 976587, Output
packets: 10451
  Filter statistics:
    Filtered packets: 68, Padded packets: 0, Output packet errors: 0
  Autonegotiation information:
```

```
    Negotiation status: Complete
    Link partner:
       Link mode: Full-duplex, Flow control: Symmetric/Asymmetric, Remote fault:
OK
    Local resolution:
        Flow control: Symmetric, Remote fault: Link OK
  Interface transmit statistics: Disabled
```

NOTE: Delete the **loopback** statement after completing your diagnosis.

Execute one of the following steps for a hardware loopback diagnostic test as needed:

- For an Ethernet PIC with a fiber optic interface—Physically loop the T_x and R_x port and check the status of the physical link with the **show interfaces *interface-name* media** operational mode command.

- For an Ethernet PIC with an RJ-45 Ethernet interface—Build a loopback plug by crossing pin 1 (T_x +) to pin 3 (R_x +) together and pin 2 (T_x -) and pin 6 (R_x -) together and check the status of the physical link with the **show interfaces *interface-name* media** operational mode command.

NOTE: For information about loopback testing, see *Performing Loopback Testing for Fast Ethernet and Gigabit Ethernet Interfaces*.

Diagnosis

1. Does the **Enabled** field display **Physical link is Up** status and the **Active alarms and Active defect** field display **None** when you perform the loopback test?

   - Yes:Go to the "Checking Other Possibilities" on page 272 section.

   - No: Continue to the next diagnostic test.

2. When the Ethernet interface is connected to a remote Ethernet device over multiple patch panels, check to see whether the connection can be looped back at the different patch panels so you can conduct a loopback diagnostic test. Is the loopback diagnostic test successful?

   - Yes:Go to the "Checking Other Possibilities" on page 272 section.

   - No: Contact JTAC for further assistance.

Checking Other Possibilities

Problem

Description: Loopback diagnostic test is successful but unable to transmit and receive packets on the Ethernet interface.

Solution

Use the following commands as needed to troubleshoot an Ethernet interface, for example, an ge-5/0/1 interface:

- Run the **show interfaces *interface-name* terse** operational command to check if the physical interface and logical interfaces are administratively disabled. For example, on ge-5/0/1 interface.

```
user@host> show interfaces ge-5/0/1 terse
     Interface               Admin Link Proto    Local                 Remote
     ge-5/0/1                up    up
     ge-5/0/1.0              up    up   inet     20.1.1.2/24
```

Diagnosis

1. Does the physical interface and its corresponding logical interfaces display **down** in the output of the **show interfaces *interface-name* terse** operational mode command?

   - Yes: Enable the interfaces as shown in "To Enable a Physical Interface" on page 273.
   - No: Continue to the next diagnostic test.

2. Are the **speed**, **duplex**, and **auto-negotiation** fields in the output of **show interfaces *interface-name* extensive** operational mode command correctly set for the interface?

   > **NOTE:** Check if the associated Flexible PIC Concentrator (FPC), Modular Port Concentrator (MPC), or Dense Port Concentrator (DPC) and its Modular Interface Card (MIC) or PIC with its 10-gigabit small form-factor pluggable transceiver (XFP) or SFP supports speed and auto-negotiation settings.

   - Yes: Check *Monitoring Fast Ethernet and Gigabit Ethernet Interfaces* for more troubleshooting tips.
   - No: Contact JTAC for further assistance.

To Enable a Physical Interface

To enable a physical interface:

1. In configuration mode, go to the **[edit interfaces]** hierarchy level.

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

2. Check if the interface is administratively disabled by executing the **show** command on the interface. For example on ge-5/0/1 interface.

   ```
   user@host# show ge-5/0/1
   ```

   ```
   disable;
   ```

3. Enable the interface and commit.

   ```
   [edit interfaces
   user@host# delete interface-name disable
   user@host# commit
   ```

SEE ALSO

show interfaces

Time Domain Reflectometry on ACX Series Routers Overview

Time Domain Reflectometry (TDR) is a technology used for diagnosing copper cable states. This technique can be used to determine if cabling is at fault when you cannot establish a link. TDR detects the defects by sending a signal through a cable, and reflecting it from the end of the cable. Open circuits, short circuits, sharp bends and other defects in the cable, reflects the signal back, at different amplitudes, depending on the severity of the defect.

Several factors that result in degraded or low-quality cable plants can cause packet loss, suboptimal connection speed, reduced network efficiency, and complete connection failures. These types of problems can occur because of poor cable construction, identification of pair twists, loose connectors, poor contacts between the points, and stretched or broken pairs of cables. Broadcom transceivers enable you to analyze the condition of the cable plant or topology and identify any problems that have occurred. This functionality is effectively used in the following scenarios:

- Troubleshooting during initial network equipment installation.
- Discovery of failures when network problems occur.
- Maintenance of optimally functioning cable plants.
- Fault determination during the testing of network equipment in production cable networks.

TDR supports the following capabilities for examination of cable faults on ACX Series routers:

- Cable status pair (open or short)—When the router operates in Gigabit Ethernet mode, all the four pairs (8 wires) are used. Only Pair-A and Pair-B are required to operate in 10/100BASE-T Ethernet mode. If either of these required pairs is open or short-circuited, the transceiver reports the following faults:
  - Any open wire
  - Wires of a particular pair that are shorted
- Distance to fault per pair—Distance at which an open or a short-circuit is detected in meters. This measurement is also termed as cable length. The transceiver reports the following faults:
  - Cable length when the cable status is normal
  - Distance to fault when the cable status is not normal
- Pair Swap—Swapping of twisted-pairs in straight-through and cross-over cable plants are detected.
- Polarity Swap—Each cable pair carries a differential signal from one end to the other end of the cable. Each wire within the pair is assigned a polarity. The wires in a pair are normally connected in a one-to-one form. This connection enables the transmitter at one end to be connected to the receiver at the other end with same polarity. Sometimes, the wiring within the pair is also swapped. This type of connection is called polarity swap. Broadcom transceivers can detect such swapping and automatically adjust the connection to enable the links to operate normally. However, the transceiver reports polarity swaps that it detects in the cable plant.

On 4-port Gigabit Ethernet and 8-port Gigabit Ethernet MICs with copper SFP transceivers (using BCM54880) and 4-port Gigabit Ethernet, 6-port Gigabit Ethernet, and 8-port Gigabit Ethernet MICs with copper and optical SFP transceivers (using BCM54640E PHY), only 10BASE-T pair polarity is supported. 100BASE-T and 1000BASE-T polarities are not supported.

When the Gigabit Ethernet link cannot be established (for example, if only two pairs are present that are fully functional), TDR in the physical layer (PHY) brings down the link to a 100 MB link, which is called a downshift in the link. The physical layer might require 10-20 seconds for the link to come up if a downgrade in wire speed occurs because it attempts to connect at 1000 MB five times before it falls back to 100BASE-TX.

TDR diagnostics is supported only on copper interfaces and not on fiber interfaces.

Keep the following points in mind when you configure TDR:

- If you connect a port undergoing a TDR test to a Gigabit Ethernet interface that is enabled to automatically detect MDI (Media Dependent Interface) and MDIX (Media Dependent Interface with Crossover) port connections, the TDR result might be invalid.
- If you connect a port undergoing a TDR test to a 100BASE-T copper interface, the unused pairs are reported as faulty because the remote end does not terminate these pairs.
- You must not modify the port configuration while the TDR test is running.
- Because of cable characteristics, you need to run the TDR test multiple times to get accurate results.
- Do not change the port status (such as removing the cable at the near or far end) because such a change can result in inaccurate statistics in the results.
- While measuring the cable length or distance to fault (per pair), sometimes, a few cable length inconsistencies might be observed during a TDR test. Broadcom transceivers have the following cable length limitations:
  - For a properly-terminated good cable, the accuracy of the cable length reported is plus or minus 10 meters.
  - If a pair is open or short-circuited, the far-end termination does not affect the computed result for that pair.
  - The accuracy of the measured cable length, when open and short-circuit conditions are detected, is plus or minus 5 meters.
  - The accuracy of a good pair, when one or more pairs are open or short-circuited, is plus or minus 10 meters.
- Polarity swap detection is supported only in 10BASE-T mode.
- The TDR test does not impact the traffic if the interface operates at 10-Gigabit Ethernet per second of bandwidth, which is the default configuration. However, if the speed of the interface is configured to be other than 10-Gigabit Ethernet, running the TDR test affects the traffic.

  TDR diagnostics might bring the link down and initialize the physical layer (PHY) with default configuration to perform its operation.

  When the TDR validation test is completed, the PHY layer resumes operation in the same manner as before the cable diagnostics test was performed. However, link flaps might be momentarily observed. We recommend that you run the TDR test at a speed of 1 gigabit per second, which is the default configuration, to obtain more accurate results.

TDR is supported on the following interfaces on ACX Series routers:

- On ACX1000 routers, 4 RJ45 (Cu) ports or 8-port Gigabit Ethernet MICs with small form-factor pluggable (SFP) transceivers and RJ45 connectors.

  On ACX1100 routers, 4-port or 8-port Gigabit Ethernet MICs with SFP transceivers and RJ45 connectors.
- On ACX2000 routers, 8-port Gigabit Ethernet MICs with SFP transceivers and RJ45 connectors.
- On ACX2100 and ACX2200 routers, 4-port Gigabit Ethernet MICs with SFP transceivers and RJ45 connectors.
- On ACX4000 routers, 4-port, 6-port, or 8-port Gigabit Ethernet MICs with SFP transceivers and RJ45 connectors.

You must select the media type as copper for the 1-Gigabit Ethernet interfaces. To specify the media type, include the **media-type** statement with the **copper** option at the **[edit interfaces *interface-name*]** hierarchy level. Media type selection is applicable to ports only in slot 2. When media-type is not set, the port accepts either type of connection. The media type is fiber if a transceiver is installed in the SFP connection. If no transceiver is installed, the media type is copper. The COMBO ports (combination ports) on ACX routers support both the copper and fiber-optic media types. On such ports or interfaces, you must configure the media type as copper to run the TDR test.

You can run the TDR test from operational mode and view the success or failure results of the test. To start a test on a specific interface, issue the **request diagnostics tdr start interface *interface-name*** command. To stop the TDR test currently in progress on the specified interface, issue the **request diagnostics tdr abort interface *interface-name*** command. To display the test results for all copper interfaces, enter the **show diagnostics tdr** command. To display the test results for a particular interface, enter the **show diagnostics tdr interface interface-name** command.

SEE ALSO

Diagnosing a Faulty Twisted-Pair Cable on ACX Series Routers | 276

Diagnosing a Faulty Twisted-Pair Cable on ACX Series Routers

Problem

Description: A 10/100BASE-T Ethernet interface has connectivity problems that you suspect might be caused by a faulty cable.

Solution

Use the time domain reflectometry (TDR) test to determine whether a twisted-pair Ethernet cable is faulty.

The TDR test:

- Detects and reports faults for each twisted pair in an Ethernet cable. Faults detected include open circuits, short circuits, and impedance mismatches.
- Reports the distance to fault to within 1 meter.
- Detects and reports pair swaps, pair polarity reversals, and excessive pair skew.

The TDR test is supported on the following ACX routers and interfaces:

- On ACX1000 routers, 4 RJ45 (Cu) ports or 8-port Gigabit Ethernet MICs with small form-factor pluggable (SFP) transceivers and RJ45 connectors.
- On ACX1100 routers, 4-port or 8-port Gigabit Ethernet MICs with SFP transceivers and RJ45 connectors.
- On ACX2000 routers, 8-port Gigabit Ethernet MICs with SFP transceivers and RJ45 connectors.
- On ACX2100 and ACX2200 routers, 4-port Gigabit Ethernet MICs with SFP transceivers and RJ45 connectors.
- On ACX4000 routers, 4-port, 6-port, or 8-port Gigabit Ethernet MICs with SFP transceivers and RJ45 connectors.

> **NOTE:** We recommend running the TDR test on an interface when there is no traffic on the interface.
>
> TDR diagnostics are applicable for copper ports only and not for optical fiber ports.

To diagnose a cable problem by running the TDR test:

1. Run the **request diagnostics tdr** command.

```
user@host> request diagnostics tdr start interface ge-0/0/10

Interface TDR detail:
Test status                     : Test successfully executed  ge-0/0/10
```

2. View the results of the TDR test with the **show diagnostics tdr** command.

```
user@host> show diagnostics tdr interface ge-0/0/10

Interface TDR detail:
Interface name                  : ge-0/0/10
Test status                     : Passed
Link status                     : Down
MDI pair                        : 1-2
```

```
    Cable status                  : Normal
    Distance fault                : 0 Meters
    Polartiy swap                 : N/A
    Skew time                     : N/A
  MDI pair                        : 3-6
    Cable status                  : Normal
    Distance fault                : 0 Meters
    Polartiy swap                 : N/A
    Skew time                     : N/A
  MDI pair                        : 4-5
    Cable status                  : Open
    Distance fault                : 1 Meters
    Polartiy swap                 : N/A
    Skew time                     : N/A
  MDI pair                        : 7-8
    Cable status                  : Normal
    Distance fault                : 0 Meters
    Polartiy swap                 : N/A
    Skew time                     : N/A
  Channel pair                    : 1
    Pair swap                     : N/A
  Channel pair                    : 2
    Pair swap                     : N/A
  Downshift                       : N/A
```

3. Examine the **Cable status** field for the four MDI pairs to determine if the cable has a fault. In the preceding example, the twisted pair on pins 4 and 5 is broken or cut at approximately one meter from the **ge-0/0/10** port connection.

> NOTE: The **Test Status** field indicates the status of the TDR test, not the cable. The value **Passed** means the test completed—it does not mean that the cable has no faults.

The following is additional information about the TDR test:

- The TDR test can take some seconds to complete. If the test is still running when you execute the **show diagnostics tdr** command, the **Test status** field displays **Started**. For example:

```
user@host> show diagnostics tdr interface ge-0/0/22

Interface TDR detail:
```

```
Interface name                    : ge-0/0/22
Test status                       : Started
```

- You can terminate a running TDR test before it completes by using the **request diagnostics tdr abort interface *interface-name*** command. The test terminates with no results, and the results from any previous test are cleared.
- You can display summary information about the last TDR test results for all interfaces on the router that support the TDR test by not specifying an interface name with the **show diagnostics tdr** command. For example:

```
user@host> show diagnostics tdr
Interface    Test status    Link status  Cable status  Max distance fault
 ge-0/0/0         Passed      UP           OK              0
 ge-0/0/1    Not Started    N/A          N/A           N/A
 ge-0/0/2         Passed      UP           OK              0
 ge-0/0/3    Not Started    N/A          N/A           N/A
 ge-0/0/4         Passed      UP           OK              0
 ge-0/0/5         Passed      UP           OK              0
 ge-0/0/6         Passed      UP           OK              0
 ge-0/0/7    Not Started    N/A          N/A           N/A
 ge-0/0/8         Passed    Down           OK              0
 ge-0/0/9    Not Started    N/A          N/A           N/A
ge-0/0/10         Passed    Down         Fault             1
ge-0/0/11         Passed      UP           OK              0
ge-0/0/12    Not Started    N/A          N/A           N/A
ge-0/0/13    Not Started    N/A          N/A           N/A
ge-0/0/14    Not Started    N/A          N/A           N/A
ge-0/0/15    Not Started    N/A          N/A           N/A
ge-0/0/16    Not Started    N/A          N/A           N/A
ge-0/0/17    Not Started    N/A          N/A           N/A
ge-0/0/18    Not Started    N/A          N/A           N/A
ge-0/0/19         Passed    Down           OK              0
ge-0/0/20    Not Started    N/A          N/A           N/A
ge-0/0/21    Not Started    N/A          N/A           N/A
ge-0/0/22         Passed      UP           OK              0
ge-0/0/23    Not Started    N/A          N/A           N/A
```

SEE ALSO

Time Domain Reflectometry on ACX Series Routers Overview | 273

request diagnostics tdr

show diagnostics tdr

4

CHAPTER

Configuration Statements

accounting

Syntax

```
accounting {
  destination-class-usage;
  source-class-usage {
    direction;
  }
}
```

Hierarchy Level

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

Release Information

Statement introduced before Junos OS Release 7.4.

Description

Enable IP packet counters on an interface.

The remaining statements are explained separately. See CLI Explorer.

Required Privilege Level

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

RELATED DOCUMENTATION

Enabling Source Class and Destination Class Usage | 193

accounting-profile

Syntax

```
accounting-profile name;
```

Hierarchy Level

```
[edit interfaces interface-name],
[edit interfaces interface-name unit logical-unit-number],
[edit interfaces interface-range 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 15.1F6 for PTX Series routers with third-generation FPCs installed.

Description

Enable collection of accounting data for the specified physical or logical interface or interface range.

Options

name—Name of the accounting profile.

Required Privilege Level

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

RELATED DOCUMENTATION

Applying an Accounting Profile to the Physical Interface | 145
Applying an Accounting Profile to the Logical Interface | 159

acfc

Syntax

```
acfc;
```

Hierarchy Level

[edit interfaces *interface-name* ppp-options compression],
[edit interfaces *interface-name* unit *logical-unit-number* ppp-options compression],
[edit logical-systems *logical-system-name* interfaces *interface-name* unit *logical-unit-number* ppp-options compression]

Release Information

Statement introduced before Junos OS Release 7.4.

Description

For interfaces with PPP encapsulation, configure compression of the Data Link Layer address and control fields. The **acfc** option is not supported with **frame-relay-ppp** encapsulation.

On M320, M120, and T Series routers, address and control field compression (ACFC) is not supported for any ISO family protocols. Do not include the **acfc** statement at the **[edit interfaces *interface-name* ppp-options compression]** hierarchy level when you include the **family iso** statement at the **[edit interfaces *interface-name* unit *logical-unit-number*]** hierarchy level.

Required Privilege Level

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

RELATED DOCUMENTATION

Configuring PPP Address and Control Field Compression

action (Policer)

Syntax

```
action {
    loss-priority high then discard;
}
```

Hierarchy Level

```
[edit firewall three-color-policer policer-name]
```

Release Information

Statement introduced in Junos OS Release 8.2.

Description

This statement discards high loss priority traffic as part of a configuration using tricolor marking on a logical interface.

Required Privilege Level

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

RELATED DOCUMENTATION

Class of Service User Guide (Routers and EX9200 Switches)

logical-interface-policer

activation-delay

Syntax

```
activation-delay seconds;
```

Hierarchy Level

```
[edit interfaces dln unit logical-unit-number dialer-options]
```

Release Information

Statement introduced before Junos OS Release 7.4.

Description

(J Series Services Routers) For ISDN interfaces, configure the ISDN dialer activation delay. Used only for dialer backup and dialer watch cases.

Options

seconds—Interval before the backup interface is activated after the primary interface has gone down.

Range: 1 through 4,294,967,295 seconds

Required Privilege Level

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

RELATED DOCUMENTATION

Junos OS Interfaces and Routing Configuration Guide

activation-priority

Syntax

```
activation-priority priority;
```

Hierarchy Level

```
[edit interfaces interface-name unit logical-unit-number dynamic-call-admission-control],
[edit logical-systems logical-system-name interfaces interface-name unit logical-unit-number
   dynamic-call-admission-control]
```

Release Information

Statement introduced in Junos OS Release 8.2.

Description

(J4350 and J6350 Services Routers supporting voice over IP with the TGM550 media gateway module) For Fast Ethernet and Gigabit Ethernet interfaces, ISDN BRI interfaces, and serial interfaces with PPP or Frame Relay encapsulation, configure the dynamic call admission control (dynamic CAC) activation priority value.

Options

priority—The activation priority in which the interface is used for providing call bandwidth. The interface with the highest activation priority value is used as the primary link for providing call bandwidth. If the primary link becomes unavailable, the TGM550 switches over to the next active interface with the highest activation priority value, and so on.

Range: 0 through 255

Default: 50

Required Privilege Level

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

RELATED DOCUMENTATION

Junos OS Services Interfaces Library for Routing Devices

Junos OS Interfaces and Routing Configuration Guide

aggregate (Hierarchical Policer)

Syntax

```
aggregate {
  if-exceeding {
    bandwidth-limit bandwidth;
    burst-size-limit burst;
  }
  then {
    discard;
  }
}
```

Hierarchy Level

```
[edit firewall hierarchical-policer]
```

Release Information

Statement introduced in Junos OS Release 9.5.

Description

On M40e, M120, and M320 (with FFPC and SFPC) edge routers and T320, T640, and T1600 core routers with Enhanced Intelligent Queuing (IQE) PICs, T4000 routers with Type 5 FPC and Enhanced Scaling Type 4 FPC, configure an aggregate hierarchical policer.

Options

Options are described separately.

Required Privilege Level

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

RELATED DOCUMENTATION

Applying Policers

Class of Service User Guide (Routers and EX9200 Switches)

alias (Interfaces)

Syntax

```
alias alias-name;
```

Hierarchy Level

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

Release Information

Statement introduced in Junos OS Release 13.3.

Description

Configure a textual description of a physical interface or the logical unit of an interface to be the alias of an interface name. The alias name can be a single line of text. If the text contains spaces, enclose it in quotation marks. If you configure an alias name, the alias name is displayed instead of the interface name in the output of all **show**, **show interfaces**, and other operational mode commands. In Junos OS Release 12.3R8 and later, display of the alias can be suppressed in favor of the actual interface name by using the **display no-interface-alias** parameter along with the show command.

Options

alias-name—Text to denote an easily identifiable, meaningful alias name for the interface. If the text includes spaces, enclose the entire text in quotation marks.

Required Privilege Level

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

RELATED DOCUMENTATION

Example: Adding an Interface Alias Name | 112

Junos OS Network Interfaces Library for Routing Devices

backup-options

Syntax

```
backup-options {
    interface interface-name;
}
```

Hierarchy Level

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

Release Information

Statement introduced before Junos OS Release 7.4.

Description

Configure an interface to be used as a backup interface if the primary interface goes down. This is used to support ISDN dial backup operation.

The remaining statement is explained separately. See CLI Explorer.

Required Privilege Level

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

RELATED DOCUMENTATION

Junos OS Interfaces and Routing Configuration Guide

callback

Syntax

```
callback;
```

Hierarchy Level

[edit interfaces dl*n* unit *logical-unit-number* dialer-options **incoming-map**],
[edit logical-systems *logical-system-name* interfaces dl*n* unit *logical-unit-number* dialer-options **incoming-map**]

Release Information

Statement introduced in Junos OS Release 7.5.

Description

On J Series Services Routers with interfaces configured for ISDN, configure the dialer to terminate the incoming call and call back the originator after the callback wait period. The default wait time is 5 seconds. To configure the wait time, include the **callback-wait-period** statement at the **[edit interfaces dl *n* unit *logical-unit-number* dialer-options]** hierarchy level.

> NOTE: The **incoming-map** statement is mandatory for the router to accept any incoming ISDN calls.

If the **callback** statement is configured, you cannot use the **caller *caller-id*** statement at the **[edit interfaces dl*n* unit *logical-unit-number* dialer-options]** hierarchy level.

Required Privilege Level

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

RELATED DOCUMENTATION

Junos OS Interfaces and Routing Configuration Guide

callback-wait-period | 296

callback-wait-period

Syntax

```
callback-wait-period time;
```

Hierarchy Level

```
[edit interfaces dln unit logical-unit-number dialer-options],
[edit logical-systems logical-system-name interfaces dln unit logical-unit-number dialer-options]
```

Release Information

Statement introduced in Junos OS Release 7.5.

Description

On J Series Services Routers with interfaces configured for ISDN with callback, specify the amount of time the dialer waits before calling back the caller. The default wait time is 5 seconds. The wait time is necessary because, when a call is rejected, the switch waits for up to 4 seconds on point-to-multipoint connections to ensure no other device accepts the call before sending the DISCONNECT message to the originator of the call. However, the default time of 5 seconds may not be sufficient for different switches or may not be needed on point-to-point connections.

To configure callback mode, include the **callback** statement at the **[edit interfaces dl*n* unit *logical-unit-number* dialer-options]** hierarchy level.

If the **callback** statement is configured, you cannot use the **caller *caller-id*** statement at the **[edit interfaces dl*n* unit *logical-unit-number* dialer-options]** hierarchy level.

Options

time—Time the dialer waits before calling back the caller.

Required Privilege Level

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

RELATED DOCUMENTATION

Junos OS Interfaces and Routing Configuration Guide

caller

Syntax

```
caller (caller-id | accept-all);
```

Hierarchy Level

```
[edit interfaces dln unit logical-unit-number dialer-options incoming-map],
[edit logical-systems logical-system-name interfaces dln unit logical-unit-number dialer-options incoming-map]
```

Release Information

Statement introduced in Junos OS Release 7.5.

Description

On J Series Services Routers with interfaces configured for ISDN, specify the dialer to accept a specified caller number or accept all incoming calls.

Options

caller-id—Incoming caller number. You can configure multiple caller IDs on a dialer. The caller ID of the incoming call is matched against all caller IDs configured on all dialers. The dialer matching the caller ID is looked at for further processing. Only a precise match is a valid match, For example, the configured caller ID 1-222-333-4444 or 222-333-4444 will match the incoming caller ID 1-222-333-4444.

If the incoming caller ID has fewer digits than the number configured, it is not a valid match. Duplicate caller IDs are not allowed on different dialers; however, for example, the numbers 1-408-532-1091, 408-532-1091, and 532-1091 can still be configured on different dialers.

Only one B-channel can map to one dialer. If one dialer is already mapped, any other call mapping to the same dialer is rejected (except in the case of a multilink dialer). If no dialer caller is configured on a dialer, that dialer will not accept any calls.

accept-all—Any incoming call in an associated interface is accepted.

Required Privilege Level

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

RELATED DOCUMENTATION

Junos OS Interfaces and Routing Configuration Guide

calling-number

Syntax

```
calling-number number;
```

Hierarchy Level

```
[edit interfaces br-pim/0/port isdn-options]
```

Release Information

Statement introduced before Junos OS Release 7.4.

Description

On J Series Services Routers with ISDN interfaces, configure the calling number to include in outgoing calls.

Options

number—Calling number.

Required Privilege Level

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

RELATED DOCUMENTATION

Configuring ISDN Physical Interface Properties

Junos OS Interfaces and Routing Configuration Guide

clock-rate

Syntax

```
clock-rate rate;
```

Hierarchy Level

```
[edit interfaces interface-name serial-options]
```

Release Information

Statement introduced before Junos OS Release 7.4.

Description

For EIA-530 and V.35 interfaces, configure the interface speed, in megahertz (MHz).

Options

rate—You can specify one of the following rates:

- 2.048 MHz
- 2.341 MHz
- 2.731 MHz
- 3.277 MHz
- 4.096 MHz
- 5.461 MHz
- 8.192 MHz
- 16.384 MHz

Default: 16.384 MHz

Required Privilege Level

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

RELATED DOCUMENTATION

Configuring the Serial Clocking Mode | 242

clocking-mode

Syntax

```
clocking-mode (dce | internal | loop);
```

Hierarchy Level

```
[edit interfaces interface-name serial-options]
```

Release Information

Statement introduced before Junos OS Release 7.4.

Description

For EIA-530 and V.35 interfaces, configure the clock mode. You cannot configure **clocking-mode dce** on a DTE router using an X.21 serial line protocol (detected automatically when an X.21 cable is plugged into the serial interface).

Options

dce—DCE timing (DTE mode only, not valid for X.21).

internal—Internal baud timing.

loop—Loop timing.

Default: loop

Required Privilege Level

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

RELATED DOCUMENTATION

Configuring the Serial Clocking Mode | 242

control-polarity

Syntax

```
control-polarity (negative | positive);
```

Hierarchy Level

```
[edit interfaces interface-name serial-options]
```

Release Information

Statement introduced before Junos OS Release 7.4.

Description

For X.21 interfaces only, configure the control signal polarity.

Options

positive—Positive signal polarity.

negative—Negative signal polarity.

Default: positive

Required Privilege Level

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

RELATED DOCUMENTATION

Configuring Serial Signal Polarities | 249

control-signal

Syntax

```
control-signal (assert | de-assert | normal);
```

Hierarchy Level

```
[edit interfaces interface-name serial-options dce-options],
[edit interfaces interface-name serial-options dte-options]
```

Release Information

Statement introduced before Junos OS Release 7.4.

Description

For X.21 interfaces only, configure the to-DCE signal.

Options

assert—The to-DCE signal must be asserted.

de-assert—The to-DCE signal must be deasserted.

normal—Normal request-to-send (RTS) signal handling, as defined by ITU-T Recommendation X.21.

Default: normal

Required Privilege Level

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

RELATED DOCUMENTATION

Configuring the Serial Signal Handling | 245

cts

Syntax

```
cts (ignore | normal | require);
```

Hierarchy Level

[edit interfaces *interface-name* **serial-options dce-options**],
[edit interfaces *interface-name* **serial-options dte-options**]

Release Information

Statement introduced before Junos OS Release 7.4.

Description

For EIA-530 and V.35 interfaces only, configure the from-DCE signal, clear-to-send (CTS).

Options

ignore—The from-DCE signal is ignored.

normal—Normal CTS signal handling as defined by the TIA/EIA Standard 530.

require—The from-DCE signal must be asserted.

Default: normal

Required Privilege Level

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

RELATED DOCUMENTATION

Configuring the Serial Signal Handling | 245

cts-polarity

Syntax

```
cts-polarity (negative | positive);
```

Hierarchy Level

[edit interfaces *interface-name* **serial-options**]

Release Information

Statement introduced before Junos OS Release 7.4.

Description

Configure CTS signal polarity.

Options

positive—Positive signal polarity.

negative—Negative signal polarity.

Default: positive

Required Privilege Level

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

RELATED DOCUMENTATION

Configuring Serial Signal Polarities | 249

damping (Interfaces)

Syntax

```
damping {
    enable;
    half-life seconds;
    max-suppress seconds;
    reuse number;
    suppress number;
}
```

Hierarchy Level

```
[edit interfaces interface--name],
[edit interfaces interface--range]
```

Release Information

Statement introduced in Junos OS Release 14.1 for PTX Series Packet Transport Routers and T Series Core Routers.
Statement introduced in Junos OS Release 14.2 for MX960, MX480, MX240, and MX80 Universal Routing Platforms and M10i Multiservice Edge Routers.

Description

Limit the number of advertisements of the up and down transitions (flapping) on an interface. Each time a transition occurs, the interface state is changed, which generates an advertisement to the upper-level routing protocols. Damping helps reduce the number of these advertisements. Every time an interface goes down, a penalty is added to the interface penalty counter. Penalty added on every interface flap is 1000.

If at some point the accumulated penalty exceeds the suppress level **max-suppress**, the interface is placed in the suppress state, and further interface state up and down transitions are not reported to the upper-level protocols.

Options

enable—Enable damping on a per-interface basis. If damping is enabled on an interface, it is suppressed during interface flaps that match the configuration settings.

Default: Disabled

half-life *seconds*—Decay half-life. ***seconds*** is the interval after which the accumulated interface penalty counter is reduced by half if the interface remains stable.

> **NOTE:** For the half-life, configure a value that is less than the max-suppress value. If you do not, the configuration is rejected.

Range: 1 through 30

Default: 5

max-suppress *seconds*—Maximum hold-down time. ***seconds*** is the maximum time that an interface can be suppressed no matter how unstable the interface has been.

> **NOTE:** For max-suppress, configure a value that is greater than the half-life. If you do not, the configuration is rejected.

Range: 1 through 20,000

Default: 20

reuse *number*—Reuse threshold. When the accumulated interface penalty counter falls below *number*, the interface is no longer suppressed.

Range: 1 through 20,000

Default: 1000

suppress *number*—Cutoff (suppression) threshold. When the accumulated interface penalty counter exceeds *number*, the interface is suppressed.

Range: 1 through 20,000

Default: 2000

Required Privilege Level

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

RELATED DOCUMENTATION

Physical Interface Damping Overview | 130
Damping Shorter Physical Interface Transitions | 137
Damping Longer Physical Interface Transitions | 139
show interfaces extensive
hold-time

dcd

Syntax

```
dcd (ignore | normal | require);
```

Hierarchy Level

[edit interfaces *interface-name* **serial-options dce-options**],
[edit interfaces *interface-name* **serial-options dte-options**]

Release Information

Statement introduced before Junos OS Release 7.4.

Description

For EIA-530 and V.35 interfaces only, configure the from-DCE signal, data-carrier-detect (DCD).

Options

ignore—The from-DCE signal is ignored.

normal—Normal DCD signal handling as defined by the TIA/EIA Standard 530.

require—The from-DCE signal must be asserted.

Default: normal

Required Privilege Level

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

RELATED DOCUMENTATION

Configuring the Serial Signal Handling | 245

dcd-polarity

Syntax

```
dcd-polarity (negative | positive);
```

Hierarchy Level

[edit interfaces *interface-name* **serial-options**]

Release Information

Statement introduced before Junos OS Release 7.4.

Description

Configure DCD signal polarity.

Options

positive—Positive signal polarity.

negative—Negative signal polarity.

Default: **positive**

Required Privilege Level

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

RELATED DOCUMENTATION

Configuring Serial Signal Polarities | 249

deactivation-delay

Syntax

```
deactivation-delay seconds;
```

Hierarchy Level

```
[edit interfaces dln unit logical-unit-number dialer-options]
```

Release Information

Statement introduced before Junos OS Release 7.4.

Description

On J Series Services Routers with ISDN interfaces, configure the ISDN deactivation delay. Used only for dialer backup and dialer watch cases.

Options

seconds—Interval before the backup interface is deactivated after the primary interface has comes up.

Range: 1 through 4,294,967,295 seconds

Default: 0 (zero)

Required Privilege Level

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

RELATED DOCUMENTATION

Junos OS Interfaces and Routing Configuration Guide

dce-options

Syntax

```
dce-options {
    control-signal (assert | de-assert | normal);
    cts (ignore | normal | require);
    dcd (ignore | normal | require);
    dsr (ignore | normal | require);
    dtr signal-handling-option;
    ignore-all;
    indication (ignore | normal | require);
    rts (assert | de-assert | normal);
    tm (ignore | normal | require);
}
```

Hierarchy Level

```
[edit interfaces interface-name serial-options]
```

Release Information

Statement introduced in Junos OS Release 8.3.
Statement previously known as **control-leads**.

Description

For J Series Services Routers, configure the serial interface signal characteristics.

The remaining statements are explained separately. See CLI Explorer.

Required Privilege Level

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

RELATED DOCUMENTATION

Configuring the Serial Signal Handling | 245

demux-destination (Underlying Interface)

Syntax

```
demux-destination family;
```

Hierarchy Level

```
[edit interfaces interface-name unit logical-unit-number],
[edit logical-systems logical-system-name interfaces interface-name unit logical-unit-number],
[edit logical-systems logical-system-name routing-instances routing-instance-name interfaces interface-name unit
   logical-unit-number]
```

Release Information

Statement introduced in Junos OS Release 9.0.
Support for aggregated Ethernet added in Junos OS Release 9.4.

Description

Configure the logical demultiplexing (demux) destination family type on the IP demux underlying interface.

> **NOTE:** The IP demux interface feature currently supports only Fast Ethernet, Gigabit Ethernet, 10-Gigabit Ethernet, or aggregated Ethernet underlying interfaces.

Required Privilege Level

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

RELATED DOCUMENTATION

Configuring an IP Demultiplexing Interface | 218
Configuring a VLAN Demultiplexing Interface | 223

demux-destination (Demux Interface)

Syntax

```
demux-destination {
    destination-prefix;
}
```

Hierarchy Level

```
[edit interfaces interface-name unit logical-unit-number family family],
[edit logical-systems logical-system-name interfaces interface-name unit logical-unit-number family family],
[edit logical-systems logical-system-name routing-instances routing-instance-name interfaces interface-name unit
    logical-unit-number family family]
```

Release Information

Statement introduced in Junos OS Release 9.0.
Support for aggregated Ethernet added in Junos OS Release 9.4.

Description

Configure one or more logical demultiplexing (demux) destination prefixes. The prefixes are matched against the destination address of packets that the underlying interface receives. When a match occurs, the packet is processed as if it was received on the demux interface.

Required Privilege Level

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

RELATED DOCUMENTATION

Configuring an IP Demultiplexing Interface | 218
Configuring a VLAN Demultiplexing Interface | 223

demux-options (Static Interface)

Syntax

```
demux–options {
    underlying-interface interface-name
}
```

Hierarchy Level

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

Release Information

Statement introduced in Junos OS Release 9.0.

Description

Configure logical demultiplexing (demux) interface options.

The remaining statement is explained separately. See CLI Explorer.

Required Privilege Level

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

RELATED DOCUMENTATION

Configuring an IP Demultiplexing Interface | 218
Configuring a VLAN Demultiplexing Interface | 223

demux-source (Demux Interface)

Syntax

```
demux-source {
   source-prefix;
}
```

Hierarchy Level

```
[edit interfaces interface-name unit logical-unit-number family family],
[edit logical-systems logical-system-name interfaces interface-name unit logical-unit-number family family],
[edit logical-systems logical-system-name routing-instances routing-instance-name interfaces interface-name unit
   logical-unit-number family family]
```

Release Information

Statement introduced in Junos OS Release 9.0.
Support for aggregated Ethernet added in Junos OS Release 9.4.

Description

Configure one or more logical demultiplexing (demux) source prefixes. The prefixes are matched against the source address of packets that the underlying interface receives. When a match occurs, the packet is processed as if it was received on the demux interface.

Required Privilege Level

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

RELATED DOCUMENTATION

Configuring an IP Demultiplexing Interface | 218
Configuring a VLAN Demultiplexing Interface | 223

demux-source (Underlying Interface)

Syntax

```
demux-source family;
```

Hierarchy Level

```
[edit interfaces interface-name unit logical-unit-number],
[edit logical-systems logical-system-name interfaces interface-name unit logical-unit-number],
[edit logical-systems logical-system-name routing-instances routing-instance-name interfaces interface-name unit
   logical-unit-number]
```

Release Information

Statement introduced in Junos OS Release 9.0.
Support for aggregated Ethernet added in Junos OS Release 9.4.

Description

Configure the logical demultiplexing (demux) source family type on the IP demux underlying interface.

> **NOTE:** The IP demux interface feature currently supports only Fast Ethernet, Gigabit Ethernet, 10-Gigabit Ethernet, or aggregated Ethernet underlying interfaces.

Options

family—Protocol family:

- **inet**—Internet Protocol version 4 suite
- **inet6**—Internet Protocol version 6 suite

Required Privilege Level

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

RELATED DOCUMENTATION

Configuring an IP Demultiplexing Interface | 218
Configuring a VLAN Demultiplexing Interface | 223

demux0 (Static Interface)

Syntax

```
demux0 {
    unit logical-unit-number {
        demux-options {
            underlying-interface interface-name
        }
        family family {
            access-concentrator name;
            {
                destination-prefix;
            }
            direct-connect;
            duplicate-protection;
            dynamic-profile profile-name;
            {
                source-prefix;
            }
            max-sessions number;
            service-name-table table-name
            targeted-distribution;
            unnumbered-address interface-name <preferred-source-address address>;
        }
        vlan-id number;
        vlan-tags outer [tpid].vlan-id [inner [tpid].vlan-id];
    }
}
```

Hierarchy Level

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

Release Information

Statement introduced in Junos OS Release 9.0.

Description

Configure the logical demultiplexing (demux) interface.

Logical IP demux interfaces do not support IPv4 and IPv6 dual stack.

The remaining statements are explained separately. See CLI Explorer.

Required Privilege Level

interface—To view this statement in the configuration.

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

RELATED DOCUMENTATION

Configuring an IP Demultiplexing Interface | 218

Configuring a VLAN Demultiplexing Interface | 223

demux0 (Dynamic Interface)

Syntax

```
demux0 {
    unit logical-unit-number {
        demux-options {
            underlying-interface interface-name
        }
        family family {
            access-concentrator name;
            address address;
            demux-source {
                source-prefix;
            }
            direct-connect;
            duplicate-protection;
            dynamic-profile profile-name;
            filter {
                input filter-name;
                output filter-name;
            }
            mac-validate (loose | strict):
            max-sessions number;
            max-sessions-vsa-ignore;
            rpf-check {
                fail-filter filter-name;
                mode loose;
            }
            service-name-table table-name
            short-cycle-protection <lockout-time-min minimum-seconds lockout-time-max maximum-seconds>;
            unnumbered-address interface-name <preferred-source-address address>;
        }
        filter {
            input filter-name;
            output filter-name;
        }
        vlan-id number;
    }
}
```

Hierarchy Level

```
[edit dynamic-profiles profile-name interfaces]
```

Release Information

Statement introduced in Junos OS Release 9.3.

Description

Configure the logical demultiplexing (demux) interface in a dynamic profile.

Logical IP demux interfaces do not support IPv4 and IPv6 dual stack.

The remaining statements are explained separately. Search for a statement in CLI Explorer or click a linked statement in the Syntax section for details.

Required Privilege Level

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

RELATED DOCUMENTATION

Configuring Dynamic Subscriber Interfaces Using IP Demux Interfaces in Dynamic Profiles

Demultiplexing Interface Overview | 214

destination-class-usage

Syntax

```
destination-class-usage;
```

Hierarchy Level

[edit interfaces *interface-name* unit *logical-unit-number* family inet **accounting**],
[edit logical-systems *logical-system-name* interfaces *interface-name* unit *logical-unit-number* family inet **accounting**]

Release Information

Statement introduced before Junos OS Release 7.4.

Description

Enable packet counters on an interface that count packets that arrive from specific customers and are destined for specific prefixes on the provider core router.

Required Privilege Level

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

RELATED DOCUMENTATION

Enabling Source Class and Destination Class Usage | **193**

accounting | **286**

source-class-usage

destination-profile

Syntax

```
destination-profile name;
```

Hierarchy Level

[edit interfaces *interface-name* unit *logical-unit-number* family inet address *address*],
[edit interfaces *interface-name* unit *logical-unit-number* family inet unnumbered-address *interface-name* destination *address*],
[edit logical-systems *logical-system-name* interfaces *interface-name* unit *logical-unit-number* family inet address *address*],
[edit logical-systems *logical-system-name* interfaces *interface-name* unit *logical-unit-number* family inet unnumbered-address *interface-name* destination *address*]

Release Information

Statement introduced before Junos OS Release 7.4.
Statement introduced in Junos OS Release 11.1 for the QFX Series.

Description

For interfaces with PPP encapsulation, assign PPP properties to the remote destination end. You define the profile at the **[edit access group-profile *name* ppp]** hierarchy level.

Options

name—Profile name defined at the **[edit access group-profile *name* ppp]** hierarchy level.

Required Privilege Level

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

RELATED DOCUMENTATION

Configuring IPCP Options for Interfaces with PPP Encapsulation | 175

destination (IPCP)

Junos OS Administration Library

dial-string

Syntax

```
dial-string [ dial-string-numbers ];
```

Hierarchy Level

```
[edit interfaces br-pim/0/port unit logical-unit-number dialer-options],
[edit logical-systems logical-system-name interfaces br-pim/0/port unit logical-unit-number dialer-options]
```

Release Information

Statement introduced before Junos OS Release 7.4.

Description

On J Series Services Routers with ISDN interfaces, specify one or more ISDN dial strings used to reach a destination subnetwork.

Options

dial-string-numbers—One or more strings of numbers to call.

Required Privilege Level

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

RELATED DOCUMENTATION

None

dialer

Syntax

```
dialer filter-name;
```

Hierarchy Level

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

Release Information

Statement introduced before Junos OS Release 7.4.

Description

Apply a dialer filter to an interface. To create the dialer filter, include the **dialer-filter** statement at the **[edit firewall filter family *family*]** hierarchy level.

Options

filter-name—Dialer filter name.

Required Privilege Level

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

RELATED DOCUMENTATION

Junos OS Interfaces and Routing Configuration Guide

dot1x

Syntax

```
dot1x {
    authenticator {
        authentication-profile-name access-profile-name;
        interface (all | [ interface-names ]) {
            authentication-order (captive-portal | dot1x | mac-radius);
            disable;
            guest-bridge-domain guest-bridge-domain;
            guest-vlan guest-vlan;
            ignore-port-bounce;
            mac-radius {
                authentication-protocol {
                    eap-md5;
                    eap-peap {
                        resume;
                    }
                    pap;
                }
                flap-on-disconnect;
                restrict;
            }
            maximum-requests number;
            multi-domain {
                max-data-session max-data-session;
                packet-action (drop-and-log | shutdown);
                recovery-timeout seconds;
            }
            (no-reauthentication | reauthentication interval );
            no-tagged-mac-authentication;
            quiet-period seconds;
            redirect-url redirect-url;
            retries number;
            server-fail (bridge-domain bridge-domain | deny | permit | use-cache | vlan-name vlan-name);
            server-fail-voip (deny | permit | use-cache | vlan-name vlan-name);
            server-reject-bridge-domain bridge-domain {
                block-interval seconds;
                eapol-block;
            }
            server-reject-vlan (vlan-id | vlan-name) {
                block-interval block-interval;
                eapol-block;
```

```
        }
        server-timeout seconds;
        supplicant (single | single-secure | multiple);
        supplicant-timeout seconds;
        transmit-period seconds;
      }
      ip-mac-session-binding;
      no-mac-table-binding;
      radius-options {
        add-interface-text-description;
        use-vlan-id;
        use-vlan-name;
      }
      static mac-address {
        bridge-domain-assignment bridge-domain-assignment;
        interface interface;
        vlan-assignment vlan-identifier;
      }
    }
    }
    ssl-certificate-path path-name;
    traceoptions {
      file filename <files files> <size size> <(world-readable | no-world-readable)>;
      flag (all | config-internal | dot1x-debug | dot1x-event | dot1x-ipc | eapol | esw-if | general | iccp | normal | parse
        | state | task | timer | vlan) {
        disable;
      }
    }
}
```

Hierarchy Level

```
[edit logical-systems name protocols],
[edit protocols]
```

Release Information

Statement introduced in Junos OS Release 9.0 for EX Series switches.
Statement introduced in Junos OS Release 9.3 for MX Series routers.
Statement introduced in Junos OS Release 14.1X53-D30 for the QFX Series.
Statement introduced in Junos OS Release 15.1X49-D80 for SRX Series.
ssl-certificate-path introduced in Junos OS Release 19.4.
ip-mac-session-binding introduced in Junos OS Release 20.2

Description

Configure IEEE 802.1X authentication for Port-Based Network Access Control. 802.1X authentication is supported on interfaces that are members of private VLANs (PVLANs).

Default

802.1X is disabled.

Options

ssl-certificate-path ***path-name***—Specify the file path for SSL certificates if you are not using the default path. The default path for SSL certificates is **/var/tmp**.

The remaining statements are explained separately. Search for a statement in CLI Explorer or click a linked statement in the Syntax section for details.

Required Privilege Level

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

RELATED DOCUMENTATION

show dot1x

Example: Setting Up 802.1X for Single-Supplicant or Multiple-Supplicant Configurations on an EX Series Switch

Example: Setting Up 802.1X in Conference Rooms to Provide Internet Access to Corporate Visitors on an EX Series Switch

Example: Setting Up VoIP with 802.1X and LLDP-MED on an EX Series Switch

Example: Configuring Static MAC Bypass of 802.1X and MAC RADIUS Authentication on an EX Series Switch

Example: Configuring MAC RADIUS Authentication on an EX Series Switch

Configuring RADIUS Server Fail Fallback (CLI Procedure)

dsr

Syntax

```
dsr (ignore | normal | require);
```

Hierarchy Level

[edit interfaces *interface-name* **serial-options dce-options**],
[edit interfaces *interface-name* **serial-options dte-options**]

Release Information

Statement introduced before Junos OS Release 7.4.

Description

For EIA-530 and V.35 interfaces only, configure the from-DCE signal, data-set-ready (DSR).

Options

ignore—The from-DCE signal is ignored.

normal—Normal DSR signal handling as defined by the TIA/EIA Standard 530.

require—The from-DCE signal must be asserted.

Default: normal

Required Privilege Level

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

RELATED DOCUMENTATION

Configuring the Serial Signal Handling | 245

dsr-polarity

Syntax

```
dsr-polarity (negative | positive);
```

Hierarchy Level

```
[edit interfaces interface-name serial-options]
```

Release Information

Statement introduced before Junos OS Release 7.4.

Description

Configure DSR signal polarity.

Options

positive—Positive signal polarity.

negative—Negative signal polarity.

Default: **positive**

Required Privilege Level

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

RELATED DOCUMENTATION

Configuring Serial Signal Polarities | 249

dte-options

Syntax

```
dte-options {
    control-signal (assert | de-assert | normal);
    cts (ignore | normal | require);
    dcd (ignore | normal | require);
    dsr (ignore | normal | require);
    dtr signal-handling-option;
    ignore-all;
    indication (ignore | normal | require);
    rts (assert | de-assert | normal);
    tm (ignore | normal | require);
}
```

Hierarchy Level

```
[edit interfaces interface-name serial-options]
```

Release Information

Statement introduced in Junos OS Release 8.3.
Statement previously known as **control-leads**.

Description

For M Series and T Series routers, configure the serial interface signal characteristics.

The remaining statements are explained separately. See CLI Explorer.

Required Privilege Level

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

RELATED DOCUMENTATION

Configuring the Serial Signal Handling | 245

dtr

Syntax

```
dtr signal-handling-option;
```

Hierarchy Level

```
[edit interfaces interface-name serial-options dce-options],
[edit interfaces interface-name serial-options dte-options]
```

Release Information

Statement introduced before Junos OS Release 7.4.

Description

For EIA-530 and V.35 interfaces only, configure the to-DCE signal, data-transmit-ready (DTR).

Options

signal-handling-option—Signal handling for the DTR signal. The signal handling can be one of the following:

assert—The to-DCE signal must be asserted.

auto-synchronize—Normal DTR signal with automatic synchronization. This statement has two substatements:

duration *milliseconds*—Pulse duration of resynchronization.

Range: 1 through 1000 milliseconds

Default: 1000 milliseconds

interval *seconds*—Offset interval for resynchronization.

Range: 1 through 31 seconds

Default: 15 seconds

de-assert—The to-DCE signal must be deasserted.

normal—Normal DTR signal handling as defined by the TIA/EIA Standard 530.

Default: normal

Required Privilege Level

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

RELATED DOCUMENTATION

Configuring the Serial Signal Handling | 245

dtr-circuit

Syntax

```
dtr-circuit (balanced | unbalanced);
```

Hierarchy Level

```
[edit interfaces interface-name serial-options]
```

Release Information

Statement introduced before Junos OS Release 7.4.

Description

For EIA-530 and V.35 interfaces only, configure a DTR circuit.

Options

balanced—Balanced DTR signal.

unbalanced—Unbalanced DTR signal.

Default: balanced

Required Privilege Level

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

RELATED DOCUMENTATION

Configuring the Serial DTR Circuit | 249

dtr-polarity

Syntax

```
dtr-polarity (negative | positive);
```

Hierarchy Level

```
[edit interfaces interface-name serial-options]
```

Release Information

Statement introduced before Junos OS Release 7.4.

Description

Configure DTR signal polarity.

Options

positive—Positive signal polarity.

negative—Negative signal polarity.

Default: positive

Required Privilege Level

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

RELATED DOCUMENTATION

Configuring Serial Signal Polarities | 249

encoding

Syntax

```
encoding (nrz | nrzi);
```

Hierarchy Level

```
[edit interfaces interface-name serial-options]
```

Release Information

Statement introduced before Junos OS Release 7.4.

Description

For serial interfaces, set the line encoding format.

Default

The default line encoding is non-return to zero (NRZ).

Options

nrz—Use NRZ line encoding.

nrzi—Use non-return to zero inverted (NRZI) line encoding.

Required Privilege Level

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

RELATED DOCUMENTATION

Configuring Serial Line Encoding | 252

f-max-period

Syntax

```
f-max-period number;
```

Hierarchy Level

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

Release Information

Statement introduced before Junos OS Release 7.4.

Description

For all adaptive services interfaces and for ISDN interfaces on J Series Services Routers. Specify the maximum number of compressed packets allowed between the transmission of full headers in a compressed Real-Time Transport Protocol (RTP) traffic stream.

Options

number—Maximum number of packets. The value can be from **1** through **65535**.

Required Privilege Level

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

RELATED DOCUMENTATION

Configuring Bandwidth on Demand

Junos OS Services Interfaces Library for Routing Devices

forward-and-send-to-re

Syntax

```
forward-and-send-to-re;
```

Hierarchy Level

[edit interfaces *interface-name* unit *logical-unit-number* family inet targeted-broadcast],
[edit logical-systems *logical-system-name* interfaces *interface-name* unit *logical-unit-number* family inet targeted-broadcast]

Release Information

Statement introduced in Junos OS Release 10.2.

Description

Specify that IP packets destined for a Layer 3 broadcast address be forwarded to an egress interface and the Routing Engine. The packets are broadcast only if the egress interface is a LAN interface.

Required Privilege Level

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

RELATED DOCUMENTATION

Configuring Targeted Broadcast | 205

targeted-broadcast

Understanding Targeted Broadcast | 204

forward-only

Syntax

```
forward-only;
```

Hierarchy Level

[edit interfaces *interface-name* unit *logical-unit-number* family inet targeted-broadcast],
[edit logical-systems *logical-system-name* interfaces *interface-name* unit *logical-unit-number* family inet
targeted-broadcast]

Release Information

Statement introduced in Junos OS Release 10.2.

Description

Specify that IP packets destined for a Layer 3 broadcast address be forwarded to an egress interface only. The packets are broadcast only if the egress interface is a LAN interface.

Required Privilege Level

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

RELATED DOCUMENTATION

Configuring Targeted Broadcast | 205
targeted-broadcast
Understanding Targeted Broadcast | 204

hierarchical-policer

Syntax

```
hierarchical-policer name {
aggregate {
    if-exceeding {
        bandwidth-limit bandwidth;
        burst-size-limit burst;
    }
    then {
        discard;
    }
}
premium {
    if-exceeding {
        bandwidth-limit bandwidth;
        burst-size-limit burst;
    }
    then {
        discard;
    }
}
```

Hierarchy Level

```
[edit firewall]
```

Release Information

Statement introduced in Junos OS Release 9.5.

Description

For M40e, M120, and M320 (with FFPC and SFPC) edge routers and T320, T640, and T1600 core routers with Enhanced Intelligent Queuing (IQE) PICs, specify a hierarchical policer.

Options

Options are described separately.

Required Privilege Level

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

RELATED DOCUMENTATION

Applying Policers

Class of Service User Guide (Routers and EX9200 Switches)

idle-timeout

Syntax

```
idle-timeout seconds;
```

Hierarchy Level

```
[edit interfaces dln unit logical-unit-number dialer-options]
```

Release Information

Statement introduced before Junos OS Release 7.4.

Description

On J Series Services Routers with ISDN interfaces, configure the number of seconds the link is idle before losing connectivity.

Options

seconds—Time for which the connection can remain idle. For interfaces configured to use a filter for traffic, the idle timeout is based on traffic.

Range: 1 through 429497295

Default: 120 seconds

Required Privilege Level

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

RELATED DOCUMENTATION

Junos OS Interfaces and Routing Configuration Guide

if-exceeding-pps (Hierarchical Policer)

Syntax

```
if-exceeding-pps {
    pps-limit pps;
    packet-burst packets;
}
```

Hierarchy Level

```
[edit dynamic-profiles  profile-name firewall hierarchical-policer hierarchical-policer-name aggregate],
[edit dynamic-profiles  profile-name firewall hierarchical-policer hierarchical-policer-name premium],
[edit firewall hierarchical-policer hierarchical-policer-name aggregate],
[edit firewall hierarchical-policer hierarchical-policer-name premium]
```

Release Information

Statement introduced in Junos OS Release 15.2 for MX Series routers with MPCs.

Description

For MX Series routers , **if-exceeding-pps** allows you to configure a packets-per-second (pps)-based trigger for a premium or aggregate component of a hierarchical policer. When applied to the loopback interface (lo0), this kind of trigger can help protect the Routing Engine from DDoS attacks. When applied in other areas, to either transit or control traffic, it is a more fine-grained monitor.

The remaining statements are explained separately. See CLI Explorer.

Required Privilege Level

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

RELATED DOCUMENTATION

Hierarchical Policer Configuration Overview

Hierarchical Policers

aggregate (Hierarchical Policer)

bandwidth-limit (Hierarchical Policer)

burst-size-limit (Hierarchical Policer)

hierarchical-policer

premium (Hierarchical Policer)

ignore-all

Syntax

```
ignore-all;
```

Hierarchy Level

[edit interfaces *interface-name* **serial-options dce-options**],
[edit interfaces *interface-name* **serial-options dte-options**]

Release Information

Statement introduced before Junos OS Release 7.4.

Description

Ignore all control leads. You can include the **ignore-all** statement in the configuration only if you do not explicitly enable other signal handling options at the **dte-options** hierarchy level.

Required Privilege Level

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

RELATED DOCUMENTATION

Configuring the Serial Signal Handling | 245

incoming-map

Syntax

```
incoming-map {
    caller caller-number | accept-all;
}
```

Hierarchy Level

```
[edit interfaces dln unit logical-unit-number dialer-options],
[edit logical-systems logical-system-name interfaces dln unit logical-unit-number dialer-options]
```

Release Information

Statement introduced in Junos OS Release 7.5.

Description

On J Series Services Routers with interfaces configured for ISDN, specify the dialer to accept incoming calls.

The remaining statements are explained separately. See CLI Explorer.

> **NOTE:** The **incoming-map** statement is mandatory for the router to accept any incoming ISDN calls.

Required Privilege Level

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

RELATED DOCUMENTATION

Junos OS Interfaces and Routing Configuration Guide

indication

Syntax

```
indication (ignore | normal | require);
```

Hierarchy Level

[edit interfaces *interface-name* **serial-options dce-options**],
[edit interfaces *interface-name* **serial-options dte-options**]

Release Information

Statement introduced before Junos OS Release 7.4.

Description

For X.21 interfaces only, configure the from-DCE signal indication.

Options

ignore—The from-DCE signal is ignored.

normal—Normal indication signal handling as defined by ITU-T Recommendation X.21.

require—The from-DCE signal must be asserted.

Default: normal

Required Privilege Level

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

RELATED DOCUMENTATION

Configuring the Serial Signal Handling | 245

indication-polarity

Syntax

```
indication-polarity (negative | positive);
```

Hierarchy Level

[edit interfaces *interface-name* **serial-options**]

Release Information

Statement introduced before Junos OS Release 7.4.

Description

For X.21 interfaces only, configure the indication signal polarity.

Options

positive—Positive signal polarity.

negative—Negative signal polarity.

Default: positive

Required Privilege Level

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

RELATED DOCUMENTATION

Configuring Serial Signal Polarities | 249

init-command-string

Syntax

```
init-command-string initialization-command-string;
```

Hierarchy Level

```
[edit interfaces umd0 modem-options]
```

Release Information

Statement introduced in Junos OS Release 8.2.

Description

For J Series Services Routers, configure the command string used to initialize the USB modem.

When you connect the USB modem to the USB port on a Services Router, the router applies the modem AT commands configured in the **init-command-string** command to the initialization commands on the modem.

For example, the initialization command string **ATS0 = 2\n** configures the USB modem to pick up a call after 2 rings.

If you do not include the **init-command-string** statement, the router applies the default initialization string to the modem.

Options

initialization-command-string—Specify an initialization command string using the following AT command values:

- **%C0**—Disables data compression.
- **&C1**—Disables reset of the modem when it loses the carrier signal.
- **&Q8**—Enables Microcom Networking Protocol (MNP) error control mode.
- **AT**—Attention. Informs the modem that a command follows.
- **E0**—Disables the display on the local terminal of commands issued to the modem from the local terminal.
- **Q0**—Enables the display of result codes.
- **S0=0**—Disables the auto-answer feature, whereby the modem automatically answers calls.
- **S7=45**—Instructs the modem to wait 45 seconds for a telecommunications service provider (carrier) signal before terminating the call.
- **V1**—Displays result codes as words.

Default: AT S7=45 S0=0 V1 X4 &C1 E0 Q0 &Q8 %C0

Required Privilege Level

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

initial-route-check

Syntax

```
initial-route-check seconds;
```

Hierarchy Level

```
[edit interfaces dln unit logical-unit-number dialer-options]
```

Release Information

Statement introduced before Junos OS Release 7.4.

Description

On J Series Services Routers with ISDN interfaces, allows the router to check whether the primary route is up after the initial startup of the router is complete and the timer expires.

Options

seconds—How long to wait to check if the primary interface is up after the router comes up.

Range: 1 through 300 seconds

Default: 120 seconds

Required Privilege Level

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

RELATED DOCUMENTATION

ISDN Interfaces Overview

Junos OS Interfaces and Routing Configuration Guide

input-list

Syntax

```
input-list [ filter-names ];
```

Hierarchy Level

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

Release Information

Statement introduced in Junos OS Release 7.6.

Description

Apply a group of filters to evaluate when packets are received on an interface.

Options

[*filter-names*]—Name of a filter to evaluate when packets are received on the interface. Up to 16 filters can be included in a filter input list.

Required Privilege Level

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

RELATED DOCUMENTATION

Applying a Filter to an Interface | 187

Routing Policies, Firewall Filters, and Traffic Policers User Guide

Junos OS Administration Library

output-list

interface (Hierarchical CoS Schedulers)

Syntax

```
interface interface-name;
```

Hierarchy Level

```
[edit interfaces interface-set interface-set-name]
```

Release Information

Statement introduced in Junos OS Release 8.5.

Description

Specify an interface that is a member of the interface set. Supported on Ethernet interfaces on an MX Series router, Ethernet interfaces on IQ2E PIC on M Series and T Series routers, and IP demux interfaces on an MX Series router.

Required Privilege Level

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

RELATED DOCUMENTATION

Class of Service User Guide (Routers and EX9200 Switches)

interface-range

Syntax

```
interface-range name {
    member-range interface-name-fpc/pic/port to interface-name-fpc/pic/port;
    member interface-name-fpc/pic/port;
    member interface-name-fpc/[low-high]/*;
    member interface-name-fpc/[pic1,pic2,pic3...picN]/port
    /*Common config is added as part of interface-range definition, as follows*/
    mtu 256;
    hold-time up 10;
    ether-options {
        flow-control;
        speed {
            100m;
        }
        802.3ad primary;
    }
}
```

Hierarchy Level

```
[edit interfaces]
```

Release Information

Statement introduced in Junos OS Release 10.0.

Description

Specify a set of identical interfaces as an interface group, to which you can apply a common configuration to the entire set of interfaces. This group can consist of both lexical member ranges of interfaces specified using the **member-range *interface-type-fpc/pic/port* to *xx-fpc/pic/port*** option (regex not supported), and of individual or non-sequential members using the **member *interface-type-fpc/pic/port*** option (with regex support to specify the ***fpc/pic/port*** values).

Options

member-range—Adds interfaces in lexical order. Regex is not supported.

> **Format:—member-range <start-range> to <end-range>**
>
> **Example:—member-range ge-0/0/0 to ge-4/0/40;**

member—To add individual interfaces or multiple interfaces using regex.

Format:—member <list of interface names>

Example:—member ge-0/0/0;

member ge-0/1/1;

member ge-0/*/*;

member ge-0/[1-10]/0;

member ge-1/[1,3,6,10]/12

Required Privilege Level

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

RELATED DOCUMENTATION

Configuring Interface Ranges | 92

interface-transmit-statistics

Syntax

```
interface-transmit-statistics;
```

Hierarchy Level

[edit interface *interface-name*]

Release Information

Statement introduced in Junos OS Release 11.4R3 for MX Series devices.

Description

Configure the interface to report the transmitted load statistics. If this statement is not included in the configuration, the interface statistics show the offered load on the interface, and not the actual transmitted load.

Required Privilege Level

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

RELATED DOCUMENTATION

Improvements to Interface Transmit Statistics Reporting | 413

show interfaces

interface-set (Ethernet Interfaces)

Syntax

```
interface-set interface-set-name {
    interface ethernet-interface-name {
        (unit unit-number | vlan-tags-outer vlan-tag);
    }
}
```

Hierarchy Level

```
[edit interfaces]
```

Release Information

Statement introduced in Junos OS Release 8.5.

Description

The set of interfaces used to configure hierarchical CoS schedulers on Ethernet interfaces on the MX Series router and IQ2E PIC on M Series and T Series routers.

The remaining statements are described separately.

Required Privilege Level

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

RELATED DOCUMENTATION

interface-set (Hierarchical Schedulers)

interface-shared-with

Syntax

```
interface-shared-with psdn;
```

Hierarchy Level

```
[edit interfaces ge-fpc/pic/slot unit logical-unit-number],
[edit interfaces so-fpc/pic/slot unit logical-unit-number],
[edit interfaces xe-fpc/pic/slot unit logical-unit-number]
```

Release Information

Statement introduced in Junos OS Release 9.3.
Statement introduced in Junos OS Release 12.3R2 for EX Series switches.

Description

Assign a logical interface under a shared physical interface to a Protected System Domain (PSD).

Options

n—PSD identification as a numeric value.

Range: **1** through **31**

Required Privilege Level

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

RELATED DOCUMENTATION

isdn-options

Syntax

```
isdn-options {
    bchannel-allocation (ascending | descending);
    calling-number number;
    incoming-called-number number <reject>;
    spid1 spid-string;
    spid2 spid-string;
    static-tei-val value;
    switch-type (att5e | etsi | ni1 | ntdms100 | ntt);
    t310 seconds;
    tei-option (first-call | power-up);
}
```

Hierarchy Level

```
[edit interfaces br-pim/0/port],
[edit interfaces ct1-pim/0/port],
[edit interfaces ce1-pim/0/port]
```

Release Information

Statement introduced before Junos OS Release 7.4.
bchannel-allocation option added in Junos OS Release 8.3.

Description

For J Series Services Routers only. Specify the ISDN options for configuring ISDN interfaces for group and user sessions.

The remaining statements are explained separately. See CLI Explorer.

Required Privilege Level

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

RELATED DOCUMENTATION

Configuring ISDN Physical Interface Properties

Allocating B-Channels for Dialout

Junos OS Interfaces and Routing Configuration Guide

keep-address-and-control

Syntax

```
keep-address-and-control;
```

Hierarchy Level

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

Release Information

Statement introduced before Junos OS Release 7.4.

Description

For interfaces with encapsulation type PPP CCC, do not remove the address and control bytes before encapsulating the packet into a tunnel.

Default

If you do not include this statement, address and control bytes are removed before encapsulating the packet into a tunnel.

Required Privilege Level

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

RELATED DOCUMENTATION

Disabling the Removal of Address and Control Bytes | 186

key

Syntax

```
key number;
```

Hierarchy Level

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

Release Information

Statement introduced before Junos OS Release 7.4.

Description

For Adaptive Services PICs on M Series routers (except the M320 and M120 routers), identify an individual traffic flow within a tunnel, as defined in RFC 2890, *Key and Sequence Number Extensions to GRE*.

Options

number—Value of the key.

Range: 0 through 4,294,967,295

Required Privilege Level

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

RELATED DOCUMENTATION

Junos OS Services Interfaces Library for Routing Devices

lcp-max-conf-req

Syntax

```
lcp-max-conf-req number
```

Hierarchy Level

```
[edit interfaces so-fpc/pic/port unit number ppp-options]
```

Release Information

Statement introduced in Junos OS Release 9.6.

Description

Set the maximum number of LCP Configure-Requests to be sent, after which the router goes to LCP down state.

Options

number—From 0 to 65,535, where 0 means send infinite LCP Configure-Requests, and any other value specifies the maximum number LCP Configure-Requests to send and then stop sending.

Default—254

Required Privilege Level

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

RELATED DOCUMENTATION

Configuring the Maximum Number of LCP Configure-Requests to be Sent

ppp-options

lcp-restart-timer

Syntax

```
lcp-restart-timer milliseconds;
```

Hierarchy Level

```
[edit interfaces interface-name unit logical-unit-number ppp-options],
[edit logical-systems logical-system-name interfaces interface-nameunit logical-unit-number ppp-options]
```

Release Information

Statement introduced in Junos OS Release 8.1.

Description

For interfaces with PPP, PPP TCC, PPP over Ethernet, PPP over ATM, and PPP over Frame Relay encapsulations, configure a restart timer for the Link Control Protocol (LCP) component of a PPP session.

Options

milliseconds—The time, in milliseconds, between successive LCP configuration requests.

Range: 20 through 10000 milliseconds

Default: 3 seconds

Required Privilege Level

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

RELATED DOCUMENTATION

Configuring the PPP Restart Timers

line-protocol

Syntax

```
line-protocol protocol;
```

Hierarchy Level

```
[edit interfaces interface-name serial-options]
```

Release Information

Statement introduced before Junos OS Release 7.4.

Description

For serial interfaces only, configure the line protocol.

Options

protocol—You can specify the one of the following line protocols:

- **eia530**—Line protocol EIA-530
- **v.35**—Line protocol V.35
- **x.21**—Line protocol X.21

Required Privilege Level

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

RELATED DOCUMENTATION

Configuring the Serial Line Protocol | 237

line-rate

Syntax

```
line-rate line-rate;
```

Hierarchy Level

```
[edit interfaces interface-name shdsl-options],
[edit logical-systems logical-system-name interfaces interface-name shdsl-options]
```

Release Information

Statement introduced in Junos OS Release 7.4.

Description

For J Series Services Routers only, configure the SHDSL line rate.

Options

line-rate—SHDSL line rate, in Kbps. Possible values are:

2-wire (Kbps): **192**, **256**, **320**, **384**, **448**, **512**, **576**, **640**, **704**, **768**, **832**, **896**, **960**, **1024**, **1088**, **1152**, **1216**, **1280**, **1344**, **1408**, **1472**, **1536**, **1600**, **1664**, **1728**, **1792**, **1856**, **1920**, **1984**, **2048**, **2112**, **2176**, **2240**, **2304**, **auto**

4-wire (Kbps): **384**, **512**, **640**, **768**, **896**, **1024**, **1152**, **1280**, **1408**, **1536**, **1664**, **1792**, **1920**, **2048**, **2176**, **2304**, **2432**, **2560**, **2688**, **2816**, **2944**, **3072**, **3200**, **3328**, **3456**, **3584**, **3712**, **3840**, **3968**, **4096**, **4224**, **4352**, **4480**, **4608**

Default: For 2-wire mode, **auto**; for 4-wire mode, **4608** Kbps

Required Privilege Level

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

load-interval

Syntax

```
load-interval seconds;
```

Hierarchy Level

```
[edit interfaces dln unit logical-unit-number dialer-options],
[edit logical-systems logical-system-name interfaces dln unit logical-unit-number dialer-options]
```

Release Information

Statement introduced before Junos OS Release 7.4.

Description

On J Series Services Routers with ISDN logical interfaces, specify the interval used to calculate the average load on the network. By default, the average interface load is calculated every 60 seconds.

Options

seconds—Number of seconds at which the average load calculation is triggered.

Range: 20 through 180, in 10-second intervals

Default: 60 seconds

Required Privilege Level

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

RELATED DOCUMENTATION

Junos OS Interfaces and Routing Configuration Guide

load-threshold

Syntax

```
load-threshold percent;
```

Hierarchy Level

```
[edit interfaces dln unit logical-unit-number dialer-options],
[edit logical-systems logical-system-name interfaces dln unit logical-unit-number dialer-options]
```

Release Information

Statement introduced before Junos OS Release 7.4.

Description

On J Series Services Routers with ISDN logical interfaces, specify the bandwidth threshold percentage used for adding interfaces. Another link is added to the multilink bundle when the load reaches the threshold value you set. Specify a percentage between 0 and 100.

Options

percent—Bandwidth threshold percentage used for adding interfaces. When set to 0, all available channels are dialed.

Range: 0 through 100 seconds

Default: 100 seconds

Required Privilege Level

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

RELATED DOCUMENTATION

Junos OS Interfaces and Routing Configuration Guide

local-password

Syntax

```
local-password password;
```

Hierarchy Level

```
[edit interfaces interface-name ppp-options pap],
[edit interfaces interface-name unit logical-unit-number ppp-options pap],
[edit logical-systems logical-system-name interfaces interface-name unit logical-unit-number ppp-options pap]
```

Release Information

Statement introduced in Junos OS Release 8.3.

Description

Configure the host password for sending PAP requests.

Required Privilege Level

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

RELATED DOCUMENTATION

Configuring the Local Password

Configuring the PPP Password Authentication Protocol On a Physical Interface

loopback (Serial)

Syntax

```
loopback mode;
```

Hierarchy Level

```
[edit interfaces interface-name serial-options]
```

Release Information

Statement introduced before Junos OS Release 7.4.

Description

Configure a loopback connection.

Default

If you do not include this statement, there is no loopback connection.

Options

mode—You can specify the one of the following loopback modes:

- **dce-local**—For EIA-530 interfaces only, loop packets back on the local DCE.
- **dce-remote**—For EIA-530 interfaces only, loop packets back on the remote DCE.
- **local**—Loop packets back on the local router's PIC.
- **remote**—Loop packets back on the line interface unit (LIU).

Required Privilege Level

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

RELATED DOCUMENTATION

Configuring Serial Loopback Capability | 250

loopback-clear-timer

Syntax

```
loopback-clear-timer seconds;
```

Hierarchy Level

```
[edit interfaces interface-name unit logical-unit-number ppp-options],
[edit logical-systems logical-system-name interfaces interface-nameunit logical-unit-number ppp-options]
```

Release Information

Statement introduced in Junos OS Release 8.5.

Description

For interfaces with PPP, PPP TCC, PPP over Ethernet, PPP over ATM, and PPP over Frame Relay encapsulations, configure a loop detection clear timer for the Link Control Protocol (LCP) component of a PPP session.

Options

seconds—The time in seconds to wait before the loop detection flag is cleared if it is not cleared by the protocol.

Range: 1 through 60 seconds

Default: 9 seconds

Required Privilege Level

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

RELATED DOCUMENTATION

Configuring the PPP Clear Loop Detected Timer

modem-options

Syntax

```
modem-options {
    dialin (console | routable);
    init-command-string initialization-command-string;
}
```

Hierarchy Level

```
[edit interfaces umd0]
```

Release Information

Statement introduced in Junos OS Release 8.2.

Description

For J Series Services Routers, configure a USB port to act as a USB modem.

The remaining statement is explained separately. See CLI Explorer.

Required Privilege Level

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

monitor-session

Syntax

```
monitor-session (interface-name | all);
```

Hierarchy Level

```
[edit protocols ppp]
```

Release Information

Statement introduced in Junos OS Release 7.5.

Description

Monitor PPP packet exchanges. When monitoring is enabled, packets exchanged during a session are logged to the default log of **/var/log/pppd**.

Default

If you do not include this statement, no PPPD-specific monitoring operations are performed.

Options

all—Monitor PPP packet exchanges on all sessions.

interface-name—Logical interface name on which to enable session monitoring.

Required Privilege Level

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

RELATED DOCUMENTATION

Monitoring a PPP Session

multipoint

Syntax

```
multipoint;
```

Hierarchy Level

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

Release Information

Statement introduced before Junos OS Release 7.4.

Description

Configure the interface unit as a multipoint connection.

Default

If you omit this statement, the interface unit is configured as a point-to-point connection.

Required Privilege Level

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

RELATED DOCUMENTATION

Configuring a Multipoint Connection | 158
point-to-point | 373

ncp-max-conf-req

Syntax

```
ncp-max-conf-req number
```

Hierarchy Level

```
[edit interfaces so-fpc/pic/port unit number ppp-options]
```

Release Information

Statement introduced in Junos OS Release 9.6.

Description

Set the maximum number of NCP Configure-Requests to be sent, after which the router goes to NCP down state.

Options

number—Ranges from 0 to 65535, where 0 means send infinite NCP Configure-Requests and any other value specifies the maximum number NCP Configure-Requests to send and then stop sending.

Default—254

Range: 0 through 65,535

Required Privilege Level

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

RELATED DOCUMENTATION

Configuring the Maximum Number of NCP Configure-Requests to be Sent

ppp-options

ncp-restart-timer

Syntax

```
ncp-restart-timer milliseconds;
```

Hierarchy Level

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

Release Information

Statement introduced in Junos OS Release 8.1.

Description

For interfaces with PPP and PPP TCC encapsulations and on multilink PPP bundle interfaces, configure a restart timer for the Network Control Protocol (NCP) component of a PPP session.

Options

milliseconds—The time in milliseconds between successive NCP configuration requests.

Range: 500 through 10,000 milliseconds

Default: 3 seconds

Required Privilege Level

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

RELATED DOCUMENTATION

Configuring the PPP Restart Timers

operating-mode

Syntax

```
operating-mode mode;
```

Hierarchy Level

```
[edit interfaces at-fpc/pic/port dsl-options]
```

Release Information

Statement introduced before Junos OS Release 7.4.

Description

For J Series Services Routers only, modify the operating mode of the digital subscriber line for an ATM interface.

Options

mode—Operating mode for ATM-over-ADSL interfaces. The mode can be one of the following:

- **adsl2plus**—Set the ADSL line to train in the ITU G.992.5 mode.
- **ansi-dmt**—Set the ADSL line to train in the ANSI T1.413 Issue 2 mode.
- **auto**—Set the ADSL line to autonegotiate the setting to match the setting of the DSL access multiplexer (DSLAM) located at the central office. The ADSL line trains in the ANSI T1.413 Issue 2 (**ansi-dmt**) or ITU G.992.1 (**itu-dmt**) mode.
- **etsi**—Set the ADSL line to train in the ETSI TS 101 388 V1.3.1 mode.
- **itu-annexb-ur2**—Set the ADSL line to train in the ITU G.992.1 UR-2 mode.
- **itu-annexb-non-ur2**—Set the ADSL line to train in the ITU G.992.1 non-UR-2 mode.
- **itu-dmt**—Set the ADSL line to train in the ITU G.992.1 mode.
- **itu-dmt-bis**—Set the ADSL line to train in the ITU G.992.3 mode.

Default: auto

Required Privilege Level

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

RELATED DOCUMENTATION

ATM-over-ADSL Overview

Junos OS Interfaces and Routing Configuration Guide

passive (PAP)

Syntax

```
passive;
```

Hierarchy Level

[edit interfaces *interface-name* ppp-options pap],
[edit interfaces *interface-name* unit *logical-unit-number* ppp-options pap],
[edit logical-systems *logical-system-name* interfaces *interface-name* unit *logical-unit-number* ppp-options pap]

Release Information

Statement introduced in Junos OS Release 8.3.

Description

Initiate an authentication request when the PAP option is received from a peer. If you omit this statement from the configuration, the interface requires the peer to initiate an authentication request.

Required Privilege Level

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

RELATED DOCUMENTATION

Configuring Passive Mode

Junos OS Administration Library

pfc

Syntax

```
pfc;
```

Hierarchy Level

[edit interfaces *interface-name* ppp-options compression],
[edit interfaces *interface-name* unit *logical-unit-number* ppp-options compression],
[edit logical-systems *logical-system-name* interfaces *interface-name* unit *logical-unit-number* ppp-options compression]

Release Information

Statement introduced before Junos OS Release 7.4.

Description

For interfaces with PPP encapsulation, configure the router to compress the protocol field to one byte.

Required Privilege Level

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

RELATED DOCUMENTATION

Configuring the PPP Protocol Field Compression

point-to-point

Syntax

```
point-to-point;
```

Hierarchy Level

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

Release Information

Statement introduced before Junos OS Release 7.4.

Description

For all interfaces except aggregated Ethernet, Fast Ethernet, and Gigabit Ethernet, configure the interface unit as a point-to-point connection. This is the default connection type.

Default

If you omit this statement, the interface unit is configured as a point-to-point connection.

Required Privilege Level

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

RELATED DOCUMENTATION

Configuring a Point-to-Point Connection | 157
multipoint | 367

policer (Interface)

Syntax

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

Hierarchy Level

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

Release Information

Statement introduced before Junos OS Release 7.4.
Support for MX Series routers added in Junos OS Release 20.2R1

Description

Apply a policer to an interface.

Options

arp *policer-template-name*—For **inet** family only, name of one policer to evaluate when ARP packets are received on the interface.

input *policer-template-name*—Name of one policer to evaluate when packets are received on the interface.

output *policer-template-name*—Name of one policer to evaluate when packets are transmitted on the interface.

Required Privilege Level

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

RELATED DOCUMENTATION

Applying Policers

Configuring Firewall Filters and Policers for VPLS

Routing Policies, Firewall Filters, and Traffic Policers User Guide

Junos OS Services Interfaces Library for Routing Devices

pool

Syntax

```
pool pool-name <priority priority>;
```

Hierarchy Level

```
[edit interfaces br-pim/0/port dialer-options],
[edit interfaces umd0 dialer-options],
[edit interfaces dln unit logical-unit-number dialer-options],
[edit logical-systems logical-system-name interfaces dln unit logical-unit-number dialer-options]
```

Release Information

Statement introduced before Junos OS Release 7.4.

Description

On J Series Services Routers, for logical and physical ISDN interfaces, specify the dial pool. The dial pool allows logical (dialer) and physical (**br-*pim*/0/*port***) interfaces to be bound together dynamically on a per-call basis. On a dialer interface, **pool** directs the dialer interface which dial pool to use. On **br-*pim*/0/*port*** interface, **pool** defines the pool to which the interface belongs.

Options

pool-name—Pool identifier.

priority *priority*—(Physical **br-*pim*/0/*port*** interfaces only) Specify a priority value of 0 (lowest) to 255 (highest) for the interface within the pool.

Required Privilege Level

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

RELATED DOCUMENTATION

Junos OS Interfaces and Routing Configuration Guide

preferred-source-address

Syntax

```
preferred-source-address address;
```

Hierarchy Level

```
[edit interfaces interface-name unit logical-unit-number family family unnumbered-address interface-name],
[edit logical-systems logical-system-name interfaces interface-name unit logical-unit-number family family
   unnumbered-address interface-name]
```

Release Information

Statement introduced in Junos OS Release 9.0.

Description

For unnumbered Ethernet interfaces configured with a loopback interface as the donor interface, specify one of the loopback interface's secondary addresses as the preferred source address for the unnumbered Ethernet interface. Configuring the preferred source address enables you to use an IP address other than the primary IP address on some of the unnumbered Ethernet interfaces in your network.

Configuration of a preferred source address for unnumbered Ethernet interfaces is supported for the IPv4 and IPv6 address families.

Options

address—Secondary IP address of the donor loopback interface.

Required Privilege Level

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

RELATED DOCUMENTATION

Configuring an Unnumbered Interface | 177

address

Junos OS Administration Library

primary (Interface for Router)

Syntax

```
primary;
```

Hierarchy Level

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

Release Information

Statement introduced before Junos OS Release 7.4.

Description

Configure the primary interface for a device. By default, the multicast-capable interface with the lowest-index address is chosen as the primary interface. If there is no such interface, the point-to-point interface with the lowest-index address is chosen. Otherwise, any interface with an address can be picked. In practice, this means that, on the device, the **fxp0** or **em0** interface is picked by default. To configure a different interface to be the primary interface, you include this statement.

The *primary interface* for the router has the following characteristics:

- It is the interface through which the packets go out when you type a command such as ping 255.255.255.255—that is, a command that does not include an interface name (there is no interface ***type*-0/0/0.0** qualifier) and where the destination address does not imply any particular outgoing interface.
- It is the interface on which multicast applications running locally on the router, such as Session Announcement Protocol (SAP), perform group joins by default.
- It is the interface from which the default local address is derived for packets sourced out of an unnumbered interface if there are no non-127 addresses configured on the loopback interface, lo0.

Required Privilege Level

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

RELATED DOCUMENTATION

Configuring Default, Primary, and Preferred Addresses and Interfaces | 169

redial-delay

Syntax

```
redial-delay time;
```

Hierarchy Level

```
[edit interfaces dln unit logical-unit-number dialer-options],
[edit logical-systems logical-system-name interfaces dln unit logical-unit-number dialer-options]
```

Release Information

Statement introduced in Junos OS Release 7.5.

Description

On J Series Services Routers with interfaces configured for ISDN with dialout, specify the delay (in seconds) between two successive calls made by the dialer. To configure callback mode, include the **callback** statement at the **[edit interfaces dl*n* unit *logical-unit-number* dialer-options]** hierarchy level.

If the **callback** statement is configured, you cannot use the **caller *caller-id*** statement at the **[edit interfaces dl*n* unit *logical-unit-number* dialer-options]** hierarchy level.

Options

time—Delay (in seconds) between two successive calls.

Range: 2 through 255 seconds

Default: 3 seconds

Required Privilege Level

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

RELATED DOCUMENTATION

ISDN Interfaces Overview

Junos OS Interfaces and Routing Configuration Guide

rts

Syntax

```
rts (assert | de-assert | normal);
```

Hierarchy Level

```
[edit interfaces interface-name serial-options dce-options],
[edit interfaces interface-name serial-options dte-options]
```

Release Information

Statement introduced before Junos OS Release 7.4.

Description

For EIA-530 and V.35 interfaces only, configure the to-DCE signal, request to send (RTS).

Options

assert—The to-DCE signal must be asserted.

de-assert—The to-DCE signal must be deasserted.

normal—Normal RTS signal handling, as defined by the TIA/EIA Standard 530.

Default: normal

Required Privilege Level

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

RELATED DOCUMENTATION

Configuring the Serial Signal Handling | 245

rts-polarity

Syntax

```
rts-polarity (negative | positive);
```

Hierarchy Level

```
[edit interfaces interface-name serial-options]
```

Release Information

Statement introduced before Junos OS Release 7.4.

Description

Configure RTS signal polarity.

Options

negative—Negative signal polarity.

positive—Positive signal polarity.

Default: positive

Required Privilege Level

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

RELATED DOCUMENTATION

Configuring Serial Signal Polarities | 249

serial-options

Syntax

```
serial-options {
    clock-rate rate;
    clocking-mode (dce | loop);
    control-polarity (negative | positive);
    cts-polarity (negative | positive);
    dcd-polarity (negative | positive);
    dce-options {
        control-signal (assert | de-assert | normal);
        cts (ignore | normal | require);
        dcd (ignore | normal | require);
        dsr (ignore | normal | require);
        dtr signal-handling-option;
        ignore-all;
        indication (ignore | normal | require);
        rts (assert | de-assert | normal);
        tm (ignore | normal | require);
    }
    dsr-polarity (negative | positive);
    dte-options {
        control-signal (assert | de-assert | normal);
        cts (ignore | normal | require);
        dcd (ignore | normal | require);
        dsr (ignore | normal | require);
        dtr signal-handling-option;
        ignore-all;
        indication (ignore | normal | require);
        rts (assert | de-assert | normal);
        tm (ignore | normal | require);
    }
    dtr-circuit (balanced | unbalanced);
    dtr-polarity (negative | positive);
    encoding (nrz | nrzi);
    indication-polarity (negative | positive);
    line-protocol protocol;
    loopback (dce-local | dce-remote | local | remote);
    rts-polarity (negative | positive);
    tm-polarity (negative | positive);
    transmit-clock invert;
}
```

Hierarchy Level

```
[edit interfaces se-pim/0/port]
```

Release Information

Statement introduced prior to Junos OS Release 7.4.

Description

Configure serial-specific interface properties.

The remaining statements are explained separately. See CLI Explorer.

Required Privilege Level

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

RELATED DOCUMENTATION

Serial Interfaces Overview | 237

no-concatenate

shdsl-options

Syntax

```
shdsl-options {
    annex (annex-a | annex-b);
    line-rate line-rate;
    loopback (local | remote | payload);
    snr-margin {
        snext margin;
    }
}
```

Hierarchy Level

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

Release Information

Statement introduced in Junos OS Release 7.4.

Description

For J Series Services Routers only, configure symmetric DSL (SHDSL) options.

The remaining statements are explained separately. See CLI Explorer.

Required Privilege Level

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

snr-margin

Syntax

```
snr-margin {
    snext margin;
}
```

Hierarchy Level

```
[edit interfaces interface-name shdsl-options],
[edit logical-systems logical-system-name interfaces interface-name shdsl-options]
```

Release Information

Statement introduced in Junos OS Release 7.4.

Description

For J Series Services Routers only, configure the SHDSL signal-to-noise ratio (SNR) margin. The SNR margin is the difference between the desired SNR and the actual SNR. Configuring the SNR creates a more stable SHDSL connection by making the line train at a SNR margin higher than the threshold. If any external noise below the threshold is applied to the line, the line remains stable.

The remaining statements are explained separately. See CLI Explorer.

Required Privilege Level

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

RELATED DOCUMENTATION

Junos OS Interfaces and Routing Configuration Guide

snext

Syntax

```
snext margin;
```

Hierarchy Level

```
[edit interfaces interface-name shdsl-options snr-margin],
[edit logical-systems logical-system-name interfaces interface-name shdsl-options snr-margin]
```

Release Information

Statement introduced in Junos OS Release 7.4.

Description

For J Series Services Routers only, configure self-near-end crosstalk (SNEXT) signal-to-noise ratio (SNR) margin for a SHDSL line. When configured, the line trains at higher than SNEXT threshold. The SNR margin is the difference between the desired SNR and the actual SNR.

Options

margin—Desired SNEXT margin. Possible values are disabled or a margin between –10dB and 10 dB.

Default: disabled

Required Privilege Level

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

RELATED DOCUMENTATION

Junos OS Interfaces and Routing Configuration Guide

spid1

Syntax

```
spid1 spid1-string;
```

Hierarchy Level

```
[edit interfaces br-pim/0/port isdn-options]
```

Release Information

Statement introduced before Junos OS Release 7.4.

Description

Configure the Service Profile Identifier (SPID).

Options

spid1-string—Numeric SPID.

Required Privilege Level

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

RELATED DOCUMENTATION

Junos OS Interfaces and Routing Configuration Guide

spid2

Syntax

```
spid2 spid2-string;
```

Hierarchy Level

```
[edit interfaces br-pim/0/port isdn-options]
```

Release Information

Statement introduced before Junos OS Release 7.4.

Description

Configure an additional SPID.

Options

spid2-string—Numeric SPID.

Required Privilege Level

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

RELATED DOCUMENTATION

static-tei-val

Syntax

```
static-tei-val value;
```

Hierarchy Level

```
[edit interfaces br-pim/0/port isdn-options]
```

Release Information

Statement introduced before Junos OS Release 7.4.

Description

For J Series Services Routers only. Statically configure the Terminal Endpoint Identifier (TEI) value. The TEI value represents any ISDN-capable device attached to an ISDN network that is the terminal endpoint. TEIs are used to distinguish between several different devices using the same ISDN links.

Options

value—Value between 0 through 63.

Required Privilege Level

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

RELATED DOCUMENTATION

Junos OS Interfaces and Routing Configuration Guide

switch-type

Syntax

```
switch-type (att5e | etsi | ni1 | ntdms-100)
```

Hierarchy Level

```
[edit interfaces br-pim/0/port isdn-options]
```

Release Information

Statement introduced before Junos OS Release 7.4.

Description

For J Series Services Routers only. Configure the ISDN variant supported.

Options

att5e—AT&T switch variant.

etsi—European Telecommunications Standards Institute switch variant.

ni1—National ISDN 1 switch variant.

ntdms-100—Northern Telecom DMS-100.

ntt—NTT Group switch for Japan.

Required Privilege Level

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

RELATED DOCUMENTATION

Junos OS Interfaces and Routing Configuration Guide

t310

Syntax

```
t310-value seconds;
```

Hierarchy Level

```
[edit interfaces br-pim/0/port isdn-options]
```

Release Information

Statement introduced before Junos OS Release 7.4.

Description

For ISDN interfaces, configure the Q.931-specific timer for T310, in seconds. The Q.931 protocol is involved in the setup and termination of connections.

Options

seconds—Timer value, in seconds.

Range: 1 through 65,536 seconds

Default: 10 seconds

Required Privilege Level

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

RELATED DOCUMENTATION

Junos OS Interfaces and Routing Configuration Guide

tei-option

Syntax

```
tei-option (first-call | power-up);
```

Hierarchy Level

```
[edit interfaces br-pim/0/portisdn-options ]
```

Release Information

Statement introduced before Junos OS Release 7.4.

Description

For ISDN interfaces, configure when the Terminal Endpoint Identifier (TEI) negotiates with the ISDN provider.

Options

first-call—Activation does not occur until the call setup is sent.

power-up—Activation occurs when the Services Router is powered on.

Default: power-up

Required Privilege Level

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

RELATED DOCUMENTATION

Junos OS Interfaces and Routing Configuration Guide

then

Syntax

```
then {
    discard;
}
```

Hierarchy Level

```
[edit firewall hierarchical-policer aggregate],
[edit firewall hierarchical-policer premium]
```

Release Information

Statement introduced in Junos OS Release 9.5.

Description

On M40e, M120, and M320 (with FFPC and SFPC) edge routers and T320, T640, and T1600 core routers with Enhanced Intelligent Queuing (IQE) PICs, discard packets when a specified bandwidth or burst limits for an aggregate level of a hierarchical policer is reached.

Options

discard—Discard packets if condition is met.

Required Privilege Level

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

RELATED DOCUMENTATION

Applying Policers

Class of Service User Guide (Routers and EX9200 Switches)

tm

Syntax

```
tm (ignore | normal | require);
```

Hierarchy Level

```
[edit interfaces interface-name serial-options dce-options],
[edit interfaces interface-name serial-options dte-options]
```

Release Information

Statement introduced before Junos OS Release 7.4.

Description

For EIA-530 interfaces only, configure the from-DCE signal, test-mode (TM).

Options

ignore—The from-DCE signal is ignored.

normal—Normal TM signal handling as defined by the TIA/EIA Standard 530.

require—The from-DCE signal must be asserted.

Default: normal

Required Privilege Level

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

RELATED DOCUMENTATION

Configuring the Serial Signal Handling | 245

tm-polarity

Syntax

```
tm-polarity (negative | positive);
```

Hierarchy Level

[edit interfaces *interface-name* **serial-options**]

Release Information

Statement introduced before Junos OS Release 7.4.

Description

Configure TM signal polarity.

Options

negative—Negative signal polarity.

positive—Positive signal polarity.

Default: **positive**

Required Privilege Level

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

RELATED DOCUMENTATION

Configuring Serial Signal Polarities | 249

traceoptions (PPP Process)

Syntax

```
traceoptions {
    file filename <files number> <match regular-expression> <size size> <world-readable | no-world-readable>;
    flag flag;
    level severity-level;
    no-remote-trace;
}
```

Hierarchy Level

```
[edit protocols ppp]
```

Release Information

Statement introduced in Junos OS Release 7.5.

Description

Define tracing operations for the PPP process.

To specify more than one tracing operation, include multiple **flag** statements.

You cannot specify a separate trace tile. Tracing information is placed in the system **syslog** file in the directory **/var/log/pppd**.

Default

If you do not include this statement, no PPPD-specific tracing operations are performed.

Options

filename—Name of the file to receive the output of the tracing operation. Enclose the name within quotation marks. All files are placed in the directory **/var/log**. By default, commit script process tracing output is placed in the file **ppd**. If you include the **file** statement, you must specify a filename. To retain the default, you can specify **eventd** as the filename.

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

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

Range: 2 through 1000

Default: 3 files

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

flag—Tracing operation to perform. To specify more than one tracing operation, include multiple **flag** statements. The following are the PPPD-specific tracing options.

- **access**—Access code
- **address-pool**—Address pool code
- **all**—All areas of code
- **auth**—Authentication code
- **chap**—Challenge Handshake Authentication Protocol (CHAP) code
- **config**—Configuration code
- **ifdb**—Interface database code
- **lcp**—LCP state machine code
- **memory**—Memory management code
- **message**—Message processing code
- **mlppp**—Trace MLPPP code
- **ncp**—NCP state machine code
- **pap**—Password Authentication Protocol (PAP) code
- **ppp**—PPP protocol processing code
- **radius**—RADIUS processing code
- **rtsock**—Routing socket code
- **session**—Session management code
- **signal**—Signal handling code
- **timer**—Timer code
- **ui**—User interface code

match *regex*—(Optional) Refine the output to include only those lines that match the given regular expression.

size *size*—(Optional) Maximum size of each trace file, in kilobytes (KB), megabytes (MB), or gigabytes (GB). When a trace file named ***trace-file*** reaches this size, it is renamed ***trace-file*.0**. When the ***trace-file*** again reaches its maximum size, ***trace-file*.0** is renamed ***trace-file*.1** and ***trace-file*** is renamed ***trace-file*.0**. This renaming scheme continues until the maximum number of trace files is reached. Then the oldest trace file is overwritten.

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

Syntax: **xk** to specify KB, **xm** to specify MB, or **xg** to specify GB

Range: 10 KB through 1 GB

Default: 128 KB

world-readable—(Optional) Enable unrestricted file access.

non-world-readable—(Optional) By default, log files can be accessed only by the user who configures the tracing operation. Specify **non-world-readable** to reset the default.

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

RELATED DOCUMENTATION

Tracing Operations of the pppd Process | **257**

transmit-clock

Syntax

```
transmit-clock invert;
```

Hierarchy Level

[edit interfaces *interface-name* **serial-options**]

Release Information

Statement introduced before Junos OS Release 7.4.

Description

Configure the transmit clock signal.

Options

invert—Shift the clock phase 180 degrees.

Required Privilege Level

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

RELATED DOCUMENTATION

Configuring the Serial Clocking Mode | 242

unnumbered-address (Demux)

Syntax

```
unnumbered-address interface-name <preferred-source-address address>;
```

Hierarchy Level

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

Release Information

Statement introduced in Junos OS Release 8.2.
preferred-source-address option introduced in Junos OS Release 9.0.
IP demultiplexing interfaces supported in Junos OS Release 9.2.

Description

For IP demultiplexing interfaces, enable the local address to be derived from the specified interface. Configuring an unnumbered interface enables IP processing on the interface without assigning an explicit IP address to the interface.

Options

interface-name—Name of the interface from which the local address is derived. The specified interface must have a logical unit number and a configured IP address, and must not be an unnumbered interface.

The **preferred-source-address** 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

Configuring an Unnumbered Interface | 177

address

Junos System Basics Configuration Guide

vlan-id-list (Ethernet VLAN Circuit)

Syntax

```
vlan-id-list [vlan-id vlan-id–vlan-id];
```

Hierarchy Level

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

Release Information

Statement introduced in Junos OS Release 9.5.

Description

Binds a single-tag logical interface to a list of VLAN IDs. Configures a logical interface to receive and forward any tag frame whose VLAN ID tag matches the list of VLAN IDs you specify.

> **NOTE:**
> When you create a circuit cross-connect (CCC) using VLAN-bundled single-tag logical interfaces on Layer 2 VPN routing instances, the circuit automatically uses **ethernet** encapsulation. For Layer 2 VPN, you need to include the **encapsulation-type** statement and specify the value **ethernet** at either of the following hierarchy levels:
>
> - **[edit routing-instances *routing-instance-name* protocols l2vpn]**
> - **[edit logical-systems *logical-system-name* routing-instances *routing-instance-name* protocols l2vpn]**
>
> For more information about the **encapsulation-type** configuration statement and the Layer 2 encapsulation types **ethernet** and **ethernet-vlan**, see the *Junos OS VPNs Library for Routing Devices*.

Options

[*vlan-id vlan-id–vlan-id*]—A list of valid VLAN ID numbers. Specify the VLAN IDs individually by using a space to separate each ID, as an inclusive list by separating the starting VLAN ID and ending VLAN ID with a hyphen, or as a combination of both.

Range: 1 through 4094. VLAN ID 0 is reserved for tagging the priority of frames.

NOTE: Configuring **vlan-id-list** with the entire vlan-id range is an unnecessary waste of system resources and is not best practice. It should be used only when a subset of VLAN IDs (not the entire range) needs to be associated with a logical interface. If you specify the entire range (1-4094), it has the same result as not specifying a range; however, it consumes PFE resources such as VLAN lookup tables entries, and so on.

The following examples illustrate this further:

```
[edit interfaces interface-name]
vlan-tagging;
unit number {
    vlan-id-range 1-4094;
}
```

```
[edit interfaces interface-name]
unit 0;
```

Required Privilege Level

interface—To view this statement in the configuration.

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

RELATED DOCUMENTATION

Binding VLAN IDs to Logical Interfaces

encapsulation (Logical Interface)

encapsulation

encapsulation-type (Layer 2 VPN routing instance), see the *Junos OS VPNs Library for Routing Devices*

flexible-vlan-tagging

vlan-tagging

vlan-tags (Dual-Tagged Logical Interface)

watch-list

Syntax

```
watch-list {
    [ routes ];
}
```

Hierarchy Level

```
[edit interfaces dln unit logical-unit-number dialer-options]
```

Release Information

Statement introduced before Junos OS Release 7.4.

Description

On J Series Services Routers with ISDN interfaces, configure an ISDN list of routes to watch. Used only for dialer watch.

Options

routes—IP prefix of a route. Specify one or more. The primary interface is considered up if there is at least one valid route for any of the addresses in the watch list to an interface other than the backup interface.

Required Privilege Level

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

RELATED DOCUMENTATION

Junos OS Interfaces and Routing Configuration Guide

5

CHAPTER

Operational Commands

Common Output Fields Description

This chapter explains the content of the output fields, which appear in the output of most **show interfaces** commands.

Damping Field

For the physical interface, the Damping field shows the setting of the following damping parameters:

- **half-life**—Decay half-life. The number of seconds after which the accumulated interface penalty counter is reduced by half if the interface remains stable.
- **max-suppress**—Maximum hold-down time. The maximum number of seconds that an interface can be suppressed irrespective of how unstable the interface has been.
- **reuse**—Reuse threshold. When the accumulated interface penalty counter falls below this number, the interface is no longer suppressed.
- **suppress**—Cutoff (suppression) threshold. When the accumulated interface penalty counter exceeds this number, the interface is suppressed.
- **state**—Interface damping state. If damping is enabled on an interface, it is suppressed during interface flaps that match the configured damping parameters.

Destination Class Field

For the logical interface, the **Destination class** field provides the names of destination class usage (DCU) counters per family and per class for a particular interface. The counters display packets and bytes arriving from designated user-selected prefixes. For example:

```
                                    Packets                           Bytes
Destination class              (packet-per-second)             (bits-per-second)

             gold                    1928095                   161959980
                               (               889)            (    597762)
            bronze                         0                            0
                               (                 0)            (          0)
```

```
          silver                                      0                             0
                            (                         0)                (           0)
```

Enabled Field

For the physical interface, the **Enabled** field provides information about the state of the interface, displaying one or more of the following values:

- **Administratively down, Physical link is Down**—The interface is turned off, and the physical link is inoperable and cannot pass packets even when it is enabled.To change the interface state to **Enabled**, use the following command:

  ```
  user@host# set interfaces interface enable
  ```

  Manually verify the connections to bring the physical link up.

- **Administratively down, Physical link is Up**—The interface is turned off, but the physical link is operational and can pass packets when it is enabled.To change the interface state to **Enabled**, use the following command:

  ```
  user@host# set interfaces interface enable
  ```

- **Enabled, Physical link is Down**—The interface is turned on, but the physical link is inoperable and cannot pass packets. Manually verify the connections to bring the physical link up.

- **Enabled, Physical link is Up**—The interface is turned on, and the physical link is operational and can pass packets.

Filters Field

For the logical interface, the **Filters** field provides the name of the firewall filters to be evaluated when packets are received or transmitted on the interface. The format is **Filters: Input: *filter-name*** and **Filters: Output: *filter-name***. For example:

```
Filters: Input: sample-all
Filters: Output: cp-ftp
```

Flags Fields

IN THIS SECTION

The following sections provide information about flags that are specific to interfaces:

Addresses, Flags Field

The **Addresses, Flags** field provides information about the addresses configured for the protocol family on the logical interface and displays one or more of the following values:

- **Dest-route-down**—The routing process detected that the link was not operational and changed the interface routes to nonforwarding status
- **Is-Default**—The default address of the router used as the source address by SNMP, ping, traceroute, and other network utilities.
- **Is-Preferred**—The default local address for packets originating from the local router and sent to destinations on the subnet.
- **Is-Primary**—The default local address for broadcast and multicast packets originated locally and sent out the interface.
- **Preferred**—This address is a candidate to become the preferred address.
- **Primary**—This address is a candidate to become the primary address.
- **Trunk**—Interface is a trunk.
- **Trunk, Inter-Switch-Link**—Interface is a trunk, and InterSwitch Link protocol (ISL) is configured on the trunk port of the primary VLAN in order to connect the routers composing the PVLAN to each other.

Device Flags Field

The **Device flags** field provides information about the physical device and displays one or more of the following values:

- **ASIC Error**—Device is down because of ASIC wedging and due to which PFE is disabled.
- **Down**—Device has been administratively disabled.
- **Hear-Own-Xmit**—Device receives its own transmissions.
- **Link-Layer-Down**—The link-layer protocol has failed to connect with the remote endpoint.
- **Loopback**—Device is in physical loopback.
- **Loop-Detected**—The link layer has received frames that it sent, thereby detecting a physical loopback.
- **No-Carrier**—On media that support carrier recognition, no carrier is currently detected.
- **No-Multicast**—Device does not support multicast traffic.
- **Present**—Device is physically present and recognized.
- **Promiscuous**—Device is in promiscuous mode and recognizes frames addressed to all physical addresses on the media.
- **Quench**—Transmission on the device is quenched because the output buffer is overflowing
- **Recv-All-Multicasts**—Device is in multicast promiscuous mode and therefore provides no multicast filtering.
- **Running**—Device is active and enabled.

Family Flags Field

The **Family flags** field provides information about the protocol family on the logical interface and displays one or more of the following values:

- **DCU**—Destination class usage is enabled.
- **Dest-route-down**—The software detected that the link is down and has stopped forwarding the link's interface routes.
- **Down**—Protocol is inactive.
- **Is-Primary**—Interface is the primary one for the protocol.
- **Mac-Validate-Loose**—Interface is enabled with loose MAC address validation.
- **Mac-Validate-Strict**—Interface is enabled with strict MAC address validation.
- **Maximum labels**—Maximum number of MPLS labels configured for the MPLS protocol family on the logical interface.

- **MTU-Protocol-Adjusted**—The effective MTU is not the configured value in the software.
- **No-Redirects**—Protocol redirects are disabled.
- **Primary**—Interface can be considered for selection as the primary family address.
- **Protocol-Down**—Protocol failed to negotiate correctly.
- **SCU-in**—Interface is configured for source class usage input.
- **SCU-out**—Interface is configured for source class usage output.
- **send-bcast-packet-to-re**—Interface is configured to forward IPv4 broadcast packets to the Routing Engine.
- **targeted-broadcast**—Interface is configured to forward IPv4 broadcast packets to the LAN interface and the Routing Engine.
- **Unnumbered**—Protocol family is configured for unnumbered Ethernet. An unnumbered Ethernet interface borrows an IPv4 address from another interface, which is referred to as the donor interface.
- **Up**–Protocol is configured and operational.
- **uRPF**—Unicast Reverse Path Forwarding is enabled.

Interface Flags Field

The **Interface flags** field provides information about the physical interface and displays one or more of the following values:

- **Admin-Test**—Interface is in test mode and some sanity checking, such as loop detection, is disabled.
- **Disabled**—Interface is administratively disabled.
- **Down**—A hardware failure has occurred.
- **Hardware-Down**—Interface is nonfunctional or incorrectly connected.
- **Link-Layer-Down**—Interface keepalives have indicated that the link is incomplete.
- **No-Multicast**—Interface does not support multicast traffic.
- **No-receive No-transmit**—Passive monitor mode is configured on the interface.
- **OAM-On-SVLAN**—(MX Series routers with MPC/MIC interfaces only) Interface is configured to propagate the Ethernet OAM state of a static, single-tagged service VLAN (S-VLAN) on a Gigabit Ethernet, 10-Gigabit Ethernet, or aggregated Ethernet interface to a dynamic or static double-tagged customer VLAN (C-VLAN) that has the same S-VLAN (outer) tag as the S-VLAN.
- **Point-To-Point**—Interface is point-to-point.
- **Pop all MPLS labels from packets of depth**—MPLS labels are removed as packets arrive on an interface that has the **pop-all-labels** statement configured. The depth value can be one of the following:
  - **1**—Takes effect for incoming packets with one label only.

- **2**—Takes effect for incoming packets with two labels only.
- **[1 2]**—Takes effect for incoming packets with either one or two labels.

- **Promiscuous**—Interface is in promiscuous mode and recognizes frames addressed to all physical addresses.
- **Recv-All-Multicasts**—Interface is in multicast promiscuous mode and provides no multicast filtering.
- **SNMP-Traps**—SNMP trap notifications are enabled.
- **Up**—Interface is enabled and operational.

Link Flags Field

The **Link flags** field provides information about the physical link and displays one or more of the following values:

- **ACFC**—Address control field compression is configured. The Point-to-Point Protocol (PPP) session negotiates the ACFC option.
- **Give-Up**—Link protocol does not continue connection attempts after repeated failures.
- **Loose-LCP**—PPP does not use the Link Control Protocol (LCP) to indicate whether the link protocol is operational.
- **Loose-LMI**—Frame Relay does not use the Local Management Interface (LMI) to indicate whether the link protocol is operational.
- **Loose-NCP**—PPP does not use the Network Control Protocol (NCP) to indicate whether the device is operational.
- **No-Keepalives**—Link protocol keepalives are disabled.
- **PFC**—Protocol field compression is configured. The PPP session negotiates the PFC option.

Logical Interface Flags Field

The **Logical interface flags** field provides information about the logical interface and displays one or more of the following values:

- **ACFC Encapsulation**—Address control field Compression (ACFC) encapsulation is enabled (negotiated successfully with a peer).
- **Device-down**—Device has been administratively disabled.
- **Disabled**—Interface is administratively disabled.
- **Down**—A hardware failure has occurred.
- **Clear-DF-Bit**—GRE tunnel or IPsec tunnel is configured to clear the Don't Fragment (DF) bit.
- **Hardware-Down**—Interface protocol initialization failed to complete successfully.

- **PFC**—Protocol field compression is enabled for the PPP session.
- **Point-To-Point**—Interface is point-to-point.
- **SNMP-Traps**—SNMP trap notifications are enabled.
- **Up**—Interface is enabled and operational.

Label-Switched Interface Traffic Statistics Field

When you use the **vrf-table-label** statement to configure a VRF routing table, a label-switched interface (LSI) logical interface label is created and mapped to the VRF routing table.

Any routes present in a VRF routing table and configured with the **vrf-table-label** statement are advertised with the LSI logical interface label allocated for the VRF routing table. When packets for this VPN arrive on a core-facing interface, they are treated as if the enclosed IP packet arrived on the LSI interface and are then forwarded and filtered based on the correct table. For more information on the **vrf-table-label** statement, including a list of supported interfaces, see the *Junos VPNs Configuration Guide*.

If you configure the **family mpls** statement at the **[edit interfaces *interface-name* unit *logical-unit-number*]** hierarchy level and you also configure the **vrf-table-label** statement at the **[edit routing-instances *routing-instance-name*]** hierarchy level, the output for the **show interface *interface-name* extensive** command includes the following output fields about the LSI traffic statistics:

- **Input bytes**—Number of bytes entering the LSI and the current throughput rate in bits per second (bps).
- **Input packets**—Number of packets entering the LSI and the current throughput rate in packets per second (pps).

> NOTE: If LSI interfaces are used with VPLS when **no-tunnel-services** is configured or L3VPN when **vrf-table-label** configuration is applied inside the routing-instance, the **Input packets** field associated with the core-facing interfaces may not display the correct value. Only the Input counter is affected because the LSI is used to receive traffic from the remote PEs. Traffic that arrives on an LSI interface might not be counted at both the Traffic Statistics and the Label-switched interface (LSI) traffic statistics levels.
>
> This note applies to the following platforms:
>
> - M Series routers with -E3 FPC model numbers or configured with an Enhanced CFEB (CFEB-E), and M120 routers
> - MX Series routers with DPC or ADPC only

The following example shows the LSI traffic statistics that you might see as part of the output of the **show interface *interface-name* extensive** command:

```
Label-switched interface (LSI) traffic statistics:
  Input bytes:             0            0 bps
  Input packets:           0            0 pps
```

Policer Field

For the logical interface, the **Policer** field provides the policers that are to be evaluated when packets are received or transmitted on the interface. The format is **Policer: Input: *type-fpc/picport*-in-policer, Output: *type-fpc/pic/port*-out-policer**. For example:

```
Policer: Input: at-1/2/0-in-policer, Output: at-2/4/0-out-policer
```

Protocol Field

For the logical interface, the **Protocol** field indicates the protocol family or families that are configured on the interface, displaying one or more of the following values:

- **aenet**—Aggregated Ethernet. Displayed on Fast Ethernet interfaces that are part of an aggregated Ethernet bundle.
- **ccc**—Circuit cross-connect (CCC). Configured on the logical interface of CCC physical interfaces.
- **inet**—IP version 4 (IPv4). Configured on the logical interface for IPv4 protocol traffic, including Open Shortest Path First (OSPF), Border Gateway Protocol (BGP), Internet Control Message Protocol (ICMP), and Internet Protocol Control Protocol (IPCP).
- **inet6**—IP version 6 (IPv6). Configured on the logical interface for IPv6 protocol traffic, including Routing Information Protocol for IPv6 (RIPng), Intermediate System-to-Intermediate System (IS-IS), and BGP.
- **iso**—International Organization for Standardization (ISO). Configured on the logical interface for IS-IS traffic.
- **mlfr-uni-nni**—Multilink Frame Relay (MLFR) FRF.16 user-to-network network-to-network (UNI NNI). Configured on the logical interface for link services bundling.
- **mlfr-end-to-end**—Multilink Frame Relay end-to-end. Configured on the logical interface for multilink bundling.

- **mlppp**—Multilink Point-to-Point Protocol (MLPPP). Configured on the logical interface for multilink bundling.
- **mpls**—Multiprotocol Label Switching (MPLS). Configured on the logical interface for participation in an MPLS path.
- **pppoe**— Point-to-Point Protocol over Ethernet (PPPoE). Configured on Ethernet interfaces enabled to support multiple protocol families.
- **tcc**—Translational cross-connect (TCC). Configured on the logical interface of TCC physical interfaces.
- **tnp**—Trivial Network Protocol (TNP). Used to communicate between the Routing Engine and the router's packet forwarding components. The Junos OS automatically configures this protocol family on the router's internal interfaces only.
- **vpls**—Virtual private LAN service (VPLS). Configured on the logical interface on which you configure VPLS.

RPF Failures Field

For the logical interface, the **RPF Failures** field provides information about the amount of incoming traffic (in packets and bytes) that failed a unicast reverse path forwarding (RPF) check on a particular interface. The format is **RPF Failures: Packets: xx,Bytes: yy**. For example:

```
RPF Failures: Packets: 0, Bytes:0
```

Source Class Field

For the logical interface, the **Source class** field provides the names of source class usage (SCU) counters per family and per class for a particular interface. The counters display packets and bytes arriving from designated user-selected prefixes. For example:

```
                              Packets                Bytes
Source class             (packet-per-second)     (bits-per-second)

                  gold          1928095           161959980
                     (              889)      (      597762)
                bronze                0                   0
                     (                0)      (           0)
                silver                0                   0
```

(	0)	(	0)

Improvements to Interface Transmit Statistics Reporting

The offered load on an interface can be defined as the amount of data the interface is capable of transmitting during a given time period. The actual traffic that goes out of the interface is the transmitted load. However, when outgoing interfaces are oversubscribed, there could be traffic drops in the schedulers attached to the outgoing interfaces. Hence, the offered load is not always the same as the actual transmitted load because the offered load calculation does not take into account possible packet drop or traffic loss.

On MX Series routers, the logical interface-level statistics show the offered load, which is often different from the actual transmitted load. To address this limitation, Junos OS introduces a new configuration option in Release 11.4 R3 and later. The new configuration option, **interface-transmit-statistics**, at the **[edit interface *interface-name*]** hierarchy level, enables you to configure Junos OS to accurately capture and report the transmitted load on interfaces.

When the **interface-transmit-statistics** statement is included at the **[edit interface *interface-name*]** hierarchy level, the following operational mode commands report the actual transmitted load:

- **show interface *interface-name* <detail | extensive>**
- **monitor interface *interface-name***
- **show snmp mib get *objectID.ifIndex***

The **show interface *interface-name*** command also shows whether the **interface-transmit-statistics** configuration is enabled or disabled on the interface.

RELATED DOCUMENTATION

interface-transmit-statistics | 350

show interfaces

show interfaces (PTX Series Packet Transport Routers)

Syntax

```
show interfaces et-fpc/pic/port
<brief | detail | extensive | terse>
<descriptions>
<media>
<snmp-index snmp-index>
<statistics>
```

Release Information

Command introduced in Junos OS Release 8.0.
Command introduced in Junos OS Release 12.1 for PTX Series Packet Transport Routers.

Description

(PTX Series Packet Transport Routers only) Display status information about the specified Ethernet interface.

Options

et-*fpc/pic/port*—Display standard information about the specified Ethernet interface.

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

descriptions—(Optional) Display interface description strings.

media —(Optional) Display media-specific information about network interfaces.

snmp-index *snmp-index*—(Optional) Display information for the specified SNMP index of the interface.

statistics—(Optional) Display static interface statistics.

Required Privilege Level

view

List of Sample Output

show interfaces brief (PTX5000 Packet Transport Router) on page 426
show interfaces extensive (PTX5000 Packet Transport Router) on page 426
show interfaces terse (PTX5000 Packet Transport Router) on page 428
show interfaces extensive (Junos OS Evolved) on page 432

Output Fields

See Table 23 on page 415 for the output fields for the **show interfaces** (PTX Series Packet Transport Routers) command.

Table 23: show interfaces PTX Series Output Fields

Field Name	Field Description	Level of Output
Physical Interface		
Physical interface	Name of the physical interface.	All levels
Enabled	State of the interface. Possible values are described in the “Enabled Field” section under “Common Output Fields Description” on page 404.	All levels
Interface index	Index number of the physical interface, which reflects its initialization sequence.	**detail extensive** none
SNMP ifIndex	SNMP index number for the physical interface.	**detail extensive** none
Generation	Unique number for use by Juniper Networks technical support only.	**detail extensive**
Link-level type	Encapsulation being used on the physical interface.	All levels
MTU	Maximum transmission unit size on the physical interface.	All levels
Speed	Speed at which the interface is running.	All levels
BPDU Error	Bridge protocol data unit (BPDU) errors (if any).	All levels
MAC-Rewrite	MAC Rewrite errors (if any).	All levels
Loopback	Loopback status: **Enabled** or **Disabled**. If loopback is enabled, type of loopback: **Local** or **Remote**.	All levels
Source filtering	Source filtering status: **Enabled** or **Disabled**.	All levels
Flow control	Flow control status: **Enabled** or **Disabled**.	All levels
Device flags	Information about the physical device. Possible values are described in the “Device Flags” section under “Common Output Fields Description” on page 404.	All levels
Interface flags	Information about the interface. Possible values are described in the “Interface Flags” section under “Common Output Fields Description” on page 404.	All levels
Link flags	Information about the link. Possible values are described in the “Links Flags” section under “Common Output Fields Description” on page 404.	All levels

Table 23: show interfaces PTX Series Output Fields *(continued)*

Field Name	Field Description	Level of Output
CoS queues	Number of CoS queues configured.	**detail extensive** none
Hold-times	Current interface hold-time up and hold-time down, in milliseconds.	**detail extensive**
Current address	Configured MAC address.	**detail extensive** none
Hardware address	Hardware MAC address.	**detail extensive** none
Last flapped	Date, time, and how long ago the interface went from down to up. The format is **Last flapped:** ***year-month-day hour:minute:second:timezone*** **(*hour:minute:second* ago)**. For example, **Last flapped: 2002-04-26 10:52:40 PDT (04:33:20 ago).**	**detail extensive** none
Statistics last cleared	Time when the statistics for the interface were last set to zero.	**detail extensive**
Traffic statistics	Number and rate of bytes and packets received and transmitted on the physical interface. • **Input bytes**—Number of bytes received on the interface. • **Output bytes**—Number of bytes transmitted on the interface. • **Input packets**—Number of packets received on the interface. • **Output packets**—Number of packets transmitted on the interface. NOTE: **Input bytes** and **output bytes** are counted as Layer 3 packet length.	**detail extensive**

Table 23: show interfaces PTX Series Output Fields *(continued)*

Field Name	Field Description	Level of Output
Input errors	Input errors on the interface. The following paragraphs explain the counters whose meaning might not be obvious: • **Errors**—Sum of the incoming frame aborts and FCS errors. • **Drops**—Number of packets dropped by the input queue of the I/O Manager ASIC. If the interface is saturated, this number increments once for every packet that is dropped by the ASIC's RED mechanism. • **Framing errors**—Number of packets received with an invalid frame checksum (FCS). • **Runts**—Number of frames received that are smaller than the runt threshold. • **Policed discards**—Number of frames that the incoming packet match code discarded because they were not recognized or not of interest. Usually, this field reports protocols that the Junos OS does not handle. • **L3 incompletes**—Number of incoming packets discarded because they failed Layer 3 (usually IPv4) sanity checks of the header. For example, a frame with less than 20 bytes of available IP header is discarded. L3 incomplete errors can be ignored by configuring the **ignore-l3-incompletes** statement. NOTE: The L3 incompletes field is *not* supported on PTX Series Packet Transport Routers. • **L2 channel errors**—Number of times the software did not find a valid logical interface for an incoming frame. • **L2 mismatch timeouts**—Number of malformed or short packets that caused the incoming packet handler to discard the frame as unreadable. • **FIFO errors**—Number of FIFO errors in the receive direction that are reported by the ASIC on the PIC. If this value is ever nonzero, the PIC is probably malfunctioning. • **Resource errors**—Sum of transmit drops.	**extensive**

Table 23: show interfaces PTX Series Output Fields *(continued)*

Field Name	Field Description	Level of Output
Output errors	Output errors on the interface. The following paragraphs explain the counters whose meaning might not be obvious: • **Carrier transitions**—Number of times the interface has gone from **down** to **up**. This number does not normally increment quickly, increasing only when the cable is unplugged, the far-end system is powered down and then up, or another problem occurs. If the number of carrier transitions increments quickly (perhaps once every 10 seconds), the cable, the far-end system, or the PIC or PIM is malfunctioning. • **Errors**—Sum of the outgoing frame aborts and FCS errors. • **Drops**—Number of packets dropped by the output queue of the I/O Manager ASIC. If the interface is saturated, this number increments once for every packet that is dropped by the ASIC's RED mechanism. • **Collisions**—Number of Ethernet collisions. The Gigabit Ethernet PIC supports only full-duplex operation, so for Gigabit Ethernet PICs, this number should always remain 0. If it is nonzero, there is a software bug. • **Aged packets**—Number of packets that remained in shared packet SDRAM so long that the system automatically purged them. The value in this field should never increment. If it does, it is most likely a software bug or possibly malfunctioning hardware. • **FIFO errors**—Number of FIFO errors in the send direction as reported by the ASIC on the PIC. If this value is ever nonzero, the PIC is probably malfunctioning. • **HS link CRC errors**—Number of errors on the high-speed links between the ASICs responsible for handling the router interfaces. • **MTU errors**—Number of packets whose size exceeded the MTU of the interface. • **Resource errors**—Sum of transmit drops.	**extensive**
Egress queues	Total number of egress queues supported on the specified interface.	**detail extensive**
Queue counters (Egress)	CoS queue number and its associated user-configured forwarding class name. • **Queued packets**—Number of queued packets. • **Transmitted packets**—Number of transmitted packets. • **Dropped packets**—Number of packets dropped by the ASIC's RED mechanism.	**detail extensive**
Ingress queues	Total number of ingress queues supported on the specified interface.	**extensive**

Table 23: show interfaces PTX Series Output Fields *(continued)*

Field Name	Field Description	Level of Output
Queue counters (Ingress)	CoS queue number and its associated user-configured forwarding class name. • **Queued packets**—Number of queued packets. • **Transmitted packets**—Number of transmitted packets. • **Dropped packets**—Number of packets dropped by the ASIC's RED mechanism.	**extensive**
Active alarms and Active defects	Ethernet-specific defects that can prevent the interface from passing packets. When a defect persists for a certain amount of time, it is promoted to an alarm. Based on the router configuration, an alarm can ring the red or yellow alarm bell on the router, or turn on the red or yellow alarm LED on the craft interface. These fields can contain the value **None** or **Link**. • **None**—There are no active defects or alarms. • **Link**—Interface has lost its link state, which usually means that the cable is unplugged, the far-end system has been turned off, or the PIC is malfunctioning. • **LOCAL-FAULT**—Link fault signaling operates between the remote PHY RS (Reconciliation sub-layer) and the local RS. A Local Fault is used to signal a detected fault between the remote RS and the local RS to the local Ethernet interface. • **REMOTE-FAULT**—When the Local Fault status reaches an RS, the RS stops sending MAC data and continuously generates the Remote Fault status on the transmit data path .	**detail extensive** none

Table 23: show interfaces PTX Series Output Fields *(continued)*

Field Name	Field Description	Level of Output
MAC statistics	**Receive** and **Transmit** statistics reported by the PIC's MAC subsystem, including the following: • **Total octets** and **total packets**—Total number of octets and packets. • **Unicast packets, Broadcast packets,** and **Multicast packets**—Number of unicast, broadcast, and multicast packets. • **CRC/Align errors**—Total number of packets received that had a length (excluding framing bits, but including FCS octets) of between 64 and 1518 octets, inclusive, and had either a bad FCS with an integral number of octets (FCS Error) or a bad FCS with a nonintegral number of octets (Alignment Error). • **FIFO error**—Number of FIFO errors that are reported by the ASIC on the PIC. If this value is ever nonzero, the PIC or a cable is probably malfunctioning. • **MAC control frames**—Number of MAC control frames. • **MAC pause frames**—Number of MAC control frames with **pause** operational code. • **Oversized frames**—Number of frames that exceed 1518 octets. • **Jabber frames**—Number of frames that were longer than 1518 octets (excluding framing bits, but including FCS octets), and had either an FCS error or an alignment error. This definition of jabber is different from the definition in IEEE-802.3 section 8.2.1.5 (10BASE5) and section 10.3.1.4 (10BASE2). These documents define jabber as the condition in which any packet exceeds 20 ms. The allowed range to detect jabber is from 20 ms to 150 ms. • **Fragment frames**—Total number of packets that were less than 64 octets in length (excluding framing bits, but including FCS octets), and had either an FCS error or an alignment error. Fragment frames normally increment because both runts (which are normal occurrences caused by collisions) and noise hits are counted. • **VLAN tagged frames**—Number of frames that are VLAN tagged. The system uses the TPID of 0x8100 in the frame to determine whether a frame is tagged or not. • **Code violations**—Number of times an event caused the PHY to indicate “Data reception error” or “invalid data symbol error.”	**extensive**

Table 23: show interfaces PTX Series Output Fields *(continued)*

Field Name	Field Description	Level of Output
Filter statistics	**Receive** and **Transmit** statistics reported by the PIC's MAC address filter subsystem. The filtering is done by the content-addressable memory (CAM) on the PIC. The filter examines a packet's source and destination MAC addresses to determine whether the packet should enter the system or be rejected. • **Input packet count**—Number of packets received from the MAC hardware that the filter processed. • **Input packet rejects**—Number of packets that the filter rejected because of either the source MAC address or the destination MAC address. • **Input DA rejects**—Number of packets that the filter rejected because the destination MAC address of the packet is not on the accept list. It is normal for this value to increment. When it increments very quickly and no traffic is entering the router from the far-end system, either there is a bad ARP entry on the far-end system, or multicast routing is not on and the far-end system is sending many multicast packets to the local router (which the router is rejecting). • **Input SA rejects**—Number of packets that the filter rejected because the source MAC address of the packet is not on the accept list. The value in this field should increment only if source MAC address filtering has been enabled. If filtering is enabled, if the value increments quickly, and if the system is not receiving traffic that it should from the far-end system, it means that the user-configured source MAC addresses for this interface are incorrect. • **Output packet count**—Number of packets that the filter has given to the MAC hardware. • **Output packet pad count**—Number of packets the filter padded to the minimum Ethernet size (60 bytes) before giving the packet to the MAC hardware. Usually, padding is done only on small ARP packets, but some very small IP packets can also require padding. If this value increments rapidly, either the system is trying to find an ARP entry for a far-end system that does not exist or it is misconfigured. • **Output packet error count**—Number of packets with an indicated error that the filter was given to transmit. These packets are usually aged packets or are the result of a bandwidth problem on the FPC hardware. On a normal system, the value of this field should not increment. • **CAM destination filters, CAM source filters**—Number of entries in the CAM dedicated to destination and source MAC address filters. There can only be up to 64 source entries. If source filtering is disabled, which is the default, the values for these fields should be 0.	**extensive**

Table 23: show interfaces PTX Series Output Fields *(continued)*

Field Name	Field Description	Level of Output

Autonegotiation information	Information about link autonegotiation.
	• **Negotiation status:**
	  ◦ **Incomplete**—Ethernet interface has the speed or link mode configured.
	  ◦ **No autonegotiation**—Remote Ethernet interface has the speed or link mode configured, or does not perform autonegotiation.
	  ◦ **Complete**—Ethernet interface is connected to a device that performs autonegotiation and the autonegotiation process is successful.
	• **Link partner status**—**OK** when Ethernet interface is connected to a device that performs autonegotiation and the autonegotiation process is successful.
	• **Link partner:**
	  ◦ **Link mode**—Depending on the capability of the attached Ethernet device, either **Full-duplex** or **Half-duplex**.
	  ◦ **Flow control**—Types of flow control supported by the remote Ethernet device. For Fast Ethernet interfaces, the type is **None**. For Gigabit Ethernet interfaces, types are **Symmetric** (link partner supports **PAUSE** on receive and transmit), **Asymmetric** (link partner supports **PAUSE** on transmit), and **Symmetric/Asymmetric** (link partner supports both **PAUSE** on receive and transmit or only **PAUSE** receive).
	  ◦ **Remote fault**—Remote fault information from the link partner—**Failure** indicates a receive link error. **OK** indicates that the link partner is receiving. **Negotiation error** indicates a negotiation error. **Offline** indicates that the link partner is going offline.
	• **Local resolution**—Information from the link partner:
	  ◦ **Flow control**—Types of flow control supported by the remote Ethernet device. For Gigabit Ethernet interfaces, types are **Symmetric** (link partner supports **PAUSE** on receive and transmit), **Asymmetric** (link partner supports **PAUSE** on transmit), and **Symmetric/Asymmetric** (link partner supports both **PAUSE** on receive and transmit or only **PAUSE** receive).
	  ◦ **Remote fault**—Remote fault information. **Link OK** (no error detected on receive), **Offline** (local interface is offline), and **Link Failure** (link error detected on receive).
		extensive

Table 23: show interfaces PTX Series Output Fields *(continued)*

Field Name	Field Description	Level of Output
Packet Forwarding Engine configuration	Information about the configuration of the Packet Forwarding Engine: • **Destination slot**—FPC slot number.	**extensive**
CoS information	Information about the CoS queue for the physical interface. • **CoS transmit queue**—Queue number and its associated user-configured forwarding class name. • **Bandwidth %**—Percentage of bandwidth allocated to the queue. • **Bandwidth bps**—Bandwidth allocated to the queue (in bps). • **Buffer %**—Percentage of buffer space allocated to the queue. • **Buffer usec**—Amount of buffer space allocated to the queue, in microseconds. This value is nonzero only if the buffer size is configured in terms of time. • **Priority**—Queue priority: **low** or **high**. • **Limit**—Displayed if rate limiting is configured for the queue. Possible values are **none** and **exact**. If **exact** is configured, the queue transmits only up to the configured bandwidth, even if excess bandwidth is available. If **none** is configured, the queue transmits beyond the configured bandwidth if bandwidth is available.	**extensive**
Logical Interface		
Logical interface	Name of the logical interface.	All levels
Index	Index number of the logical interface, which reflects its initialization sequence.	**detail extensive** none
SNMP ifIndex	SNMP interface index number for the logical interface.	**detail extensive** none
Generation	Unique number for use by Juniper Networks technical support only.	**detail extensive**
Flags	Information about the logical interface. Possible values are described in the “Logical Interface Flags” section under “Common Output Fields Description” on page 404.	All levels

Table 23: show interfaces PTX Series Output Fields *(continued)*

Field Name	Field Description	Level of Output
VLAN-Tag	Rewrite profile applied to incoming or outgoing frames on the outer (**Out**) VLAN tag or for both the outer and inner (**In**) VLAN tags. • **push**—An outer VLAN tag is pushed in front of the existing VLAN tag. • **pop**—The outer VLAN tag of the incoming frame is removed. • **swap**—The outer VLAN tag of the incoming frame is overwritten with the user-specified VLAN tag information. • **push**—An outer VLAN tag is pushed in front of the existing VLAN tag. • **push-push**—Two VLAN tags are pushed in from the incoming frame. • **swap-push**—The outer VLAN tag of the incoming frame is replaced by a user-specified VLAN tag value. A user-specified outer VLAN tag is pushed in front. The outer tag becomes an inner tag in the final frame. • **swap-swap**—Both the inner and the outer VLAN tags of the incoming frame are replaced by the user-specified VLAN tag value. • **pop-swap**—The outer VLAN tag of the incoming frame is removed, and the inner VLAN tag of the incoming frame is replaced by the user-specified VLAN tag value. The inner tag becomes the outer tag in the final frame. • **pop-pop**—Both the outer and inner VLAN tags of the incoming frame are removed.	**brief detail extensive** none
Demux	IP demultiplexing (demux) value that appears if this interface is used as the demux underlying interface. The output is one of the following: • Source Family Inet • Destination Family Inet	**detail extensive** none
Encapsulation	Encapsulation on the logical interface.	All levels
Protocol	Protocol family. Possible values are described in the “Protocol Field” section under “Common Output Fields Description” on page 404.	**detail extensive** none
MTU	Maximum transmission unit size on the logical interface.	**detail extensive** none
Maximum labels	Maximum number of MPLS labels configured for the MPLS protocol family on the logical interface.	**detail extensive** none

Table 23: show interfaces PTX Series Output Fields *(continued)*

Field Name	Field Description	Level of Output
Traffic statistics	Number and rate of bytes and packets received and transmitted on the specified interface set. • **Input bytes, Output bytes**—Number of bytes received and transmitted on the interface set • **Input packets, Output packets**—Number of packets received and transmitted on the interface set. NOTE: **Input bytes** and **output bytes** are counted as Layer 3 packet length.	**detail extensive**
IPv6 transit statistics	Number of IPv6 transit bytes and packets received and transmitted on the logical interface if IPv6 statistics tracking is enabled.	**extensive**
Local statistics	Number and rate of bytes and packets destined to the router.	**extensive**
Transit statistics	Number and rate of bytes and packets transiting the switch.	**extensive**
Generation	Unique number for use by Juniper Networks technical support only.	**detail extensive**
Route Table	Route table in which the logical interface address is located. For example, **0** refers to the routing table inet.0.	**detail extensive** none
Flags	Information about protocol family flags. Possible values are described in the "Family Flags" section under "Common Output Fields Description" on page 404.	**detail extensive**
Donor interface	(Unnumbered Ethernet) Interface from which an unnumbered Ethernet interface borrows an IPv4 address.	**detail extensive** none
Preferred source address	(Unnumbered Ethernet) Secondary IPv4 address of the donor loopback interface that acts as the preferred source address for the unnumbered Ethernet interface.	**detail extensive** none
Input Filters	Names of any input filters applied to this interface. If you specify a precedence value for any filter in a dynamic profile, filter precedence values appear in parentheses next to all interfaces.	**detail extensive**
Output Filters	Names of any output filters applied to this interface. If you specify a precedence value for any filter in a dynamic profile, filter precedence values appear in parentheses next to all interfaces.	**detail extensive**

Table 23: show interfaces PTX Series Output Fields *(continued)*

Field Name	Field Description	Level of Output
Mac-Validate Failures	Number of MAC address validation failures for packets and bytes. This field is displayed when MAC address validation is enabled for the logical interface.	**detail extensive** none
Addresses, Flags	Information about the address flags. Possible values are described in the “Addresses Flags” section under “Common Output Fields Description” on page 404.	**detail extensive** none
protocol-family	Protocol family configured on the logical interface. If the protocol is **inet**, the IP address of the interface is also displayed.	**brief**
Flags	Information about flags (possible values are described in the “Addresses Flags” section under “Common Output Fields Description” on page 404.	**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** none
Broadcast	Broadcast address of the logical interlace.	**detail extensive** none
Generation	Unique number for use by Juniper Networks technical support only.	**detail extensive**

Sample Output

show interfaces brief (PTX5000 Packet Transport Router)

```
user@host> show interfaces brief et-7/0/0
```

```
Physical interface: et-7/0/0, Enabled, Physical link is Up
  Link-level type: Ethernet, MTU: 1514, Speed: 10Gbps, Loopback: Disabled, Source
 filtering: Disabled, Flow control: Enabled
  Device flags   : Present Running
  Interface flags: SNMP-Traps Internal: 0x4000
  Link flags     : None
```

show interfaces extensive (PTX5000 Packet Transport Router)

`user@host>` **show interfaces et-7/0/0 extensive**

```
Physical interface: et-7/0/0, Enabled, Physical link is Up
  Interface index: 168, SNMP ifIndex: 501, Generation: 171
  Link-level type: Ethernet, MTU: 1514, Speed: 10Gbps, BPDU Error: None, MAC-REWRITE
 Error: None, Loopback: Disabled, Source filtering: Disabled, Flow control: Enabled

  Device flags   : Present Running
  Interface flags: SNMP-Traps Internal: 0x4000
  Link flags     : None
  CoS queues     : 8 supported, 8 maximum usable queues
  Hold-times     : Up 0 ms, Down 0 ms
  Current address: 88:e0:f3:3b:de:43, Hardware address: 88:e0:f3:3b:de:43
  Last flapped   : 2012-01-18 11:48:24 PST (01:51:00 ago)
  Statistics last cleared: 2012-01-18 13:38:54 PST (00:00:30 ago)
  Traffic statistics:
   Input  bytes  :                    0                    0 bps
   Output bytes  :                    0                    0 bps
   Input  packets:                    0                    0 pps
   Output packets:                    0                    0 pps
   IPv6 transit statistics:
    Input  bytes  :                   0
    Output bytes  :                   0
    Input  packets:                   0
    Output packets:                   0
  Input errors:
    Errors: 0, Drops: 0, Framing errors: 0, Runts: 0, Policed discards: 0, L3
incompletes: 0, L2 channel errors: 0, L2 mismatch timeouts: 0, FIFO errors: 0,
Resource errors: 0
  Output errors:
    Carrier transitions: 0, Errors: 0, Drops: 0, Collisions: 0, Aged packets: 0,
FIFO errors: 0, HS link CRC errors: 0, MTU errors: 0, Resource errors: 0
  Egress queues: 8 supported, 4 in use
  Queue counters:       Queued packets  Transmitted packets      Dropped packets
    0 best-effort                    0                    0                    0
    1 expedited-fo                   0                    0                    0
    2 assured-forw                   0                    0                    0
    3 network-cont                   0                    0                    0
  Queue number:         Mapped forwarding classes
    0                   best-effort
    1                   expedited-forwarding
    2                   assured-forwarding
    3                   network-control
  Active alarms  : None
  Active defects : None
  MAC statistics:                      Receive         Transmit
```

```
    Total octets                         0                 0
    Total packets                        0                 0
    Unicast packets                      0                 0
    Broadcast packets                    0                 0
    Multicast packets                    0                 0
    CRC/Align errors                     0                 0
    FIFO errors                          0                 0
    MAC control frames                   0                 0
    MAC pause frames                     0                 0
    Oversized frames                     0
    Jabber frames                        0
    Fragment frames                      0
    VLAN tagged frames                   0
    Code violations                      0
  Filter statistics:
    Input packet count                   0
    Input packet rejects                 0
    Input DA rejects                     0
    Input SA rejects                     0
    Output packet count                                    0
    Output packet pad count                                0
    Output packet error count                              0
    CAM destination filters: 0, CAM source filters: 0
  Autonegotiation information:
    Negotiation status: Incomplete
  Packet Forwarding Engine configuration:
    Destination slot: 7
  CoS information:
    Direction : Output
    CoS transmit queue               Bandwidth               Buffer Priority
Limit
                               %             bps     %             usec
    0 best-effort             95      9500000000    95                0       low
none
    3 network-control          5       500000000     5                0       low
none
  Interface transmit statistics: Disabled
```

show interfaces terse (PTX5000 Packet Transport Router)

`user@host> show interfaces terse`

```
Interface                Admin Link Proto    Local                 Remote
et-2/0/0                 up    up
```

```
et-2/0/1                up    up
et-2/0/2                up    up
et-2/0/3                up    up
et-2/0/4                up    up
et-2/0/5                up    down
et-2/0/6                up    up
et-2/0/7                up    up
et-2/0/8                up    up
et-2/0/9                up    down
et-2/0/10               up    up
et-2/0/11               up    up
et-2/0/12               up    up
et-2/0/13               up    down
et-2/0/14               up    up
et-2/0/15               up    up
et-2/0/16               up    up
et-2/0/17               up    down
et-2/0/18               up    down
et-2/0/19               up    up
et-2/0/20               up    down
et-2/0/21               up    up
et-2/0/22               up    down
et-2/0/23               up    up
et-2/1/0                up    up
et-2/1/1                up    up
et-2/1/2                up    up
et-2/1/3                up    up
et-2/1/4                up    up
et-2/1/5                up    up
et-2/1/6                up    up
et-2/1/7                up    up
et-2/1/8                up    up
et-2/1/9                up    up
et-2/1/10               up    up
et-2/1/11               up    up
et-2/1/12               up    up
et-2/1/13               up    up
et-2/1/14               up    up
et-2/1/15               up    up
et-2/1/16               up    up
et-2/1/17               up    up
et-2/1/18               up    up
et-2/1/19               up    up
et-2/1/20               up    up
```

```
et-2/1/21               up    up
et-2/1/22               up    up
et-2/1/23               up    up
et-5/0/0                up    up
et-5/0/0.0              up    up   ccc
et-5/0/0.32767          up    up   multiservice
et-5/0/1                up    up
et-5/0/2                up    up
et-5/0/3                up    down
et-5/0/4                up    down
et-5/0/5                up    up
et-5/0/5.0              up    up   ccc
et-5/0/5.32767          up    up   multiservice
et-5/0/6                up    up
et-5/0/7                up    up
et-5/0/8                up    down
et-5/0/9                up    up
et-5/0/10               up    up
et-5/0/11               up    up
et-5/0/12               up    up
et-5/0/13               up    down
et-5/0/14               up    down
et-5/0/15               up    up
et-5/0/16               up    up
et-5/0/17               up    up
et-5/0/18               up    up
et-5/0/19               up    up
et-5/0/20               up    down
et-5/0/21               up    down
et-5/0/22               up    up
et-5/0/23               up    up
et-5/1/0                up    up
et-5/1/1                up    up
et-7/0/0                up    up
et-7/0/1                up    up
et-7/0/2                up    up
et-7/0/3                up    up
et-7/0/4                up    up
et-7/0/5                up    up
et-7/0/6                up    up
et-7/0/7                up    up
et-7/0/8                up    up
et-7/0/9                up    up
et-7/0/10               up    down
```

```
et-7/0/11               up    down
et-7/0/12               up    down
et-7/0/13               up    down
et-7/0/14               up    down
et-7/0/15               up    down
et-7/0/16               up    down
et-7/0/17               up    down
et-7/0/18               up    down
et-7/0/19               up    down
et-7/0/20               up    down
et-7/0/21               up    down
et-7/0/22               up    down
et-7/0/23               up    down
dsc                     up    up
em0                     up    up
em0.0                   up    up   inet     192.168.177.61/25
gre                     up    up
ipip                    up    up
ixgbe0                  up    up
ixgbe0.0                up    up   inet     10.0.0.4/8
                                            128.0.0.1/2
                                            128.0.0.4/2
                                   inet6    fe80::200:ff:fe00:4/64
                                            fec0::a:0:0:4/64
                                   tnp      0x4
ixgbe1                  up    up
ixgbe1.0                up    up   inet     10.0.0.4/8
                                            128.0.0.1/2
                                            128.0.0.4/2
                                   inet6    fe80::200:1ff:fe00:4/64
                                            fec0::a:0:0:4/64
                                   tnp      0x4
lo0                     up    up
lo0.0                   up    up   inet     10.255.177.61       --> 0/0
                                            127.0.0.1           --> 0/0
                                   iso
47.0005.80ff.f800.0000.0108.0001.0102.5517.7061
                                   inet6    abcd::10:255:177:61
                                            fe80::ee9e:cd0f:fc02:b01e
lo0.16384               up    up   inet     127.0.0.1           --> 0/0
lo0.16385               up    up   inet
lsi                     up    up
mtun                    up    up
pimd                    up    up
```

```
pime                    up    up
tap                     up    up
```

show interfaces extensive (Junos OS Evolved)

```
user@host> show interfaces et-0/0/0 extensive
```

```
Physical interface: et-0/0/0, Enabled, Physical link is Up
  Interface index: 1002, SNMP ifIndex: 505, Generation: 113
  Link-level type: Ethernet, MTU: 1514, LAN-PHY mode, Speed: 100Gbps, BPDU Error:
 None, MAC-REWRITE Error: None, Loopback: Disabled, Source filtering: Disabled,
  Flow control: Enabled
  Device flags   : Present Running
  Interface flags: SNMP-Traps
  Link flags     : None
  CoS queues     : 8 supported, 8 maximum usable queues
  Hold-times     : Up 0 ms, Down 0 ms
  Damping        : half-life: 0 sec, max-suppress: 0 sec, reuse: 0, suppress: 0,
state: unsuppressed
  Current address: 88:e0:f3:3b:de:43, Hardware address: 88:e0:f3:3b:de:43
  Last flapped   : Never
  Statistics last cleared: Never
  Traffic statistics:
   Input  bytes  :                   0                    0 bps
   Output bytes  :                   0                    0 bps
   Input  packets:                   0                    0 pps
   Output packets:                   0                    0 pps
   IPv6 transit statistics:
    Input  bytes  :                  0
    Output bytes  :                  0
    Input  packets:                  0
    Output packets:                  0
  Input errors:
    Errors: 0, Drops: 0, Framing errors: 0, Runts: 0, Policed discards: 0, L3
incompletes: 0, L2 channel errors: 0, L2 mismatch timeouts: 0, FIFO errors: 0,
    Resource errors: 0
  Output errors:
    Carrier transitions: 0, Errors: 0, Drops: 0, Collisions: 0, Aged packets: 0,
FIFO errors: 0, HS link CRC errors: 0, MTU errors: 0, Resource errors: 0
  Egress queues: 8 supported, 4 in use
  Queue counters:       Queued packets  Transmitted packets     Dropped packets
    0             16045690984503098046                    0                   0
    1             16045690984503098046                    0                   0
    2             16045690984503098046                    0                   0
```

```
  3             16045690984503098046                    0                   0
Queue number:         Mapped forwarding classes
  0                   best-effort
  1                   expedited-forwarding
  2                   assured-forwarding
  3                   network-control
Active alarms  : None
Active defects : None
PCS statistics                      Seconds
  Bit errors                             0
  Errored blocks                         0
MAC statistics:                      Receive         Transmit
  Total octets                             0                0
  Total packets                            0                0
  Unicast packets                          0                0
  Broadcast packets                        0                0
  Multicast packets                        0                0
  CRC/Align errors                         0                0
  FIFO errors                              0                0
  MAC control frames                       0                0
  MAC pause frames                         0                0
  Oversized frames                         0
  Jabber frames                            0
  Fragment frames                          0
  VLAN tagged frames                       0
  Code violations                          0
  Total errors                             0                0
Filter statistics:
  Input packet count                       0
  Input packet rejects                     0
  Input DA rejects                         0
  Input SA rejects                         0
  Output packet count                                       0
  Output packet pad count                                   0
  Output packet error count                                 0
  CAM destination filters: 0, CAM source filters: 0
```

show interfaces media

Syntax

```
show interfaces media
```

Release Information

Command introduced before Junos OS Release 7.4.
Command introduced on PTX Series Packet Transport Routers for Junos OS Release 12.1.

Description

Display media-specific information about all configured network interfaces.

> **NOTE:** **show interfaces media** lists details for all interfaces, whereas **show interfaces media** *interface-name* lists details only for the specified interface.

Options

This command has no options.

Additional Information

Output from both the **show interfaces *interface-name* detail** and the **show interfaces *interface-name* extensive** commands includes all the information displayed in the output from the **show interfaces media** command.

Required Privilege Level

view

List of Sample Output

show interfaces media (SONET/SDH) on page 435
show interfaces media (MX Series Routers) on page 435
show interfaces media (PTX Series Packet Transport Routers) on page 436

Output Fields

The output from the **show interfaces media** command includes fields that display interface media-specific information. These fields are also included in the **show interfaces *interface-name*** command for each particular interface type, and the information provided in the fields is unique to each interface type.

One field unique to the **show interfaces media** command is interface-type errors (for example, SONET errors). This field appears for channelized E3, channelized T3, channelized OC, E1, E3, SONET, T1, and T3 interfaces. The information provided in this output field is also provided in the output from the **show interfaces *interface-name*** command. (For example, for SONET interfaces, these fields are SONET section, SONET line, and SONET path). For a description of errors, see the chapter with the particular interface type in which you are interested.

Sample Output

show interfaces media (SONET/SDH)

The following example displays the output fields unique to the **show interfaces media** command for a SONET interface (with no level of output specified):

`user@host>` **show interfaces media so-4/1/2**

```
Physical interface: so-4/1/2, Enabled, Physical link is Up
  Interface index: 168, SNMP ifIndex: 495
  Link-level type: PPP, MTU: 4474, Clocking: Internal, SONET mode, Speed: OC48,
  Loopback: None, FCS: 16, Payload scrambler: Enabled
  Device flags   : Present Running
  Interface flags: Point-To-Point SNMP-Traps 16384
  Link flags     : Keepalives
  Keepalive settings: Interval 10 seconds, Up-count 1, Down-count 3
  Keepalive: Input: 1783 (00:00:00 ago), Output: 1786 (00:00:08 ago)
  LCP state: Opened
  NCP state: inet: Not-configured, inet6: Not-configured, iso: Not-configured,
  mpls: Not-configured
  CHAP state: Not-configured
  CoS queues     : 8 supported
  Last flapped   : 2005-06-15 12:14:59 PDT (04:31:29 ago)
  Input rate     : 0 bps (0 pps)
  Output rate    : 0 bps (0 pps)
  SONET alarms   : None
  SONET defects  : None
  SONET errors:
     BIP-B1: 121, BIP-B2: 916, REI-L: 0, BIP-B3: 137, REI-P: 16747, BIP-BIP2: 0
  Received path trace: routerb so-1/1/2
  Transmitted path trace: routera so-4/1/2
```

show interfaces media (MX Series Routers)

`user@host>`**show interfaces media xe-0/0/0**

```
Physical interface: xe-0/0/0, Enabled, Physical link is Up
  Interface index: 145, SNMP ifIndex: 592
  Link-level type: Ethernet, MTU: 1514, LAN-PHY mode, Speed: 10Gbps, BPDU Error:
```

```
None,
  Loopback: None, Source filtering: Disabled, Flow control: Enabled
  Pad to minimum frame size: Enabled
  Device flags   : Present Running
  Interface flags: Hardware-Down SNMP-Traps Internal: 0x0
  Link flags     : None
  CoS queues     : 8 supported, 8 maximum usable queues
  Current address: 08:81:f4:82:a3:f0, Hardware address: 08:81:f4:82:a3:f0
  Last flapped   : 2013-10-26 03:20:40 test (1w6d 00:19 ago)
  Input rate     : 0 bps (0 pps)
  Output rate    : 0 bps (0 pps)
  Active alarms  : LINK
  Active defects : LINK
  PCS statistics                      Seconds
    Bit errors                            78
    Errored blocks                        78
  MAC statistics:
    Input bytes: 0, Input packets: 0, Output bytes: 0, Output packets: 0
  Filter statistics:
    Filtered packets: 0, Padded packets: 0, Output packet errors: 0
  Interface transmit statistics: Disabled
```

show interfaces media (PTX Series Packet Transport Routers)

`user@host>` **show interfaces media em0**

```
Physical interface: em0, Enabled, Physical link is Up
      Interface index: 8, SNMP ifIndex: 0
      Type: Ethernet, Link-level type: Ethernet, MTU: 1514, Speed: 1000mbps
      Device flags   : Present Running
      Interface flags: SNMP-Traps
      Link type      : Full-Duplex
      Current address: 00:80:f9:25:00:1b, Hardware address: 00:80:f9:25:00:1b
      Last flapped   : Never
    Input packets : 215151
    Output packets: 72
```

show interfaces terse

Syntax

```
show interfaces terse
```

Release Information

Command introduced before Junos OS Release 7.4.
Command introduced on PTX Series Packet Transport Routers for Junos OS Release 12.1.

Description

Display summary information about interfaces.

Options

This command has no options.

Additional Information

Interfaces are always displayed in numerical order, from the lowest to the highest FPC slot number. Within that slot, the lowest PIC slot is shown first. On an individual PIC, the lowest port number is always first.

Required Privilege Level

view

RELATED DOCUMENTATION

Setting Up Logical Systems

List of Sample Output

show interfaces terse on page 438
show interfaces terse (TX Matrix Plus Router) on page 439
show interfaces terse (PTX Series Packet Transport Routers) on page 440

Output Fields

Table 24 on page 437 lists the output fields for the **show interfaces terse** command. Output fields are listed in the approximate order in which they appear.

Table 24: show interfaces terse Output Fields

Field Name	Field Description
Interface	Interface name.
Admin	Whether the interface is turned on (up) or off (down).

Table 24: show interfaces terse Output Fields *(continued)*

Field Name	Field Description
Link	Link state: **up** or **down**.
Proto	Protocol family configured on the logical interface. A logical interface on a router that supports Ethernet OAM always shows the multiservice protocol.
Local	Local IP address of the logical interface.
Remote	Remote IP address of the logical interface.

Sample Output

show interfaces terse

user@host> **show interfaces terse**

```
Interface               Admin Link Proto    Local                 Remote
t1-0/1/0:0              up    up
t1-0/1/0:0.0            up    up   inet     192.168.220.18/30
t1-0/1/0:1              up    up
t1-0/1/0:2              up    up
t1-0/1/0:3              up    up
at-1/0/0                up    up
at-1/0/1                up    up
dsc                     up    up
fxp0                    up    up
fxp0.0                  up    up   inet     192.168.71.249/21
fxp1                    up    up
fxp1.0                  up    up   inet     10.0.0.4/8
                                   tnp      4
gre                     up    up
ipip                    up    up
lo0                     up    up
lo0.0                   up    up   inet     10.0.1.4            --> 0/0
                                            127.0.0.1           --> 0/0
lo0.16385               up    up   inet

lsi                     up    up
```

```
mtun                    up    up
```

show interfaces terse (TX Matrix Plus Router)

`user@host>` **show interfaces terse**

```
Interface               Admin Link Proto    Local                 Remote
xe-0/0/0                up    up
xe-0/0/1                up    up
xe-0/0/2                up    up
xe-0/0/3                up    up
xe-6/0/0                up    up
xe-6/0/1                up    up
xe-6/0/2                up    up
xe-6/0/3                up    up
xe-6/1/0                up    up
xe-6/1/1                up    up
xe-6/1/2                up    up
xe-6/1/3                up    up
so-0/0/0                up    up
so-0/0/0.0              up    up   inet     1.1.1.1/30
ge-1/3/0.0              up    up   inet     --> 0/0
ge-7/0/0                up    up
ge-7/0/0.0              up    up   inet     2.15.1.1/30
ge-7/0/0.1              up    up   inet     2.15.1.5/30
ge-7/0/0.2              up    up   inet     2.15.1.9/30
ge-7/0/0.3              up    up   inet     2.15.1.13/30
ge-7/0/0.4              up    up   inet     2.15.1.17/30
ge-7/0/0.5              up    up   inet     2.15.1.21/30
...
em0                     up    up
em0.0                   up    up   inet     192.168.178.11/25
gre                     up    up
ipip                    up    up
ixgbe0                  up    up
ixgbe0.0                up    up   inet     10.34.0.4/8
                                            162.0.0.4/2
                                   inet6    fe80::200:ff:fe22:4/64
                                            fec0::a:22:0:4/64
                                   tnp      0x22000004
ixgbe1                  up    up
ixgbe1.0                up    up   inet     10.34.0.4/8
```

```
                                           162.0.0.4/2
                                  inet6    fe80::200:1ff:fe22:4/64
                                           fec0::a:22:0:4/64
                                  tnp      0x22000004
```

show interfaces terse (PTX Series Packet Transport Routers)

`user@host>` **show interfaces em0 terse**

```
Interface               Admin Link Proto    Local                 Remote
 em0                     up    up
 em0.0                   up    up   inet     192.168.3.30/24
```

6

CHAPTER

Protocol-Independent Routing Operational Commands

show route match-prefix

Syntax

```
show route match-prefix match-prefix;
```

Release Information

Command introduced in Junos OS Release 11.4.

Description

Allows you to search for routes using regular expressions based on the extended (modern) regular expressions as defined in POSIX 1003.2.

Options

match-prefix—Regular expression to match formatted prefix.

Additional Information

Required Privilege Level

view

RELATED DOCUMENTATION

> *Regular Expressions for Allowing and Denying Junos OS Operational Mode Commands, Configuration Statements, and Hierarchies*

List of Sample Output

Output Fields

For information about output fields, see the output field tables for the **show route** command, the **show route detail** command, the **show route extensive** command, or the **show route terse** command.

Sample Output

show route match-prefix *:10.255.2.200:6:* (Show all routes matching route distributor 10.255.2.200:6)

```
user@host> show route match-prefix *:10.255.2.200:6:*
```

show route match-prefix 7* (Show all mvpn type-7 routes)

`user@host>` **show route table blue.mvpn.0 match-prefix 7***

```
Paste
router command output here
```

show route match-prefix *:224.* (Show all routes matching group 224/4)

`user@host>` **show route match-prefix *:224.***