Juniper Networks | Engineering Simplicity

Junos® OS Evolved

Interfaces Fundamentals for Junos OS Evolved

Published
2022-03-14

Junos OS Evolved

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

Juniper Networks, the Juniper Networks logo, Juniper, and Junos are registered trademarks of Juniper Networks, Inc. in the United States and other countries. All other trademarks, service marks, registered marks, or registered service marks are the property of their respective owners.

Juniper Networks assumes no responsibility for any inaccuracies in this document. Juniper Networks reserves the right to change, modify, transfer, or otherwise revise this publication without notice.

Junos® OS Evolved Interfaces Fundamentals for Junos OS Evolved
Copyright © 2022 Juniper Networks, Inc. All rights reserved.

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

YEAR 2000 NOTICE

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

END USER LICENSE AGREEMENT

The Juniper Networks product that is the subject of this technical documentation consists of (or is intended for use with) Juniper Networks software. Use of such software is subject to the terms and conditions of the End User License Agreement ("EULA") posted at 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

About This Guide | viii

1 **Device Interfaces**

Device Interfaces Overview | 2

Device Interfaces Overview | 2

Types of Interfaces | 3

Interface Naming Overview | 3

Interface Descriptors Overview | 7

Physical Part of an Interface Name | 9

> Interface Names for ACX Series, PTX Series, and QFX Series Devices | 9

Displaying Interface Configurations | 9

Interface Encapsulations Overview | 10

Understanding Internal Ethernet Interfaces | 12

Physical Interface Properties | 12

Physical Interface Properties Overview | 13

Configure the Interface Description | 13

How to Specify an Aggregated Interface | 14

Interface Speed | 15

> Configuring the Interface Speed on Ethernet Interfaces | 15
>
> Configure Interface Speeds for PTX10003 Routers and QFX5220 Switches | 16
>
> Configure the Aggregated Ethernet Link Speed | 17

Forward Error Correction (FEC) | 19

> Benefits of FEC | 19
>
> Overview | 20
>
> Configure FEC | 20

Interface Aliases | 21

Interface Encapsulation on Physical Interfaces | 22

Encapsulation Capabilities | 23

Configure Encapsulation on a Physical Interface | 24

Configure Interface Encapsulation on PTX Series Routers | 24

Enable SNMP Notifications on Physical Interfaces | 26

Accounting for Physical Interfaces | 27

Overview | 27

Configure an Accounting Profile for a Physical Interface | 27

How to Display the Accounting Profile | 29

Disable a Physical Interface | 30

How to Disable a Physical Interface | 31

Example: Disable a Physical Interface | 32

Media MTU and Protocol MTU | 34

Media MTU Overview | 34

Configure the Media MTU | 35

Protocol MTU | 36

Configure the Protocol MTU | 36

Encapsulation Overhead by Interface Encapsulation Type | 37

Interface Ranges for Physical Interfaces | 39

Configure Interface Ranges | 39

Expanded Interface Range Statements | 42

Configuration Inheritance Priority | 44

Configuration Inheritance for Member Interfaces | 45

Common Configuration Inheritance | 46

Configuration Group Inheritance | 47

Configuration Expansion Where Interface Range Is Used | 48

Damping Physical Interfaces | 50

Physical Interface Damping Overview | 50

Configure Damping of Shorter Physical Interface Transitions | 59

Configure Damping of Longer Physical Interface Transitions | 60

Logical Interface Properties | 61

Logical Interface Properties Overview | 62

Specify the Logical Interface Number | 62

Add a Logical Unit Description to the Configuration | 63

Configure the Interface Bandwidth | 63

Configure Interface Encapsulation on Logical Interfaces | 64

Understand the Interface Encapsulation on Logical Interfaces | 65

Configure the Encapsulation on a Logical Interface | 65

Display the Encapsulation on a Logical Interface | 66

Configure Interface Encapsulation on PTX Series Routers | 67

Overview of Accounting for the Logical Interface | 68

Accounting Profiles Overview | 69

Configure Accounting for the Logical Interface | 69

Introduction to Displaying the Accounting Profile for the Logical Interface | 70

Enable or Disable SNMP Notifications on Logical Interfaces | 72

Disable a Logical Interface | 72

Protocol Family and Interface Address Properties | 74

Configure the Protocol Family | 74

Assign the Interface Address | 75

Configure Default, Primary, and Preferred Addresses and Interfaces | 77

Default, Primary, and Preferred Addresses and Interfaces | 77

Configure the Primary Interface for the Router | 77

Configure the Primary Address for an Interface | 78

Configure the Preferred Address for an Interface | 78

Operational Behavior of Interfaces with the Same IPv4 Address | 79

Configure Unnumbered Interfaces: Overview | 83

Configure an Unnumbered Point-to-Point Interface | 83

Configure an Unnumbered Ethernet or Demux Interface | **84**

Configure a Secondary Address as a Preferred Source Address for Unnumbered Ethernet or Demux Interfaces | **85**

Restrictions for Unnumbered Ethernet Interface Configurations | **86**

Example: Display the Unnumbered Ethernet Interface Configuration | **87**

Example: Display the Configured Preferred Source Address for an Unnumbered Ethernet Interface | **88**

Example: Display the Configuration for the Unnumbered Ethernet Interface as the Next Hop for a Static Route | **90**

Protocol MTU | **91**

Configure the Protocol MTU | **92**

Disable the Removal of Address and Control Bytes | **93**

Disable the Transmission of Redirect Messages on an Interface | **93**

Apply a Filter to an Interface | **94**

Define Interface Groups in Firewall Filters | **94**

Apply a Filter to an Interface | **95**

Enable Source Class and Destination Class Usage | **99**

Source Class and Destination Class Usage Overview | **99**

Enable Source Class and Destination Class Usage | **103**

Understanding Targeted Broadcast | **109**

Configure Targeted Broadcast | **110**

Configure Targeted Broadcast and Its Options | **111**

Display Targeted Broadcast Configuration Options | **112**

2 **Other Interfaces**

Discard Interfaces | 115

Discard Interface Overview | **115**

Discard Interface Configuration | **116**

Configure the Discard Interface | **116**

Configure an Output Policy | **117**

Loopback Interfaces | 118

Loopback Interface Overview | **118**

Loopback Interface Configuration | **119**

Configure the Loopback Interface | **120**

Example: Configure Two Addresses on the Loopback Interface with Host Routes | **121**

3 **Troubleshooting Interfaces**

Troubleshooting Interfaces | 124

Troubleshooting: Management Interface Link Is Down for Junos OS Evolved | **124**

Troubleshooting: Invalid Port Speed Configuration for Junos OS Evolved | **127**

Troubleshooting: Faulty Ethernet Physical Interface for Junos OS Evolved | **131**

Check the Cable Connection | **131**

Check the Physical Link Status of the Interface | **133**

Check the Interface Statistics in Detail | **134**

Perform the Loopback Diagnostic Test | **137**

Check for Other Possibilities | **140**

Enable a Physical Interface | **141**

About This Guide

Use this guide to configure, monitor and troubleshoot various interfaces installed on a Juniper Networks device with the Junos OS Evolved CLI.

CHAPTER 1

Device Interfaces

Device Interfaces Overview | 2

Physical Interface Properties | 12

Media MTU and Protocol MTU | 34

Interface Ranges for Physical Interfaces | 39

Damping Physical Interfaces | 50

Logical Interface Properties | 61

Protocol Family and Interface Address Properties | 74

Device Interfaces Overview

IN THIS SECTION

- Device Interfaces Overview | 2
- Types of Interfaces | 3
- Interface Naming Overview | 3
- Interface Descriptors Overview | 7
- Physical Part of an Interface Name | 9
- Displaying Interface Configurations | 9
- Interface Encapsulations Overview | 10
- Understanding Internal Ethernet Interfaces | 12

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

Device Interfaces Overview

Juniper devices typically contain several different types of interfaces suited to various functions. For the interfaces on a device to function, you must configure them. Specifically, you must configure 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. 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 device as well as interfaces that are not currently present but that are expected to be added in the future. Junos OS Evolved detects the interface after the hardware has been installed and applies the pre-set configuration to it.

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

You can configure Junos OS Evolved 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.

Types of Interfaces

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

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

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

  - *Understanding Management Ethernet Interfaces*
  - "Understanding Internal Ethernet Interfaces" on page 12

- Transient interfaces—Interfaces that can be inserted into or removed from the device depending on your network configuration needs.
- Networking interfaces—Interfaces that primarily provide traffic connectivity.
- Services interfaces—Interfaces that provide specific capabilities for manipulating traffic before it is delivered to its destination.

Interface Naming Overview

IN THIS SECTION

- Physical Part of an Interface Name | 4
- Logical Part of an Interface Name | 6
- Separators in an Interface Name | 6
- Chassis Interface Naming | 7

Each interface has an interface name, which specifies the media type, the slot in which the Flexible PIC Concentrator (FPC) or Dense Port Concentrator (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, such as 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 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 management interface is dependent on the Routing Engine. To identify if the Routing Engine is using this type of interface, use the following command:

show interfaces terse

```
user@host> show interfaces terse
Interface               Admin Link Proto    Local                 Remote
pfh-1/0/0               up    up
pfh-1/0/0.16383         up    up   inet
et-1/0/1                up    up
et-1/0/1.16386          up    up   multiservice
et-1/0/3                up    up
et-1/0/3.16386          up    up   multiservice
et-1/0/5                up    up
et-1/0/5.16386          up    up   multiservice
et-1/0/7                up    up
et-1/0/7.16386          up    up   multiservice
et-1/0/9                up    up
et-1/0/9.16386          up    up   multiservice
```

```
et-1/0/11               up    up
et-1/0/11.16386         up    up   multiservice
et-1/0/13               up    up
et-1/0/13.16386         up    up   multiservice
et-1/0/15               up    up
et-1/0/15.16386         up    up   multiservice
re0:mgmt-0              up    up <----------------
re0:mgmt-0.0            up    up   inet     10.53.95.205/19
re1:mgmt-0              up    up <----------------
re1:mgmt-0.0            up    up   inet     10.53.95.194/19
dsc                     up    up
esi                     up    up
fti0                    up    up
fti1                    up    up
fti2                    up    up
fti3                    up    up
fti4                    up    up
fti5                    up    up
fti6                    up    up
fti7                    up    up
irb                     up    up
lo0                     up    up
lo0.0                   up    up   inet     128.53.95.205       --> 0/0
                                   iso      47.0005.80ff.f800.0000.0108.0001
                                   inet6    abcd::128:53:95:205 -->
                                            fe80::5604:15f0:0:c200-->
lsi                     up    up
pip0                    up    up
vtep                    up    up
```

This part of the interface name has the following format:

```
type-fpc/pic/port[:channel]
```

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.
- dsc—Discard interface.

- et—Ethernet interfaces (10-, 25-, 40-, 50-, 100-, 200-, and 400-Gigabit Ethernet interface).
- gr—Generic routing encapsulation (GRE) tunnel interface.
- lo—Loopback interface. The Junos OS Evolved automatically configures one loopback interface (lo0). The logical interface lo0.16383 is a nonconfigurable interface for router control traffic.
- lsi—Internally generated interface that is not configurable.
- pip—Provider Instance Port (PIP) interface for EVPNs.
- vtep—Virtual tunnel endpoint interface for VXLANS.

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.

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. The slots in an FPC with four PIC slots are numbered 0 through 3. The slots in an FPC 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.

channel identifies the channel identifier part of the interface name and is required only on channelized interfaces. For channelized interfaces, channel 0 identifies the first channelized interface.

Logical Part of an Interface Name

The logical unit part of the interface name corresponds to the logical unit number. The range of available numbers varies for different interface types.

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

```
type-fpc/pic/port[:channel]
.logical-unit
```

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.

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 {
        ...
    }
}
```

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 is located
- The location on the FPC in which the PIC is installed
- The PIC port
- The interface's channel and logical unit numbers (optional)

Each physical interface descriptor can contain one or more *logical interface* descriptors. These descriptors enable you to map one or more logical (or virtual) interfaces to a single physical device. Creating multiple logical interfaces enables you to 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)
- Ethernet (ethernet switching)
- Circuit cross-connect (CCC)
- Translational cross-connect (TCC)
- International Organization for Standardization (ISO)
- Multiprotocol Label Switching (MPLS)

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 `et-0/0/0.1`, where the logical unit number is 1, as shown in the following examples:

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

Physical Part of an Interface Name

IN THIS SECTION

- Interface Names for ACX Series, PTX Series, and QFX Series Devices | 9

Interface Names for ACX Series, PTX Series, and QFX Series Devices

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: Some Juniper devices do not have actual PICs. 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 devices with PICs 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 (for example, **et**) from the FPC number. A slash (**/**) separates the FPC, PIC, and port numbers. A colon (**:**) separates the port number and channel (optional):

```
type-fpc/pic/port[:channel]
```

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, first from lowest to highest slot number, and then from lowest to highest PIC number, and finally from lowest to highest port number.

Interface Encapsulations Overview

Table 1 on page 10 lists encapsulation support by interface type.

Table 1: 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
dsc—Discard interface	NA	NA

Table 1: Encapsulation Support by Interface Type *(Continued)*

Interface Type	Physical Interface Encapsulation	*Logical Interface* Encapsulation
Ethernet interfaces (et)	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
lo—Loopback interface; the Junos OS Evolved automatically configures one loopback interface (lo0).	NA	NA
Services interface (gr)	NA	NA
Unconfigurable, internally generated interface (lsi)	NA	NA

Understanding Internal Ethernet Interfaces

Within a Juniper device, internal Ethernet interfaces provide communication between the Routing Engine and the Packet Forwarding Engines. Junos OS Evolved automatically configures internal Ethernet interfaces when Junos OS Evolved boots. Junos OS Evolved boots the packet-forwarding component hardware. When these components run, the Control Board (CB) uses the internal Ethernet interface to transmit hardware status information to the Routing Engine. Hardware status information includes the internal router temperature, the condition of the fans, whether an FPC has been removed or inserted, and information from the LCD on the craft interface.

NOTE: Do not modify or remove the configuration for the internal Ethernet interface that Junos OS Evolved automatically configures. If you do, the device stops functioning.

- Most Juniper devices—Junos OS Evolved creates the internal Ethernet interface. The internal Ethernet interface connects the Routing Engine `re0` to the Packet Forwarding Engines.

  If the device 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.

Each device also has one or two serial ports, labeled **CON** (*console*) or **AUX** (*auxiliary*), for connecting tty type terminals to the device using standard PC-type tty cables. Although these ports are not network interfaces, they do provide access to the device. Refer to your devices hardware guide for details.

Physical Interface Properties

IN THIS SECTION

- Physical Interface Properties Overview | **13**
- Configure the Interface Description | **13**
- How to Specify an Aggregated Interface | **14**
- Interface Speed | **15**
- Forward Error Correction (FEC) | **19**
- Interface Aliases | **21**

- Interface Encapsulation on Physical Interfaces | 22
- Enable SNMP Notifications on Physical Interfaces | 26
- Accounting for Physical Interfaces | 27
- Disable a Physical Interface | 30

Use this topic to configure various properties of physical interfaces on your device. Read on to configure properties such as interface descriptions, interface speeds, and accounting profiles for physical interfaces.

Physical Interface Properties Overview

The software driver for each network media type sets reasonable default values for general interface properties. These properties include the interface's maximum transmission unit (MTU) size, receive and transmit leaky bucket properties, and speed.

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

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

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

For example:

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

NOTE: You can configure the extended DHCP relay to include the interface description in the option 82 Agent Circuit ID suboption.

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

```
user@host> show interfaces et-1/0/1
Physical interface: et-1/0/1, 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`.

For information about describing logical units, see "Adding a Logical Unit Description to the Configuration" on page 63.

How to Specify an Aggregated Interface

An aggregated interface is a group of interfaces. To specify an aggregated Ethernet interface, configure ae*x* at the `[edit interfaces]` hierarchy level, where *x* is an integer starting at 0.

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

Interface Speed

IN THIS SECTION

- Configuring the Interface Speed on Ethernet Interfaces | 15
- Configure Interface Speeds for PTX10003 Routers and QFX5220 Switches | 16
- Configure the Aggregated Ethernet Link Speed | 17

The interface speed is the maximum amount of data that can travel through an interface per second. An interface speed ending in `m` is in megabits per second (Mbps). A link speed ending in `g` is in gigabits per second (Gbps).

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 RouterMX 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*

Configure Interface Speeds for PTX10003 Routers and QFX5220 Switches

For a PTX10003 routers and QFX5220 switches, configure the port speed at the `[edit chassis]` level rather than the `[edit interface]` level.

1. Navigate to the configuration hierarchy of the FPC slot number, PIC number, and port number you want to configure.

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

2. To configure a port to operate at a specific speed:

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

   For example, to configure the interface et-1/0/3 to operate as a 100-Gigabit Ethernet port:

```
[edit chassis fpc 1 pic 0 port 3]
user@host# set speed 100g
```

3. (Optional) You can channelize the interface into sub-ports. The interface speed of each sub-port is based on the speed you configure for the main port. The default number of sub-ports is 1 (a non-channelized port). The maximum number of sub-ports depends on your device, the line card, and the port number. To channelize the port, include the `number-of-sub-ports` statement:

```
[edit chassis fpc slot-number pic pic-number port port-number]
user@host# set number-of-sub-ports number
```

   For example:

```
[edit chassis fpc 1 pic 0 port 3]
user@host# set number-of-sub-ports 4
```

   Since we configured the interface et-1/0/3 to have a speed of 100 Gbps, each of the four sub-ports has a speed of 25 Gbps. The four sub-ports are et-1/0/3:0, et-1/0/3:1, et-1/0/3:2, and et-1/0/3:3.

Configure the Aggregated Ethernet Link Speed

On aggregated Ethernet interfaces, you can set the required link speed for all interfaces included in the bundle.

Some devices support mixed rates and mixed modes. For example, you could configure the following on the same aggregated Ethernet interface:

- Member links of different modes (WAN and LAN) for 10-Gigabit Ethernet links

- Member links of different rates: 10-Gigabit Ethernet, 25-Gigabit Ethernet, 40-Gigabit Ethernet, 50-Gigabit Ethernet, 100-Gigabit Ethernet, 400-Gigabit Ethernet, and OC192 (10-Gigabit Ethernet WAN mode)

NOTE:

- You can only configure 50-Gigabit Ethernet member links using the 50-Gigabit Ethernet interfaces of 100-Gigabit Ethernet PIC with CFP (PD-1CE-CFP-FPC4).
- You can only configure 100-Gigabit Ethernet member links using the two 50-Gigabit Ethernet interfaces of a 100-Gigabit Ethernet PIC with CFP. You can include this 100-Gigabit Ethernet member link in an aggregated Ethernet link that includes member links of other interfaces as well.

To configure the aggregated Ethernet link speed:

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

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

For example:

```
[edit]
user@host# edit interfaces ae0 aggregated-ether-options
```

2. Configure the link speed.

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

For example, to set the link speed of all member links of the aggregated Ethernet interface to 10 Gbps:

```
[edit interfaces ae0 aggregated-ether-options]
user@host# set link-speed 10g
```

3. (Optional) If you plan to configure the link speed of the member links to be different speeds, set the link speed for the aggregated Ethernet interface to `mixed`.

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

For example:

```
[edit interfaces ae0 aggregated-ether-options]
user@host# set link-speed mixed
```

NOTE: The QFX5000 line of switches does not support mixed link speed for aggregated Ethernet interfaces.

Forward Error Correction (FEC)

SUMMARY

Forward error correction (FEC) improves the reliability of the data transmitted by your device. When FEC is enabled on an interface, that interface sends redundant data. The receiver accepts data only where the redundant bits match, which removes erroneous data from the transmission. Junos OS Evolved enables you (the network administrator) to configure Reed-Solomon FEC (RS-FEC) and BASE-R FEC on Ethernet interfaces. RS-FEC is compliant with IEEE 802.3-2015 Clause 91. BASE-R FEC is compliant with IEEE 802.3-2015 Cause 74.

IN THIS SECTION

- Benefits of FEC | 19
- Overview | 20
- Configure FEC | 20

Benefits of FEC

When you configure FEC on Ethernet interfaces, FEC improves your device function in these ways:

- Enhances the reliability of the connection
- Enables the receiver to correct transmission errors without requiring retransmission of the data

- Extends the reach of optics

Overview

By default, Junos OS Evolved enables or disables FEC based on the plugged-in optics. For instance, Junos OS Evolved enables RS-FEC for 25 Gigabit (Gb) or 50 Gb SR4 optics and disables RS-FEC for 25 Gb or 50 Gb LR4 optics. You can override the default behavior and explicitly enable or disable FEC. You must disable FEC mode if you do not want it assigned by default.

You can enable or disable RS-FEC for 25-, 50-, and 100-Gigabit Ethernet (GbE) interfaces. You can enable or disable BASE-R FEC for 25GbE and 50GbE interfaces. If you enable or disable FEC, this behavior applies to any 25GbE or 50GbE optical transceiver installed in the port associated with the interface. You can configure FEC clauses CL74 on 25 Gb and 50 Gb interfaces and CL91 on 25 Gb, 50 Gb, or 100 Gb interfaces. Because the FEC clauses are applied by default on these interfaces, you must disable the FEC clauses if you do not want to apply them.

NOTE: FEC is always enabled on 200GbE and 400GbE interfaces. You cannot disable it.

If there is an FEC mismatch between pairs of nodes, the link between nodes can go down. To prevent the nodes from going down, you must reconfigure them.

For instance, consider two peer nodes, Node1 and Node2. Node1 is running Junos OS Evolved Release 21.1R1, where the default is FEC91. The peer node (Node2) is running Junos OS Evolved Release 20.1R1, where the default is FEC74. The link between the two nodes will go down because the FEC modes don't match. To enable the link to come back up, you must manually change the FEC on one of the nodes.

Configure FEC

To disable or enable an FEC mode on an interface and any associated interfaces, complete the relevant action:

1. To disable FEC mode:

```
[edit]
user@device# set interfaces interface-name ether-options fec none
```

2. To enable an FEC mode:

```
[edit]
user@device# set interfaces interface-name ether-options fec (fec74 | fec91)
```

Alternatively:

```
[edit]
user@device# delete interfaces interface-name ether-options fec none
```

3. To view the FEC mode on an interface, use the `show interfaces` *interface-name* command. The output lists FEC statistics for that particular interface, including the number of FEC corrected errors, the number of FEC uncorrected errors, and the type of FEC that was disabled or enabled.

Interface Aliases

IN THIS SECTION

- Overview | 21
- Configuration | 21

Overview

An interface alias is a textual description of a logical unit on a physical interface. An alias enables you to give a single meaningful and easily identifiable name to an interface. Interface aliasing is supported only at the unit level.

The alias name is displayed instead of the interface name in the output of all `show`, `show interfaces`, and other operational mode commands. Configuring an alias for a logical unit of an interface has no effect on how the interface operates on the device.

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. When the operating system processes query the configuration database for the *interface-name* variable, the exact value of the *interface-name* variable is returned instead of the alias name for system operations and computations.

Configuration

To specify an interface alias, use the `alias` statement at the `[edit interfaces` *interface-name* `unit` *logical-unit-number*`]` hierarchy level. Start the alias name with a letter followed by letters, numbers, dashes, dots,

underscores, colons, or slashes. Avoid starting the alias with any part of a valid interface name. Use between 5 and 128 characters.

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

For example:

```
[edit interfaces et-1/0/1 unit 0]
user@device# set alias controller-sat1-downlink1
```

NOTE: If you configure the same alias name on more than one logical interface, the router displays an error message, and the commit fails.

You can use interface alias names to make it easy to see the roles interfaces play in your configuration. For example, to make it easy to identify satellite connection interfaces:

1. Group physical interfaces as one aggregated interface using a link aggregation group (LAG) or LAG bundle. Name that aggregated interface sat1 to show it is a satellite connection interface.
2. Select a logical interface as a member of the LAG bundle or the entire LAG. Name that interface et-0/0/1 to represent a satellite device port or a service instance.
3. You can combine the satellite name and the interface name aliases to wholly represent the satellite port name. For example, you could give your satellite port the alias sat1:et-0/0/1.

Interface Encapsulation on Physical Interfaces

IN THIS SECTION

- Encapsulation Capabilities | 23
- Configure Encapsulation on a Physical Interface | 24
- Configure Interface Encapsulation on PTX Series Routers | 24

Point-to-Point Protocol (PPP) encapsulation is the default encapsulation type for physical interfaces. You don't need to configure encapsulation for physical interfaces that support PPP encapsulation, because 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. On a logical interface, you can optionally configure an encapsulation type that Junos OS Evolved uses within certain packet types.

Encapsulation Capabilities

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 circuit cross-connect (CCC) encapsulation for Ethernet interfaces with standard Tag Protocol Identifier (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 Ethernet interfaces, 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.

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

When you configure a translational cross-connect (TCC) encapsulation, some modifications are needed to handle VPN connections over dissimilar Layer 2 and Layer 2.5 links and terminate the Layer 2 and Layer 2.5 protocol locally. The device performs the following media-specific changes:

- Point-to-Point Protocol (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. Junos OS Evolved strips all PPP encapsulation data from incoming frames before forwarding them. For output, the next hop is changed to PPP encapsulation.
- Cisco High-Level Data Link Control (HDLC) TCC—Keepalive processing is terminated on the router. Junos OS Evolved 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. Junos OS Evolved strips all Frame Relay encapsulation data from incoming frames before forwarding them. For output, the next hop is changed to Frame Relay encapsulation.

Configure Encapsulation on a Physical Interface

To configure encapsulation on a physical interface:

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

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

2. Configure the encapsulation type.

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

Configure Interface Encapsulation on PTX Series 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` or `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-1/0/3 {
        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;
                }
            }
        }
    }
}
```

Enable SNMP Notifications on Physical Interfaces

By default, Junos OS Evolved sends Simple Network Management Protocol (SNMP) notifications when the state of an interface or a connection changes. You can enable or disable SNMP notifications based on your requirements.

To explicitly enable sending SNMP notifications on the physical interface:

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 SNMP notifications when the state of the connection changes.

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

To disable SNMP notifications on the physical interface:

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 SNMP notifications when the state of the connection changes.

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

Accounting for Physical Interfaces

IN THIS SECTION

- Overview | 27
- Configure an Accounting Profile for a Physical Interface | 27
- How to Display the Accounting Profile | 29

Devices running Junos OS Evolved can collect various kinds of data about traffic passing through the device. You (the systems administrator) can set up one or more *accounting profiles* that specify some common characteristics of this data. These characteristics include 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

Overview

There are two types of accounting profiles: filter profiles and interface profiles. Configure the profiles using statements at the `[edit accounting-options]` hierarchy level.

Configure filter profiles by including the `filter-profile` statement at the `[edit accounting-options]` hierarchy level. 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.

Configure interface profiles by including the `interface-profile` statement at the `[edit accounting-options]` hierarchy level. Read on to learn how to configure interface profiles.

Configure an Accounting Profile for a Physical Interface

Before You Begin

Configure an accounting data log file at the `[edit accounting-options]` hierarchy level. The operating system logs the statistics in the accounting data log file.

Configuration

Configure an interface profile to collect error and statistic information for input and output packets on a particular physical interface. The interface profile specifies the information that the operating system writes to the log file.

To configure an interface profile:

1. Navigate to the `[edit accounting-options interface-profile]` hierarchy level. Include the *profile-name* to name the interface profile.

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

2. To configure which statistics should be collected for an interface, include the `fields` statement.

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

3. Each accounting profile logs its statistics to a file in the **/var/log** directory. To configure which file to use, use the `file` statement.

```
[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.

4. The operating system collects statistics from each interface with an accounting profile enabled. It collects the statistics once per interval time specified for the accounting profile. The operating system schedules statistics collection time evenly over the configured interval. To configure the interval, use the `interval` statement:

```
[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.

5. Apply the interface profile to a physical interface by including the `accounting-profile` statement at the `[edit interfaces interface-name]` hierarchy level. The operating system performs the accounting on the interfaces that you specify.

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

How to Display the Accounting Profile

IN THIS SECTION

- Purpose | 29
- Action | 29
- Meaning | 30

Purpose

To display the configured accounting profile of a particular physical interface at the `[edit accounting-options interface-profile profile-name]` hierarchy level that has been configured with the following:

- interface-name—et-1/0/1
- Interface profile —`if_profile`
- File name—`if_stats`
- Interval—15 minutes

Action

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

```
[edit interfaces et-1/0/1]
user@host# show
accounting-profile if_profile;
```

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

```
[edit accounting-options]
user@host# show
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.

Disable a Physical Interface

IN THIS SECTION

- How to Disable a Physical Interface | **31**
- Example: Disable a Physical Interface | **32**

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

How to Disable a Physical Interface

CAUTION: Dynamic subscribers and logical interfaces use physical interfaces for connection to the network. You can 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 the `[edit interfaces interface-name]` hierarchy level.

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

2. Include the `disable` statement.

```
[edit interfaces interface-name]
user@device# set disable
```

For example:

```
[edit interfaces et-1/0/7]
user@device# set disable
```

NOTE: 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. On a device with newer PICs, the laser turns off when the interface is disabled.

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

3. Alternatively, include the `unused` statement. If you configure a port as unused, no interfaces are created for that port irrespective of the port profile configuration for that port.

a. For PTX10003 routers and QFX5220 switches:

```
[edit chassis fpc slot-number pic pic-number port port-number]
user@device# set unused
```

For example:

```
[edit chassis fpc 1 pic 0 port 7]
user@device# set unused
```

b. For all other devices running Junos OS Evolved:

```
[edit interfaces interface-name]
user@device# set unused
```

For example:

```
[edit interfaces et-1/0/7]
user@device# set unused
```

Example: Disable a Physical Interface

Sample interface configuration:

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

Disable the interface:

```
[edit interfaces et-0/3/2]
user@device# set disable
```

Verify the interface configuration:

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

Release History Table

Release	Description
21.1R1	Starting in Junos OS Evolved Release 21.1R1, the default FEC is FEC91. In earlier releases, the default is FEC74.
14.2	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 10mmaximum 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.
11.4	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.

Media MTU and Protocol MTU

SUMMARY

A maximum transmission unit (MTU) is the largest data unit that can be forwarded without fragmentation. Configure the media MTU for a physical interface and the MTU for a protocol to optimize traffic over your network.

IN THIS SECTION

- Media MTU Overview | 34
- Configure the Media MTU | 35
- Protocol MTU | 36
- Encapsulation Overhead by Interface Encapsulation Type | 37

Media MTU Overview

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

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

The default media MTU for a physical interface is calculated as follows:

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

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 when you calculate MTUs for interoperability.

Keep the following in mind when configuring the media MTU:

- The MTU size must be the same on both sides of a point-to-point connection.
- All interfaces in the subnet of point-to-multipoint connections must use the same MTU size.
- The physical MTU for Ethernet interfaces does not include the 4-byte frame check sequence (FCS) field of the Ethernet frame.

- The maximum number of data-link connection identifiers (DLCIs) is determined by the MTU on the interface. If you have keepalives enabled with the MTU set to 5012, the maximum number of DLCIs is 1000.

Because tunnel services interfaces are considered logical interfaces, you cannot configure the MTU setting for the associated physical interface. This means that you cannot configure the MTU size for the following interface types:

- Loopback (lo-)

Configure the Media MTU

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. In other words:

```
Minimum media MTU = protocol MTU + encapsulation overhead
```

The maximum media MTU size that you can configure depends on your device and the type of interface.

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

To configure the media MTU:

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]
user@host# set mtu bytes
```

Protocol MTU

IN THIS SECTION

- Configure the Protocol MTU | 36

Overview

The default protocol MTU depends on your device and the interface type. When you initially configure an interface, the protocol MTU is calculated automatically. If you subsequently change the media MTU, the protocol MTU on existing address families automatically changes.

If you reduce the media MTU size but one or more address families are already configured and active on the interface, you must also reduce the protocol MTU size. 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.

If you do not configure an MPLS MTU, Junos OS Evolved 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. 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 as follows:

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

You can configure the protocol MTU on all tunnel interfaces.

Configure the Protocol MTU

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

To configure the protocol MTU:

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. Include the mtu statement for each family you want to configure with a non-default MTU value.

   If you configure the protocol MTU for any family, the configured value is applied to all families that are configured on the logical interface.

```
[edit interfaces interface-name unit logical-unit-number]
user@host# set family family mtu bytes
```

> **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 families. We do not recommend configuring 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 2: 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

Table 2: Encapsulation Overhead by Encapsulation Type *(Continued)*

Interface Encapsulation	Encapsulation Overhead (Bytes)
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
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

Interface Ranges for Physical Interfaces

IN THIS SECTION

- Configure Interface Ranges | 39
- Expanded Interface Range Statements | 42
- Configuration Inheritance Priority | 44
- Configuration Inheritance for Member Interfaces | 45
- Common Configuration Inheritance | 46
- Configuration Group Inheritance | 47
- Configuration Expansion Where Interface Range Is Used | 48

Junos OS Evolved enables 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. Interface ranges reduce the number of configuration statements required. They save time and produce a compact configuration.

Configure Interface Ranges

To configure an interface range, use the `interface-range` statement at the `[edit interfaces]` hierarchy level. The `interface-range` statement accepts only physical networking interface names in its definition. Junos OS Evolved supports interface ranges for Ethernet interfaces: `et-`*`fpc/pic/port`*.

To configure an interface range:

1. Use the `interface-range` statement at the `[edit interfaces]` hierarchy level. Include the name you have chosen for your interface range.

```
[edit]
user@device# edit interfaces interface-range range-name
```

For example, to configure an interface range named "range1":

```
[edit]
user@device# edit interfaces interface-range range1
```

2. To specify a member range, use the `member-range` *`start-range`* `to` *`end-range`* statement at the `[edit interfaces interface-range` *`range-name`*`]` hierarchy level. For example:

```
[edit interfaces interface-range range1]
user@device# set member-range et-1/0/0 to et-4/0/40
```

3. To specify an individual member, use the `member` statement at the `[edit interfaces interface-range` *`range-name`*`]` hierarchy level. For example:

```
[edit interfaces interface-range range1]
user@device# set member et-0/0/0
```

4. You can specify a list of interface range members using regular expressions with the `member` *`range of interface names`* statement. A range for a member statement can 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 from low to high. Specifies a range of sequential interfaces.
   - `[num1, num2, num3]`—Numbers `num1`, `num2`, and `num3` specify multiple specific interfaces.

   Regular expressions and wildcards are not supported for interface-type prefixes. For example, prefixes `et` and `xe` must be mentioned explicitly.

For example:

```
[edit interfaces interface-range range1]
user@device# set member et-0/*/*
set member et-0/[1-10]/0
set member et-0/[1,2,3]/3
```

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

An `interface-range` definition having just `member` or `member-range` statements and no common configuration statement is valid. However, you can optionally add a common configuration statement to an interface range as a part of the `interface-range` definition. For example:

```
[edit]
interfaces {
    +    interface-range range1 {
    +        member-range et-1/0/0 to et-4/0/40;
    +        member et-0/0/0;
    +        member et-0/*/*;
    +        member et-0/[1-10]/0;
    +        member et-0/[1,2,3]/3;

        /*Common configuration is added as part of interface-range definition*/
        mtu 500;
        ether-options {
            flow-control;
            speed {
                100m;
            }
            802.3ad primary;
        }

    }
}
```

These defined interface ranges can be used in other configuration hierarchies in places where an `interface` node exists. For example:

```
protocols {
    dot1x {
        authenticator {
            interface range1 {
                retries 1;
            }
        }
    }
}
```

In the preceding 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.

Expanded Interface Range Statements

The operating system expands all `member` and `member-range` statements in an interface range definition to generate the final list of interface names for the specified interface range.

An example configuration looks like this before it is expanded:

```
[edit]
interfaces {
    interface-range range1 {
        member-range et-0/0/0 to et-4/0/20;
        member et-10/1/1;
        member et-5/[0-5]/*;

        /*Common configuration is added as 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 et-0/0/0 to et-4/0/20
```

expands to:

```
      [et-0/0/0, et-0/0/1 ... et-0/0/max_ports
       et-0/1/0  et-0/1/1 ... et-0/1/max_ports
       et-0/2/0  et-0/2/1 ... et-0/2/max_ports
                           .
                           .
       et-0/MAX_PICS/0 ... et-0/max_pics/max_ports
       et-1/0/0  et-1/0/1 ... et-1/0/max_ports
                           .
       et-1/MAX_PICS/0 ... et-1/max_pics/max_ports
                           .
                           .
       et-4/0/0 et-4/0/1  ... et-4/0/max_ports]
```

The following `member` statement:

```
et-5/[0-5]/*
```

expands to:

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

The following `member` statement:

```
et-5/1/[2,3,6,10]
```

expands to:

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

Configuration Inheritance Priority

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

In this example, interface `et-1/1/1` exists in both interface range `int-grp-one` and interface range `int-grp-two`:

```
[edit]
interfaces {
    interface-range int-grp-one {
        member-range et-0/0/0 to et-4/0/47;
        member et-1/1/1;

        /*Common config is added part of the interface-range definition*/
        mtu 500;
        hold-time up 10;
    }
    interface-range int-grp-two {
        member-range et-5/0/0 to et-7/0/47;
        member et-1/1/1;

        mtu 1024;
    }
}
```

Interface `et-1/1/1` inherits `mtu` *`500`* from interface range `int-grp-one` because it was defined first.

Configuration Inheritance for Member Interfaces

When Junos OS Evolved 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 operating system copies the common configuration to all the interface range's member interfaces.

Foreground interface configuration takes priority over configuration that the interface inherits from the interface range configuration.

In this example, interface `et-1/0/1` has an MTU value of 1024 because that is its foreground configuration:

```
interfaces {
    interface-range range1 {
        member-range et-1/0/0 to et-7/0/47;
        mtu 500;
        }

    et-1/0/1 {
        mtu 1024;
    }
}
```

You can verify this in the output of the `show interfaces | display inheritance` command:

```
user@host: show interfaces | display inheritance
##
## 'et-1/0/0' was expanded from interface-range 'range1'
##
et-1/0/0 {
    ##
    ## '500' was expanded from interface-range 'range1'
    ##
    mtu 500;
}
et-1/0/1 {
    mtu 1024;
}
##
## 'et-1/0/2' was expanded from interface-range 'range1'
##
```

```
et-1/0/2 {
    ##
    ## '500' was expanded from interface-range 'range1'
    ##
    mtu 500;
}
      .........
      .........
##
## 'et-10/0/47' was expanded from interface-range 'range1'
##
et-10/0/47 {
    ##
    ## '500' was expanded from interface-range 'range1'
    ##
    mtu 500;
}
```

Common Configuration Inheritance

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

For example:

```
[edit]
interfaces {
    interface-range int-grp-one {
        member-range et-0/0/0 to et-4/0/40;

        mtu 256;
    }
    interface-range int-grp-two {
        member-range et-4/0/0 to et-4/0/40;

        hold-time up 10;
    }
}
```

In this example, interfaces `et-4/0/0` through `et-4/0/40` have both `hold-time` and `mtu` configured.

Configuration Group Inheritance

Interface range member interfaces inherit configurations from configuration groups like any other foreground configuration. The only difference is that the `interface-range` goes through a member interfaces expansion before the operating system reads this configuration.

In this example, Junos OS Evolved applies the `hold-time` configuration to all members of the interface range `range1`:

```
groups {
     global {
         interfaces {
             <*> {
                  hold-time up 10;
             }
         }
     }
}
apply-groups [global];
interfaces {
     interface-range range1 {
         member-range et-1/0/0 to et-7/0/47;
         mtu 500;
     }
}
```

You can verify this with `show interfaces | display inheritance`, as follows:

```
user@host# show interfaces | display inheritance
[...]
##
## 'et-1/0/0' was expanded from interface-range 'range1'
##
et-1/0/0 {
    ##
    ## '500' was expanded from interface-range 'range1'
    ##
    mtu 500;
    ##
    ## 'hold-time' was inherited from group 'global'
    ## '10' was inherited from group 'global'
```

```
    ##
    hold-time up 10;
}
##
## 'et-1/0/1' was expanded from interface-range 'range1'
##
et-1/0/1 {
    ##
    ## '500' was expanded from interface-range 'range1'
    ##
    mtu 500;
    ##
    ## 'hold-time' was inherited from group 'global'
    ## '10' was inherited from group 'global'
    ##
    hold-time up 10;
}
##
## 'et-7/0/47' was expanded from interface-range 'range1'
##
et-7/0/47 {
    ##
    ## '500' was expanded from interface-range 'range1'
    ##
    mtu 500;
    ##
    ## 'hold-time' was inherited from group 'global'
    ## '10' was inherited from group 'global'
    ##
    hold-time up 10;
}
```

Configuration Expansion Where Interface Range Is Used

In this example, `interface-range` *range1* is used under the `protocols` hierarchy:

```
[edit]
interfaces {
    interface-range range1 {
```

```
        member et-7/1/1;
        member et-5/0/1;

        mtu 500;
        hold-time up 10;
        ether-options {
            flow-control;
            speed {
                100m;
            }
            802.3ad primary;
        }
    }
    protocols {
        dot1x {
            authenticator {
                interface range1 {
                    retries 1;
                }
            }
        }
    }
}
```

The `interface` node present under `authenticator` expands into member interfaces of the interface range `range1` as follows:

```
protocols {
    dot1x {
        authenticator {
            interface et-7/1/1 {
                retries 1;
            }
            interface et-5/0/1 {
                retries 1;
            }
        }
    }
}
```

The `interface` *`range-1`* statement is expanded into two interfaces, et-7/1/1 and et-5/0/1, and the operating system copies the configuration `retries` *`1`* under those two interfaces.

You can verify this configuration using the `show protocols dot1x | display inheritance` command.

Damping Physical Interfaces

SUMMARY

You (the network administrator) can configure damping to reduce the advertisement of physical interface transitions between up and down states.

IN THIS SECTION

- Physical Interface Damping Overview | 50
- Configure Damping of Shorter Physical Interface Transitions | 59
- Configure Damping of Longer Physical Interface Transitions | 60

Physical Interface Damping Overview

IN THIS SECTION

- Damping Overview for Shorter Physical Interface Transitions | 51
- Damping Overview for Longer Physical Interface Transitions | 52

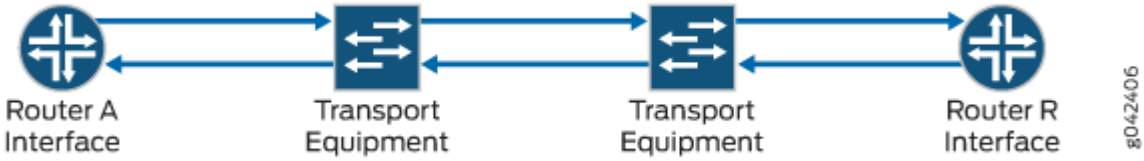
Physical interface damping limits the advertisement of the up-and-down transitions (flapping) on an interface. Each time a transition occurs, the interface state is changed, which generates an advertisement to the upper-level routing protocols. Damping helps reduce the number of these advertisements.

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 1 on page 51 is used to describe these types of interface flaps and the damping configuration that you can use in each case.

Figure 1: Two Router Interfaces Connected Through Transport Equipment



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

The types of interface damping depend upon the transition time length.

Damping Overview for Shorter Physical Interface Transitions

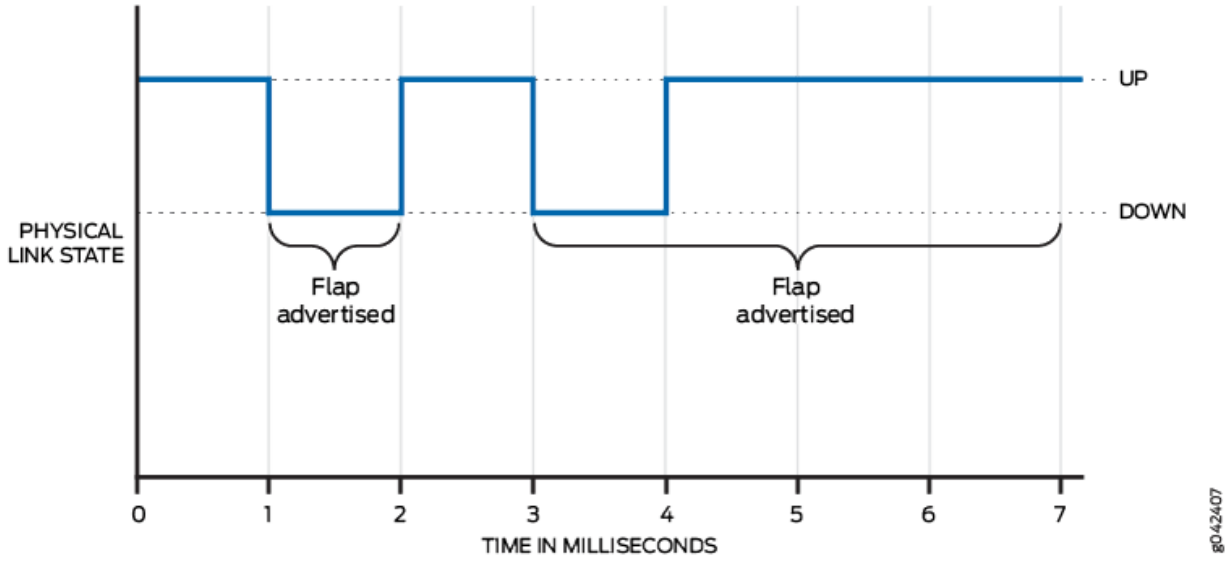
Figure 1 on page 51 shows two routers with two transport devices between them. If a redundant link between the two transport devices fails, Junos OS Evolved performs link switching. Link switching takes a number of milliseconds. As shown in Figure 2 on page 52, during switching, both device interfaces 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 2: Multiple Flaps of Short Duration (Milliseconds)

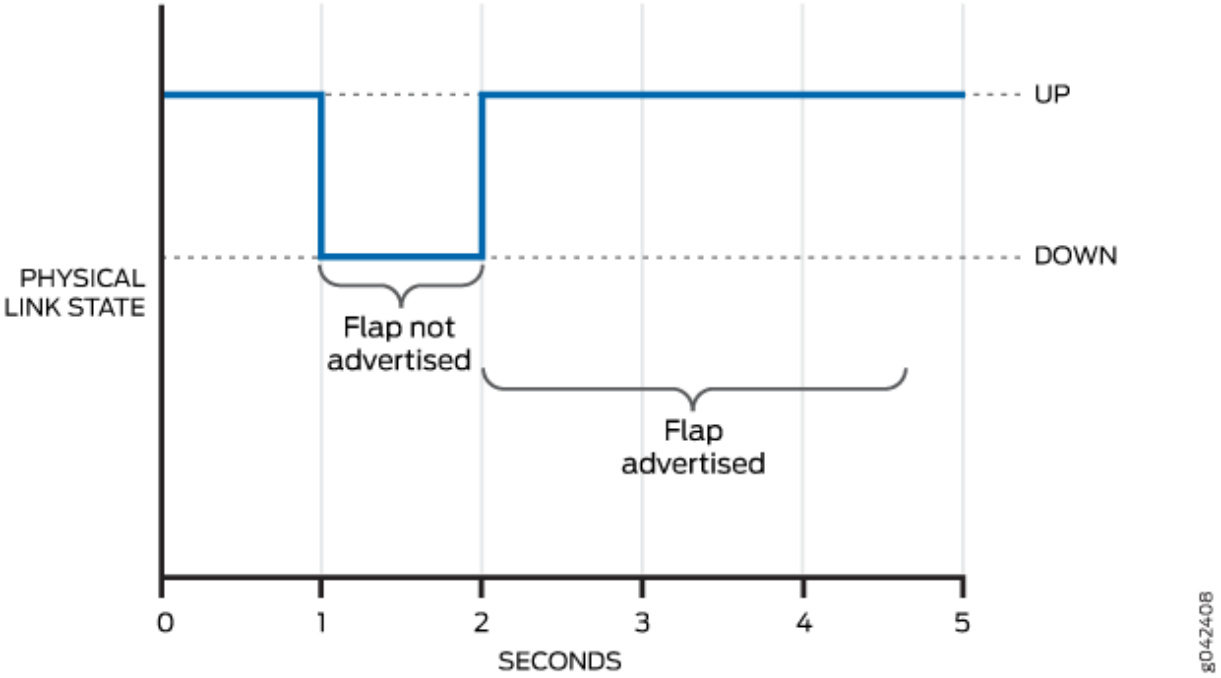


Damping Overview for Longer Physical Interface Transitions

When the link between a router interface and the transport devices is not stable, this can lead to periodic flapping, as shown in Figure 3 on page 53. 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 because 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. Increasing the duration therefore fails to suppress periodically flapping interfaces on the system.

Figure 3: Periodic Flaps of Long Duration (Seconds)



For longer periodic interface flaps, 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, Junos OS Evolved adds a penalty to the interface penalty counter. If at some point the accumulated penalty exceeds the suppress level, Junos OS Evolved places the interface in the suppress state. In this case, Junos OS Evolved does not report further interface link up-and-down events to the upper-level protocols.

The penalty added on every interface flap is 1000. At all times, the interface penalty counter follows an exponential decay process. Figure 4 on page 56 and Figure 5 on page 58 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 4 on page 56 and Figure 5 on page 58 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 4 on page 56 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 4: Physical-Level Link Is Down When the Penalty Falls Below the Reuse Level

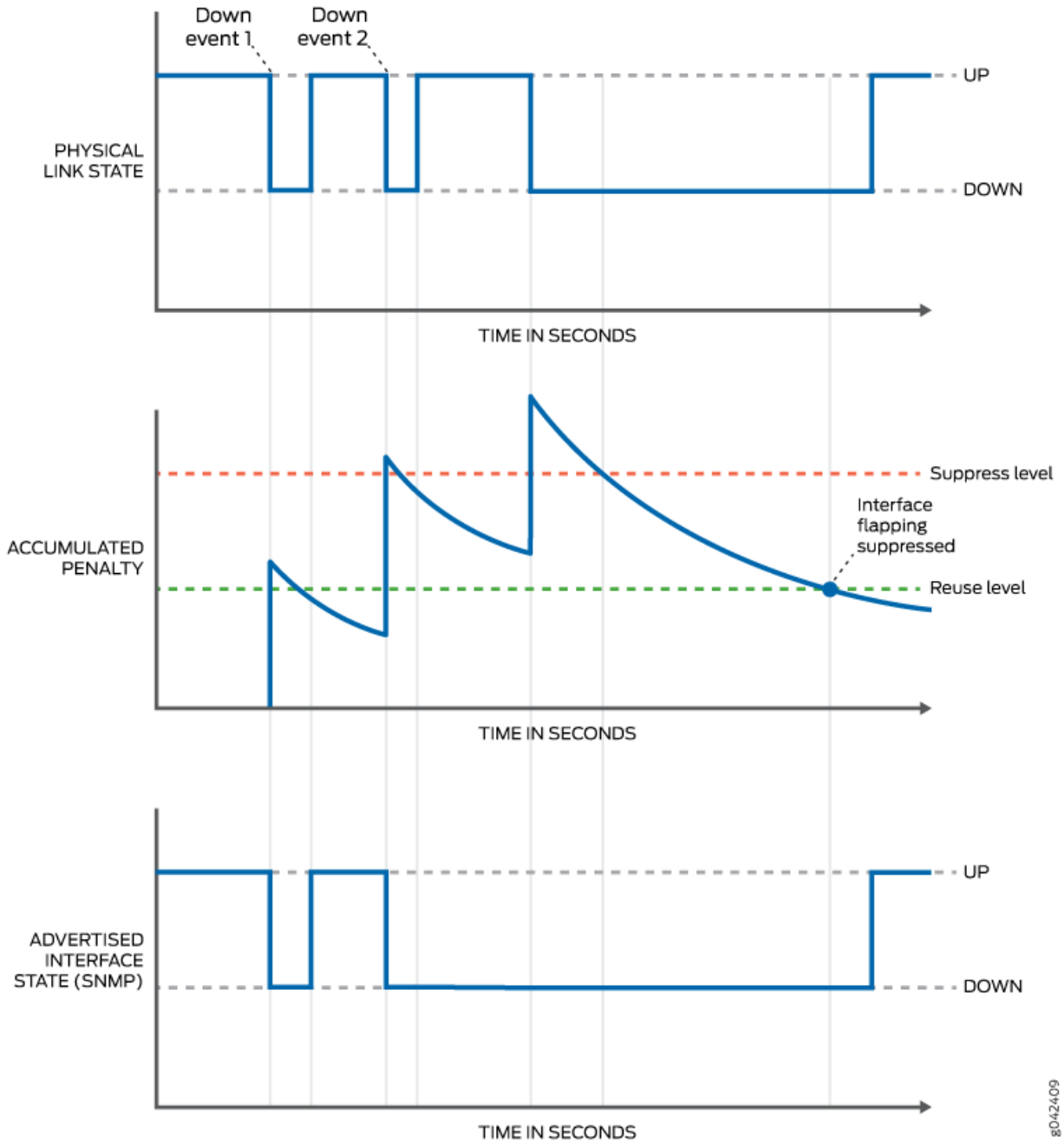
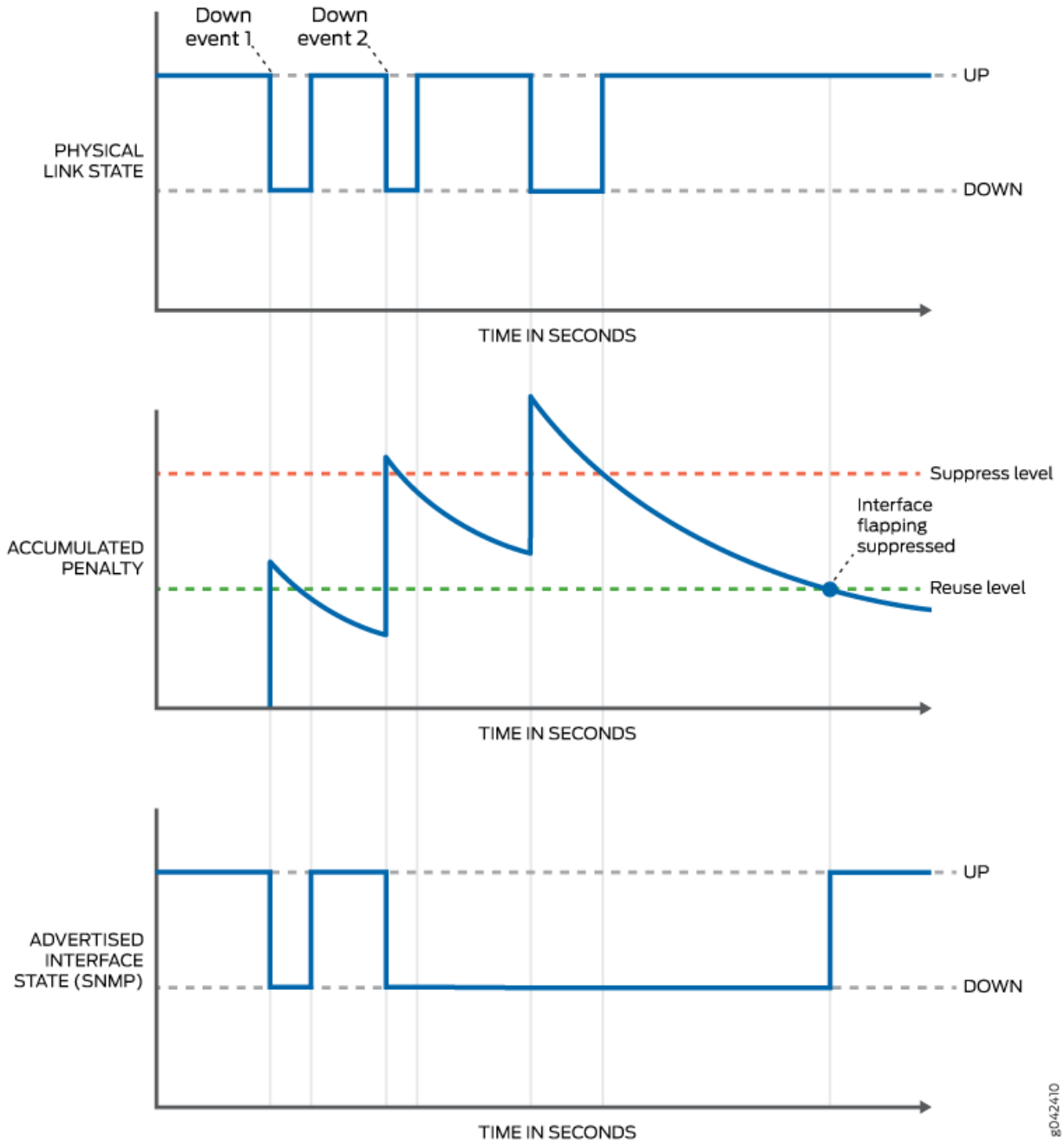


Figure 5 on page 58 shows the penalty dropping below the reuse level when the physical link is up. The system is notified of a state change immediately.

Figure 5: Physical-Level Link Is Up When the Penalty Falls Below the Reuse Level



Configure Damping of Shorter Physical Interface Transitions

By default, when an interface changes from up to down or from down to up, this transition is advertised immediately to the hardware and Junos OS Evolved. In some situations, you might want to damp interface transitions.

Damping the interface means not advertising the interface's transition until a certain period of time has passed, called the *hold-time*. When 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. If 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. If the timer expires and the interface state is still *up*, then the router begins to advertise the interface as being up.

To configure damping of shorter physical interface transitions in milliseconds:

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 Evolved advertises the transition within 100 milliseconds of the time value you specify.

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

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

Configure Damping of 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, 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.

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 damping
```

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 `half-life` to a value that is less than the value of `max-suppress`; 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
```

NOTE: The system does not indicate whether an interface is down because of suppression or because that is the actual state of the physical interface. Therefore, neither SNMP link traps nor Operation, Administration, and Maintenance (OAM) protocols can differentiate the damped version of the link state from the real version. Therefore, 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.

Logical Interface Properties

IN THIS SECTION

Logical Interface Properties Overview | 62

- Specify the Logical Interface Number | 62
- Add a Logical Unit Description to the Configuration | 63
- Configure the Interface Bandwidth | 63
- Configure Interface Encapsulation on Logical Interfaces | 64
- Configure Interface Encapsulation on PTX Series Routers | 67
- Overview of Accounting for the Logical Interface | 68
- Enable or Disable SNMP Notifications on Logical Interfaces | 72
- Disable a Logical Interface | 72

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. Properties 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 (DLCI), virtual channel identifiers (VCI) and virtual path identifiers (VPI), and traffic shaping.

To configure logical interface properties, include the statement at the following hierarchy level:

```
[edit interfaces interface-name]
```

Specify 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.

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 level:

`[edit interfaces interface-name]`

The range of number available for the logical unit number varies for different interface types. See *Ethernet Interfaces User Guide for Routing Devices* for current range values.

Add a Logical Unit Description to the Configuration

You can include a text description of each logical unit in the configuration file. Any descriptive text that you include displays in the output of the `show interfaces` commands. It 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 `[edit interfaces interface-name unit logical-unit-number]` hierarchy level.

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

For information about describing physical interfaces, see Configure the Interface Description.

Configure the Interface Bandwidth

By default, the operating system uses the physical interface 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 calculation for a dynamic routing protocol, such as OSPF. By default, the interface cost for a dynamic routing protocol is the following formula:

```
cost = reference-bandwidth/bandwidth,
```

In the formula, 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 level:

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

rate is the peak rate, in bits per second (bps) or cells per second (cps). You can specify a value in bps 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 cps by entering a decimal number followed by the abbreviation `c`. Values expressed in cps are converted to bps 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.

Configure Interface Encapsulation on Logical Interfaces

IN THIS SECTION

- Understand the Interface Encapsulation on Logical Interfaces | 65
- Configure the Encapsulation on a Logical Interface | 65
- Display the Encapsulation on a Logical Interface | 66

Understand the Interface Encapsulation on Logical Interfaces

An encapsulation is used with certain packet types. You can configure an encapsulation on a logical interface.

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 circuit cross-connect (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 data-link connection identifier (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 also 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
- 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 the only encapsulation type allowed on the logical interfaces.

Configure 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, or Ethernet 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*] hierarchy level.

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

2. Configure the encapsulation type.

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

Display the Encapsulation on a Logical Interface

IN THIS SECTION

- Purpose | 66
- Action | 66
- Meaning | 67

Purpose

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

- interface-name—et-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 et-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.

Configure Interface Encapsulation on PTX Series 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` or `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-1/0/3 {
        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;
                }
            }
        }
    }
}
```

Overview of Accounting for the Logical Interface

IN THIS SECTION

- Accounting Profiles Overview | **69**
- Configure Accounting for the Logical Interface | **69**
- Introduction to Displaying the Accounting Profile for the Logical Interface | **70**

This section discusses on how to configure accounting on logical interfaces.

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.

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.

Configure 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 which statistics are collected and written to a log file.

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 are 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 fields field-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.

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 logical 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-number  accounting-profile profile-name
```

Introduction to Displaying the Accounting Profile for the Logical Interface

IN THIS SECTION

- Purpose | 71
- Action | 71

Meaning | 72

Purpose

Displaying the configured accounting profile of a particular logical interface at the `[edit accounting-options interface-profile profile-name]` hierarchy level requires that you specify certain parameters:

- interface-name—et-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 et-1/0/1 unit 1]` hierarchy level.

```
[edit interfaces et-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.

Enable or Disable 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:

```
(traps);
```

To explicitly disable these notifications on the logical interface, include the `no-traps` statement:

```
(no-traps);
```

You can include these statements at the following hierarchy level:

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

Disable a Logical Interface

You can unconfigure a logical interface, effectively disabling that interface, without removing the logical interface configuration statements from the configuration. To unconfigure a logical interface, include the `disable` statement:

```
disable;
```

You can include this statement at the following hierarchy level:

`[edit 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: Disable 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

- Configure the Protocol Family | 74
- Assign the Interface Address | 75
- Configure Default, Primary, and Preferred Addresses and Interfaces | 77
- Operational Behavior of Interfaces with the Same IPv4 Address | 79
- Configure Unnumbered Interfaces: Overview | 83
- Protocol MTU | 91
- Disable the Removal of Address and Control Bytes | 93
- Disable the Transmission of Redirect Messages on an Interface | 93
- Apply a Filter to an Interface | 94
- Enable Source Class and Destination Class Usage | 99
- Understanding Targeted Broadcast | 109
- Configure Targeted Broadcast | 110

This section discusses on how to configure protocol family and interface address properties.

Configure 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.

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.

To configure the protocol family for the logical interface, include the `family` statement, specifying the selected family.

When configuring the protocol family, complete the following tasks under the `[edit interfaces interface-name unit logical-unit-number family family]` hierarchy.

- Configure MTU.
- Configure the unit and family so that the interface can transmit and receive multicast traffic only.
- Disable the sending of redirect messages by the router.
- Assign an address to an interface.

Assign 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.

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 IP version 4 (IPv4) address on routers and switches, 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
     ```

- To configure an IP version 6 (IPv6) address on routers and switches, 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 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] For interfaces that carry IPv6 traffic, configure the host to assign itself 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
```

Configure Default, Primary, and Preferred Addresses and Interfaces

IN THIS SECTION

- Default, Primary, and Preferred Addresses and Interfaces | 77
- Configure the Primary Interface for the Router | 77
- Configure the Primary Address for an Interface | 78
- Configure the Preferred Address for an Interface | 78

The following sections describe how to configure default, primary, and preferred addresses and interfaces.

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 select 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.

Configure 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.

To configure a different interface to be the primary interface, include the `primary` statement:

```
primary
```

You can include this statement at the following hierarchy level:

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

Configure 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 et-0/0/0.0 255.255.255.255` command is the primary address on interface `et-0/0/0.0`. The primary address flag can also 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`*`]`

Configure 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`) is 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`*`]`

Operational Behavior of Interfaces with the Same IPv4 Address

You can configure the same IP version 4 (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.

If you configure the same IP address on multiple interfaces in the same routing instance, the operating system 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 interfaces that are implicitly and explicitly point-to-point interfaces. The examples also show the **show interfaces terse** command outputs that correspond to the implicit and explicit point-to-point interfaces to display their operational status.

1. **Configuring the same IPv4 address on two non-P2P interfaces:**

```
[edit interfaces]
user@host# show
et-0/1/0 {
     unit 0 {
        family inet {
            address 203.0.113.1/24;
        }
    }
}
```

```
et-3/0/1 {
     unit 0 {
        family inet {
            address 203.0.113.1/24;
```

```
        }
    }
}
```

The following sample output (for the preceding configuration) reveals that only `et-0/1/0.0` was assigned the same IPv4 address `203.0.113.1/24` and its `link` state was `up`, while `et-3/0/1.0` was not assigned the IPv4 address, although its `link` state was up, which means that it will be operational only when it gets a unique IPv4 address other than `203.0.113.1/24`.

show interfaces terse

```
user@host> show interfaces terse et*
Interface               Admin Link Proto    Local                 Remote
        et-0/1/0                up    up
        et-0/1/0.0              up    up   inet     203.0.113.1/24
                                     multiservice
        et-0/1/1                up    down
        et-3/0/0                up    down
        et-3/0/1                up    up
        et-3/0/1.0              up    up   inet
                                     multiservice
```

2. **Configuring the same IPv4 address on (implicit) P2P interfaces:**

```
[edit]
user@host# show
et-0/0/0 {
     unit 0 {
        family inet {
            address 203.0.113.1/24;
        }
    }
}
et-0/0/3 {
     unit 0 {
        family inet {
            address 203.0.113.1/24;
        }
    }
}
```

The following sample output (for the preceding configuration) reveals that both `et-0/0/0.0` and `et-0/0/3.0` were assigned the same IPv4 address `203.0.113.1/24` and that their `link` states were down. The interfaces are down due to an issue with the link and not because the same IPv4 address is assigned to both the interfaces. It is expected that not more than one of the interfaces is up at any given time (following a redundancy scheme outside of the Junos OS Evolved devices scope), because both being up may cause adverse effects.

show interfaces terse

```
user@host> show interfaces terse et*
Interface               Admin Link Proto    Local                 Remote
et-0/0/0                up    up
et-0/0/0.0              up    down inet     203.0.113.1/24
et-0/0/1                up    up
et-0/0/2                up    down
et-0/0/3                up    up
et-0/0/3.0              up    down inet     203.0.113.1/24
et-1/1/0                up    down
et-1/1/1                up    down
et-1/1/2                up    up
et-1/1/3                up    up
et-2/0/0                up    up
et-2/0/1                up    up
et-2/0/2                up    up
et-2/0/3                up    down
```

3. **Configuring the same IPv4 address in multiple instances of a non-P2P interface:**

```
[edit interfaces]
user@host# show
et-0/0/1  {
    vlan-tagging;
         unit 0 {
        vlan-id 1;
            family inet {
            address 10.1.1.1/24;
        }
    }
     unit 1{
        vlan-id 2;
            family inet {
```

```
            address 10.1.1.1/24;
        }
    }
}
```

On a non-P2P interface, you cannot configure the same local address on different units of different interfaces. If you do, a commit error will be thrown and the configuration will fail.

4. **Configuring the same IPv4 address in multiple instances of the same P2P interface:**

```
[edit interfaces]
user@host# show
et-0/0/10  {
     unit 0 {
        tunnel {
            source 10.1.1.1;
            destination 10.1.1.2;
        }
        family inet {
            mtu 1500;
            address 10.2.2.2/24;
        }
    }
     unit 1{
        family inet {
            address 10.2.2.2/24;
        }
    }
}
```

The following sample output (for the preceding configuration) reveals that only one interfaces gets successfully configured on P2P interfaces when you try to configure the same IPv4 address for multiple instance of different interfaces.

show interfaces terse

```
user@host> show interfaces terse | match 10.2.2.2
Interface               Admin Link Proto    Local     Remote
et-0/0/10.0             up    up   inet     10.2.2.2/24
```

Configure Unnumbered Interfaces: Overview

IN THIS SECTION

- Configure an Unnumbered Point-to-Point Interface | 83
- Configure an Unnumbered Ethernet or Demux Interface | 84
- Configure a Secondary Address as a Preferred Source Address for Unnumbered Ethernet or Demux Interfaces | 85
- Restrictions for Unnumbered Ethernet Interface Configurations | 86
- Example: Display the Unnumbered Ethernet Interface Configuration | 87
- Example: Display the Configured Preferred Source Address for an Unnumbered Ethernet Interface | 88
- Example: Display the Configuration for the Unnumbered Ethernet Interface as the Next Hop for a Static Route | 90

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 IP version 6 (IPv6), in which conserving addresses is not a major concern, you can configure unnumbered interfaces to share the same subnet across multiple interfaces.

The IPv6 unnumbered interfaces are supported only 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:

Configure an Unnumbered Point-to-Point Interface

To configure 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. Configure the protocol family, but do not include the `address` statement.

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

NOTE:

- When configuring unnumbered interfaces, you must ensure that a source address is configured on an 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*.

  When you configure a routable address on the `lo0` interface, that address is always the default address. This is ideal because the loopback interface is independent of any physical interfaces and therefore is always accessible.

Configure an Unnumbered Ethernet or Demux Interface

To configure an unnumbered Ethernet or demultiplexing (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 demux 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 IP version 4 (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 therefore 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 or loopback interface that has a logical unit number and configured IP address and is not itself an unnumbered interface. For an unnumbered IP demux 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.

Configure a Secondary Address as 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 demultiplexing (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.

To configure a secondary address on a loopback donor interface as the preferred source address for unnumbered Ethernet or demux interfaces:

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. 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 IP version 4 (IPv4) address family for demux interfaces, and for IPv4 and IP version 6 (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 Unnumbered Ethernet Interface Configurations

The following requirements and restrictions apply when you configure unnumbered Ethernet interfaces:

- The `unnumbered-address` statement currently supports the configuration of unnumbered Ethernet interfaces for IP version 4 (IPv4) and IP version 6 (IPv6) address families.
- You can assign an IP address only to an Ethernet interface that is not already configured as an unnumbered interface.
- You must configure one or more IP addresses on the donor interface for an unnumbered Ethernet interface.
- You cannot configure the donor interface for an unnumbered Ethernet interface as unnumbered.
- An unnumbered Ethernet interface does not support configuration of the following `address` statement options: `arp`, `broadcast`, `primary`, `preferred`, or `vrrp-group`.
- You can run Internet Group Management Protocol (IGMP) and Physical Interface Module (PIM) only on unnumbered Ethernet interfaces that directly face the host and have no downstream PIM neighbors. You cannot run either IGMP or PIM on unnumbered Ethernet interfaces that act as upstream interfaces in a PIM topology.
- You can run OSPF over unnumbered Ethernet interfaces configured as a point-to-point (P2P) connection. However, you cannot run OSPF or IS-IS on unnumbered Ethernet interfaces that are not configured as P2P.

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 that 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, the operating system uses only the first configuration. In this scenario, the remaining address configurations are ignored and can leave interfaces without an address. An interface that does 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 et-0/0/1.0 is ignored:

```
interfaces {
  et-0/0/0 {
    unit 0 {
      family inet {
        address 192.168.1.1/24;
      }
    }
  }
  et-0/0/1 {
    unit 0 {
     family inet {
       address 192.168.1.1/24;
     }
  }
}
```

Example: Display the Unnumbered Ethernet Interface Configuration

IN THIS SECTION

- Purpose | 88
- Action | 88

Purpose

To display the configured unnumbered interface at the [edit interfaces *interface-name* unit *logical-unit-number*] hierarchy level:

- Unnumbered interface —et-1/0/0
- Donor interface —et-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 {
    et-0/0/0 {
        unit 0 {
            family inet {
                address 4.4.4.1/24;
            }
        }
    }
    et-1/0/0 {
        unit 0 {
            family inet {
                unnumbered-address et-0/0/0.0;
            }
        }
    }
}
```

Example: Display the Configured Preferred Source Address for an Unnumbered Ethernet Interface

IN THIS SECTION

Purpose | 89

Action | 89
Meaning | 90

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 —et-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 {
    et-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 an unnumbered Ethernet interface `et-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.

Example: Display the Configuration for the Unnumbered Ethernet Interface as the Next Hop for a Static Route

IN THIS SECTION

- Purpose | **90**
- Action | **90**
- Meaning | **91**

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 —et-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 {
    et-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 et-0/0/0.0.

```
static {
route 7.7.7.1/32 {
    qualified-next-hop et-0/0/0.0;
}
}
```

Meaning

In this example, `et-0/0/0` is the unnumbered interface. A loopback interface, `lo0`, is the donor interface from which `et-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 `et-0/0/0.0`.

Protocol MTU

IN THIS SECTION

- Configure the Protocol MTU | **92**

Overview

The default protocol MTU depends on your device and the interface type. When you initially configure an interface, the protocol MTU is calculated automatically. If you subsequently change the media MTU, the protocol MTU on existing address families automatically changes.

If you reduce the media MTU size but one or more address families are already configured and active on the interface, you must also reduce the protocol MTU size. 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.

If you do not configure an MPLS MTU, Junos OS Evolved 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. 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 as follows:

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

You can configure the protocol MTU on all tunnel interfaces.

Configure the Protocol MTU

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

To configure the protocol MTU:

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. Include the `mtu` statement for each family you want to configure with a non-default MTU value.

   If you configure the protocol MTU for any family, the configured value is applied to all families that are configured on the logical interface.

   ```
   [edit interfaces interface-name unit logical-unit-number]
   user@host# set family family mtu bytes
   ```

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 families. We do not recommend configuring different MTU size values for `inet` and `inet6` families that are configured on the same logical interface.

Disable the Removal of Address and Control Bytes

For some interfaces, the address and control bytes are removed by default before the packet is encapsulated into a tunnel.

However, you can disable the removal of address and control bytes.

To disable the removal of address and control bytes, 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]`

SEE ALSO

keep-address-and-control

Disable 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 level:

`[edit 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.

Apply a Filter to an Interface

IN THIS SECTION

- Define Interface Groups in Firewall Filters | 94
- Apply a Filter to an Interface | 95

Define Interface Groups in Firewall Filters

IN THIS SECTION

- Filter-Based Forwarding on the Output Interface | 95

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]`

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 the 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 heading to a different routing instance, since the routing table is not set up to handle the source lookup.

Apply 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 ];
}
```

You can include up to 16 filter names 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*]

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: MPLS family firewall filters applied on the output interface are not supported on the PTX10003 router, due to product limitation.

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.

If you apply the filter to interface lo0, it is applied to packets received or transmitted by the Routing Engine.

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

```
[edit interfaces]
et-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 `et-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 egress interface `et-0/0/3.0`.

3. The output filter installed at `et-0/0/3.0` redirects the packet to routing table `fbf.inet.0`.

4. The packet matches entry `10.50.100.0/25` in the `fbf.inet.0` table, and the packet finally leaves the router from interface `et-2/0/0.0`.

```
[edit interfaces]
et-0/0/3 {
    unit 0 {
        family inet {
            filter {
                output fbf;
            }
            address 10.50.10.2/25;
        }
    }
}
et-1/2/0 {
    unit 0 {
        family inet {
            address 10.50.50.2/25;
        }
    }
}
et-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 et-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];
    }
}
```

Enable Source Class and Destination Class Usage

IN THIS SECTION

- Source Class and Destination Class Usage Overview | **99**
- Enable Source Class and Destination Class Usage | **103**

Source Class and Destination Class Usage Overview

For interfaces that carry IP version 4 (IPv4), IP version 6 (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) accounting 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 located in the forwarding table.

NOTE: Neither SCU nor destination class usage (DCU) accounting works 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.

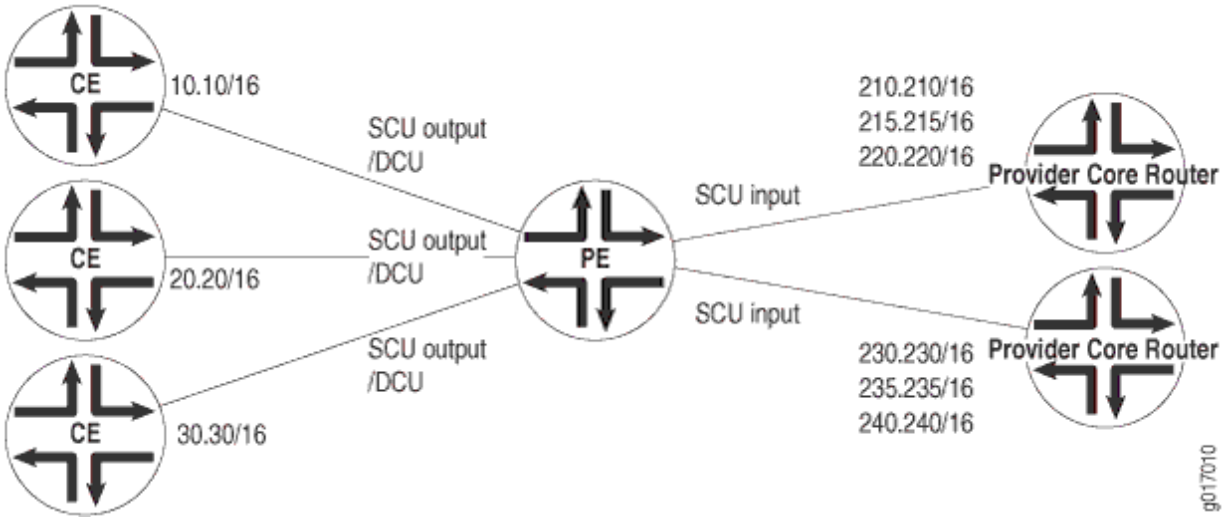
Destination class usage (DCU) counts packets from customers by performing lookup of the IP destination address. DCU makes it possible to track traffic originating from the customer edge and destined for specific prefixes on the provider core router.

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 1 illustrates an 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 that are sent from prefix `210.210/16` and `215.215/16` and that are transmitted on a specific output interface.

Figure 6: 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 operating system 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 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)]`

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 Modular Port Concentrator/Modular Interface Card (MPC/MIC) interfaces, a single incoming packet is counted for both SCU and DCU accounting if both SCU and DCU are configured. To ensure that 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 operating system 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`.

After you enable accounting on an interface, the operating system 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.

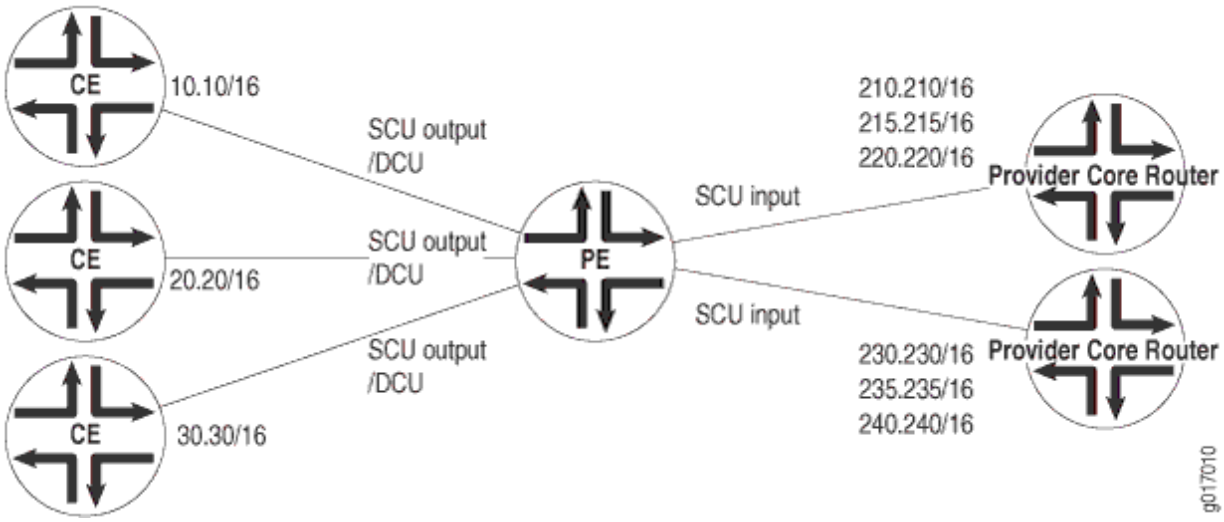
NOTE: When configuring policy action statements, you can configure only one source class for each matching route. In other words, more than one source class cannot be applied to the same route.

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: You cannot enable DCU counters on the label-switched interface (LSI) that is created dynamically when the `vrf-table-label` statement is configured within a VRF.

Enable Source Class and Destination Class Usage

Figure 7: Prefix Accounting with Source and Destination Classes



Before you can enable source class usage (SCU) and destination class usage (DCU), you must configure DCU and SCU output on one interface:

```
[edit]
interfaces {
    et-6/1/0 {
        unit 0 {
            family inet {
                accounting {
                    destination-class-usage;
                    source-class-usage {
                        output;
                    }
                }
            }
        }
    }
}
```

To enable source class and destination class usage:

1. **Complete the 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 `et-0/0/2` in the OSPF process.

2.

   ```
   Router A
   [edit]
   interfaces {
       et-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 et-0/0/2.0;
            interface lo0.0;
        }
    }
}
```

3. Apply the policy to the forwarding table, configuring a route filter policy statement that matches the prefixes of the loopback addresses from routers A and B.

   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.

   When you 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.

```
Router SCU
[edit]
interfaces {
    et-0/0/1 {
        unit 0 {
            family inet {
                accounting {
                    source-class-usage {
                        input;
                        output;
                    }
```

```
                }
                address 10.255.50.1/24;
            }
        }
    }
    et-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 et-0/0/1.0;
            interface et-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. Configure 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 `et-0/0/4` in the OSPF process.

```
Router B
interfaces {
    et-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 et-0/0/4.0;
            interface lo0.0;
        }
```

```
    }
}
```

5. Configure a virtual loopback tunnel interface on a provider edge router equipped with a tunnel PIC.

   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:

```
Enabling Packet Counting for Layer 3 VPNs
[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.

   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.

```
[edit routing-instances]
VPN-A {
    instance-type vrf;
    interface et-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]
et-2/1/0 {
    unit 0 {
        family inet {
            accounting {
                source-class-usage {
                    output;
                }
            }
        }
    }
}
```

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, as follows:

1. 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.
2. The IP packets are then translated into broadcast IP packets, which flood the target subnet only through the LAN interface, and all hosts on the target subnet receive the IP packets. The packets are discarded If no LAN interface exists,
3. The final step in the sequence depends on targeted broadcast:
   - 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.

You can configure targeted broadcast to forward the IP packets only to an egress interface. This is helpful when the router is flooded with packets to process, or to both an egress interface and the Routing Engine.

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 traffic and not as local next-hop traffic, and you can apply a firewall filter only to local next-hop routes for traffic directed towards the Routing Engine.

Configure Targeted Broadcast

IN THIS SECTION

- Configure Targeted Broadcast and Its Options | **111**
- Display Targeted Broadcast Configuration Options | **112**

The following sections explain how to configure targeted broadcast on an egress interface and its options:

Configure Targeted Broadcast and Its Options

You can configure targeted broadcast on an egress interface with different options.

Either of these configurations is acceptable:

- You can 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.
- 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:

- 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
```

- 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
```

Display Targeted Broadcast Configuration Options

IN THIS SECTION

- Example: Forward IP Packets on the Egress Interface and to the Routing Engine | **112**
- Example: Forward IP Packets on the Egress Interface Only | **113**

The following example topics display targeted broadcast configuration options:

Example: Forward IP Packets on the Egress Interface and to the Routing Engine

IN THIS SECTION

- Purpose | **112**
- Action | **113**

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 et-2/0/0, the unit value is set to 0, and 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;
}
```

Example: Forward IP Packets on the Egress Interface Only

IN THIS SECTION

- Purpose | **113**
- Action | **113**

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 et-2/0/0, the unit value is set to 0, and the protocol family is set to inet.

```
[edit interfaces interface-name unit interface-unit-number family inet]
user@host#show
targeted-broadcast {
    forward-only;
}
```

2
CHAPTER

Other Interfaces

Discard Interfaces | 115

Loopback Interfaces | 118

Discard Interfaces

IN THIS SECTION

- Discard Interface Overview | 115
- Discard Interface Configuration | 116

The discard interface *dsc* is not a physical interface but a virtual interface that discards packets.

Discard Interface Overview

The discard interface is a virtual interface that silently discards packets as they arrive. The discard interface is especially useful when the network is under a denial-of-service (DoS) attack. You (the network administrator) can configure a policy to drop millions of requests from being sent to a given target address or set of addresses.

You can configure which traffic Junos OS Evolved forwards to the discard interface and what it does with that traffic. A local policy determines which traffic Junos OS Evolved forwards to the discard interface. Junos OS Evolved performs the action specified by an output filter before it discards the traffic.

Benefits

- With a discard interface, you can configure filters for counting, logging, and sampling the traffic before any type of attack occurs. Discard static routes don't give you the same flexibility.
- The discard interface allows you to identify the ingress point of a DoS attack. When your network is under attack, Junos OS Evolved identifies the target host IP address while the local policy forwards attacking packets to the discard interface.

Discard Interface Configuration

IN THIS SECTION

- Configure the Discard Interface | 116
- Configure an Output Policy | 117

Keep the following guidelines in mind when configuring the discard interface:

- Only the *logical interface* unit 0 is supported.
- A discard interface can have only one logical unit (unit 0), but you can configure multiple IP addresses on that unit.
- 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).

Configure the Discard Interface

To configure a discard interface:

1. In configuration mode, navigate to the `[edit interfaces]` hierarchy level.

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

2. Configure the discard interface. Note that you must use `dsc` to configure the discard interface and ensure that no other discard interface is already configured.

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

3. Configure the logical interface (unit 0) and the protocol family.

```
[edit interfaces dsc]
user@host# edit unit 0 family family
```

4. (Optional) Apply an output filter to the discard interface.

```
[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 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
```

Loopback Interfaces

IN THIS SECTION

- Loopback Interface Overview | 118
- Loopback Interface Configuration | 119

This topic discusses about the use of loopback interface, step-by-step procedure on how to configure loopback interfaces with examples.

Loopback Interface Overview

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 interface (`lo0.16384`). The internal loopback interface is a particular instance of the loopback interface with the logical unit number 16384.

You use the loopback interface to identify the device. While you can use any interface address 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 configuration or operation. You can use the loopback interface to address these issues.

Benefits

- 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 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 logical unit to filter packets originating from, or destined for, the Routing Engine.
- Junos OS Evolved creates a separate loopback interface for the internal routing instance, which prevents any filter on `lo0.0` from disrupting internal traffic.

Loopback Interface Configuration

IN THIS SECTION

Configure the Loopback Interface | 120

- Example: Configure Two Addresses on the Loopback Interface with Host Routes | 121

You (a system administrator, network administrator, or end user) can use this procedure to configure the loopback interface on your device.

Configure the Loopback Interface

When specifying the loopback address on a device, do not include a destination prefix. Also, in most cases, specify a loopback address only on unit 0 and no others.

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 some applications, such as SSL for Junos XML protocol, at least one address for the interface `lo0.0` must be `127.0.0.1`.

You can configure loopback interfaces using a host (recommended), a subnetwork address for both `inet` and `inet6` address families, or an ISO network entity title (NET) address for the `iso` address family. Many protocols require a loopback address as their source address. Configuring a loopback address as a donor interface for unnumbered interfaces enables these protocols to run on unnumbered interfaces.

In some cases, the loopback interface can also be the router identifier (router ID). If the router ID is not explicitly configured, the device determines its router ID as shown in the following table:

Table 3: Default Router ID

If the loopback interface is:	Then the default router ID is:
Configured	The loopback interface
Not configured	The lowest IP address of any interface in operational state `up`

In both cases, the router ID changes when the operational state of the interface changes. Therefore, we recommend configuring the address on a stable loopback interface.

If you configure more than one address on the loopback interface, we recommend that you configure one to be the primary address. The device selects the primary address as the router ID when the router

ID is not configured. The device also uses the primary address as the default source address for traffic sourced from the loopback interface by the Routing Engine.

To configure the physical loopback interface (`lo0`), 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;
        }
    }
}
```

You can configure one or more addresses on the loopback interface. You can configure more than just `unit 0` for `lo0`, but you must place each additional unit in a separate routing instance.

Example: Configure Two Addresses on the Loopback Interface with Host Routes

In the following example, the user configures 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 10.0.0.1
[edit interfaces lo0 unit 0 family inet]
user@host# set address 172.16.0.1
[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;
            172.16.0.1/32;
```

```
        }
    }
}
```

3
CHAPTER

Troubleshooting Interfaces

Troubleshooting Interfaces | 124

Troubleshooting Interfaces

IN THIS SECTION

- Troubleshooting: Management Interface Link Is Down for Junos OS Evolved | 124
- Troubleshooting: Invalid Port Speed Configuration for Junos OS Evolved | 127
- Troubleshooting: Faulty Ethernet Physical Interface for Junos OS Evolved | 131

This topic discusses various troubleshooting scenarios.

Troubleshooting: Management Interface Link Is Down for Junos OS Evolved

IN THIS SECTION

- Problem | 124
- Diagnosis | 125
- Resolution | 126

Problem

Description

The `Ethernet Link Down` alarm is raised when you run the `show chassis alarm` operational mode command on the following devices:

- ACX routers (ACX7100-32C, ACX7100-48L)
- PTX routers (PTX10001-36MR, PTX10003, PTX10004, PTX10008)
- QFX switches (QFX5130-32CD, QFX5220-32CD, QFX5220-128C)

Diagnosis

Perform the following tests to check if the management interface is down on the primary 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
   2020-10-19 11:13:02 MYT Major Host 1  re0:mgmt-0.0 : Ethernet Link Down
   ```

   Is the alarm `Ethernet Link Down` displayed against the management interface of the primary Routing Engine (Host 0)?

   - Yes: Contact Juniper Networks Technical Assistance Center (JTAC) for further assistance.
   - No: Continue to the next diagnostic test.

1. Run the `show interfaces re0:mgmt-0` and the `show interfaces re0:mgmt-0 terse` operational mode commands.

   show interfaces re0:mgmt-0

   ```
   user@host0> show interfaces re0:mgmt-0
   Physical interface: re0:mgmt-0, Enabled, Physical link is Up
     Interface index: 1001, SNMP ifIndex: 193
     Type: Ethernet, Link-level type: Ethernet, MTU: 1500, Speed: 1Gbps, Auto-negotiation:
   Enabled
     Device flags   : Present Running
     Interface flags: SNMP-Traps
   ...
   ```

   show interfaces re0:mgmt-0 terse

   ```
   user@host0> show interfaces re0:mgmt-0 terse
   Interface               Admin Link Proto    Local                 Remote
   re0:mgmt-0              up    up
   re0:mgmt-0.0            up    up   inet     10.100.100.1/30
   ```

   Is the management interface on the primary Routing Engine `up`?

- Yes: Continue to resolution.
- No: Contact JTAC for further assistance.

Resolution

To Resolve This Issue

The chassis alarm was raised for the management interface in the backup Routing Engine (Host 1) and not for the primary Routing Engine (Host 0).

Implement one of the following solutions on the backup Routing Engine to resolve this issue:

- Disable the management 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 re1:mgmt-0 interface.

     ```
     [edit groups re1]
     user@host1# set interfaces re1:mgmt-0 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: Invalid Port Speed Configuration for Junos OS Evolved

IN THIS SECTION

- Problem | 127
- Diagnosis | 127
- Resolution | 130

Problem

Description

The `Invalid Port Speed Configuration` alarm is displayed when you run the `show system alarms` operational mode command.

For example:

```
user@host> show system alarms

Alarm time               Class  Description
2021-12-06 23:09:21 UTC  Minor  FPC-0 PIC-0: Invalid Port Speed Configuration
```

Diagnosis

Run the `show picd config` command from the `cli-pfe` level to identify invalid port speed configurations. Running the command displays details of port speed/channelization configurations, including which ports are valid and reasons for any invalid PIC configurations.

For example:

```
root@re0:~# cli-pfe
root@re0:pfe> show picd config
```

```
pic_info_table               :
              default   config     config       config   computed
pic_or_port   speed     pic_mode   port_speed   valid    speed
supported_speeds                                          hidden_speeds
-----------   -------   --------   ----------   ------   --------
-------------------------------------------------------   -------------
pic-0/0       -         -          -            yes      -
port-0/0/0    1x400G    -          1x100G       -        1x100G     [ 1x400G 1x200G 4x10G 1x40G
4x25G 1x100G 4x100G 2x200G ]   [ ]
port-0/0/1    1x400G    -          1x100G       -        1x100G     [ 1x400G 1x200G 4x10G 1x40G
4x25G 1x100G 4x100G 2x200G ]   [ ]
port-0/0/2    1x400G    -          1x100G       -        1x100G     [ 1x400G 1x200G 4x10G 1x40G
4x25G 1x100G 4x100G 2x200G ]   [ ]
port-0/0/3    1x400G    -          1x100G       -        1x100G     [ 1x400G 1x200G 4x10G 1x40G
4x25G 1x100G 4x100G 2x200G ]   [ ]
...
port-0/0/11   1x400G    -          -            -        1x400G     [ 1x400G 1x200G 4x10G 1x40G
4x25G 1x100G 4x100G 2x200G ]   [ ]
port-0/0/12   1x400G    -          -            -        1x400G     [ 1x400G 1x200G 4x10G 1x40G
4x25G 1x100G 4x100G 2x200G ]   [ ]
port-0/0/13   1x400G    -          -            -        1x400G     [ 1x400G 1x200G 4x10G 1x40G
4x25G 1x100G 4x100G 2x200G ]   [ ]
port-0/0/14   1x400G    -          -            -        1x400G     [ 1x400G 1x200G 4x10G 1x40G
4x25G 1x100G 4x100G 2x200G ]   [ ]
...
     usr_config                  :
         pic_usr_config          :
             pic_name            : pic-0/0
             pic_mode_valid      : false
             pic_mode            :
             num_ports_valid     : false
             num_ports           : 0
             config_invalid      : true
             invalid_reason      : Port 12: Speed 4x40G not supported
                                   Port 13: Speed 4x40G not supported
             ptp_mode            : false
```

```
            port_usr_config      :
                port_name        : port-0/0/0
                port_speed_valid : true
                port_speed       : 1x100G
                sub_ports_cfg    : false
            port_usr_config      :
                port_name        : port-0/0/1
                port_speed_valid : true
                port_speed       : 1x100G
                sub_ports_cfg    : false
            port_usr_config      :
                port_name        : port-0/0/10
                port_speed_valid : true
                port_speed       : 1x40G
                sub_ports_cfg    : false
            port_usr_config      :
                port_name        : port-0/0/11
                port_speed_valid : true
                port_speed       : 1x40G
                sub_ports_cfg    : false
            port_usr_config      :
                port_name        : port-0/0/12
                port_speed_valid : true
                port_speed       : 4x40G
                sub_ports_cfg    : true
            port_usr_config      :
                port_name        : port-0/0/13
                port_speed_valid : true
                port_speed       : 4x40G
                sub_ports_cfg    : true
            port_usr_config      :
                port_name        : port-0/0/20
                port_speed_valid : true
                port_speed       : 1x100G
                sub_ports_cfg    : false
```

Do the results of the `show picd config` command indicate an invalid port speed configuration?

- Yes: Continue to resolution.
- No: Contact Juniper Networks Technical Assistance Center (JTAC) for further assistance.

In the example, the results indicate:

```
Port 12: Speed 4x40G not supported
Port 13: Speed 4x40G not supported
```

Resolution

To Resolve This Issue

The system alarm was raised for a port speed configuration that was not supported. Configure all ports to a valid port speed to resolve the issue. Port speed is configured at the PIC-level.

> **NOTE**: When an invalid configuration is applied on a port, the currently configured speed/channelization on the other ports of the PIC remain in effect until the next reboot of the device or the next picd app restart. After device reboot, or picd restart, all ports of that PIC revert to default speed. The invalid state and the resulting effect of all ports being in default speed remains in effect until the invalid configuration is corrected.

To determine valid port speeds of a logical PIC, run the `show chassis pic fpc-slot slot-number pic-slot slot-number` command.

The example contains an unsupported port speed/channelization configuration on ports 0/0/12 and 0/0/13. To determine valid port speeds for the example, run `show chassis pic fpc-slot 0 pic-slot 0`.

```
root@re0> show  chassis pic fpc-slot 0 pic-slot 0
---
Port speed information:
Port  PFE      Capable Port Speeds
 0      0      1x40G 1x100G 4x25G 4x10G
 1      0      1x40G 1x100G 4x25G 4x10G
 4      0      1x40G 1x100G 4x25G 4x10G
 5      0      1x40G 1x100G 4x25G 4X10G
 8      0      1x40G 1x100G 4x25G 4x10G
 9      0      1x40G 1x100G 4x25G 4x10G
12     0      1x40G 1x100G 4x25G 4x10G
13     0      1x40G 1x100G 4x25G 4x10G
```

The invalid port speed example can be resolved by configuring ports 0/0/12 and 0/0/13 with one of the following valid port speeds:

```
12      1x40G 1x100G 4x25G 4x10G
13      1x40G 1x100G 4x25G 4x10G
```

Troubleshooting: Faulty Ethernet Physical Interface for Junos OS Evolved

IN THIS SECTION

- Check the Cable Connection | **131**
- Check the Physical Link Status of the Interface | **133**
- Check the Interface Statistics in Detail | **134**
- Perform the Loopback Diagnostic Test | **137**
- Check for Other Possibilities | **140**
- Enable a Physical Interface | **141**

You can follow the basic troubleshooting actions as explained in the following topics to troubleshoot an Ethernet physical interface on a device that Junos OS Evolved supports.

Check the Cable Connection

IN THIS SECTION

- Problem | **132**
- Diagnosis | **132**
- Resolution | **132**

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 "Check the Physical Link Status of the Interface" on page 133.
   - No: See Resolve the Cabling Issue.

Resolution

Resolve the 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. If the existing cable is damaged, replace it with a known good cable.
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 Check the Fiber Optic Cable Integrity.
4. Connect the correct small form-factor pluggable transceiver (SFP) on both sides of the cable.

Check the Fiber Optic Cable Integrity

To check the integrity of the fiber optic cable with an external cable diagnostic testing tool:

> **NOTE**: A single-mode fiber cable must be connected to a single-mode interface.
> 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 the 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.

Check the Physical Link Status of the Interface

IN THIS SECTION

- Problem | 133
- Solution | 133
- Diagnosis | 134

Problem

Description

Unable to transmit or 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 et-5/0/1 interface:

```
user@host> show interfaces et-5/0/1 media
Physical interface: et-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   : 2020-11-30 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
```

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 "Check the Interface Statistics in Detail" on page 134.
   - No: Continue to the next diagnostic test.

1. 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 Resolve the Cabling Issue.
   - No: Continue.

Check the Interface Statistics in Detail

IN THIS SECTION

- Problem | 135
- Solution | 135
- Diagnosis | 137

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 the et-5/0/1 interface:

```
user@host> show interfaces et-5/0/1 extensive
Physical interface: et-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
```

Diagnosis

1. Does the `Policed discards`, `L2 channel errors`, `Input DA rejects`, or `Input SA rejects` field display any errors?

   - Yes: Resolve the errors as needed. Resolving these errors is beyond the scope of this topic.
   - No: Continue with "Perform the Loopback Diagnostic Test" on page 137.

Perform the Loopback Diagnostic Test

IN THIS SECTION

- Problem | **138**
- Solution | **138**
- Diagnosis | **139**

Problem

Description

The interface cable is connected correctly, and no alarms or errors are associated with the Ethernet physical interface, but 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 an internal loopback diagnostic test on an Ethernet interface, for example on et-5/0/1 interface:

1. In configuration mode, go to the `[edit interfaces et-5/0/1]` hierarchy level.

```
[edit]
user@host# edit interface et-5/0/1
```

2. Set the `ether-options` option as loopback, commit the configuration, and quit configuration mode.

```
[edit interfaces et-5/0/1
user@host# set ether-options loopback
user@host# commit
user@host# quit
```

3. In operational mode, execute the `show interfaces et-5/0/1 media` command.

```
user@host> show interfaces et-5/0/1 media
Physical interface: et-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. Then, check the status of the physical link with the `show interfaces` *`interface-name`* `media` operational mode command.

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 "Check for Other Possibilities" on page 140 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 "Check for Other Possibilities" on page 140 section.
- No: Contact Juniper Networks Technical Assistance Center (JTAC) for further assistance.

Check for Other Possibilities

IN THIS SECTION

- Problem | 140
- Solution | 140
- Diagnosis | 140

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 such as an et-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 an et-5/0/1 interface:

```
user@host> show interfaces et-5/0/1 terse
     Interface               Admin Link Proto    Local                 Remote
     et-5/0/1                up    up
     et-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 "Enable a Physical Interface" on page 141.

- No: Continue to the next diagnostic test.

1. Are the `speed`, `duplex`, and `auto-negotiation` fields in the output of `show interfaces` *`interface-name`* `extensive` operational mode command set correctly for the interface?

   NOTE: Check if the associated device supports speed and auto-negotiation settings: 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.

   - Yes: Check *Ethernet Interfaces User Guide for Routing Devices* for more troubleshooting tips.
   - No: Contact JTAC for further assistance.

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 the et-5/0/1 interface:

```
user@host# show et-5/0/1
```

```
disable;
```

3. Enable the interface, and commit.

```
[edit interfaces
user@host# delete interface-name disable
user@host# commit
```