Engineering
Simplicity

Junos® OS

Interfaces Fundamentals for Junos OS

Published
2025-12-12

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

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 Interfaces Fundamentals for Junos OS

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 | xi

1 Device Interfaces

Overview | 2

Device Interfaces Overview | 4

Device Interfaces Overview | 4

Types of Interfaces | 5

Interface Naming Overview | 6

Interface Descriptors Overview | 24

Physical Part of an Interface Name | 26

Interface Names for ACX Series, PTX Series, and QFX Series Devices | 26

Interface Names for M Series and T Series Routers | 26

Interface Names for MX Series Routers | 27

Displaying Interface Configurations | 27

Interface Encapsulations Overview | 28

Understanding Transient Interfaces | 42

Understanding Services Interfaces | 43

Understanding Container Interfaces | 44

Understanding Internal Ethernet Interfaces | 47

Understanding Interfaces on ACX Series Universal Metro Routers | 49

TX Matrix Plus and T1600 Router (Routing Matrix) Management Ethernet Interfaces | 53

T1600 Routers (Routing Matrix) Internal Ethernet Interfaces | 53

Physical Interface Properties | 54

Physical Interface Properties Overview | 55

Configure the Interface Description | 55

How to Specify an Aggregated Interface | 57

Configure the Link Characteristics | 57

Interface Speed | 58

- Configuring the Interface Speed on Ethernet Interfaces | 58
- Configure the Aggregated Ethernet Link Speed | 60
- Configure the SONET/SDH Interface Speed | 63

Forward Error Correction (FEC) | 65

- Benefits of FEC | 65
- Overview | 65
- Configure FEC | 66

Interface Aliases | 66

Example: Add an Interface Alias Name | 68

- Requirements | 69
- Overview | 69
- Configuration | 69
- Verification | 72

Clock Source Overview | 73

Configure the Clock Source | 74

Interface Encapsulation on Physical Interfaces | 75

- Encapsulation Capabilities | 75
- Encapsulation Types | 76
- Configure Encapsulation on a Physical Interface | 78
- Display the Encapsulation on a Physical SONET/SDH Interface | 79
- Configure Interface Encapsulation on PTX Series Routers | 80

Keepalives | 82

Understanding Unidirectional Traffic Flow on Physical Interfaces | 85

Enable Unidirectional Traffic Flow on Physical Interfaces | 86

Enable SNMP Notifications on Physical Interfaces | 86

Accounting for Physical Interfaces | 87

- Overview | 88

Configure an Accounting Profile for a Physical Interface | 88

Display the Accounting Profile | 89

Disable a Physical Interface | 91

How to Disable a Physical Interface | 91

Example: Disable a Physical Interface | 92

Configure Ethernet Loopback Capability | 93

Configure Short Reach Mode on QFX5100-48T | 93

Configure Flow Control | 95

Set the Mode on an SFP+ or SFP+ MACsec Uplink Module | 96

Set the Operational Mode on a 2-Port 40-Gigabit Ethernet QSFP+/100-Gigabit Ethernet QSFP28 Uplink Module | 97

Configure the Media Type on Dual-Purpose Uplink Ports | 98

Logical Interface Properties | 100

Logical Interface Properties Overview | 100

Specify the Logical Interface Number | 101

Add a Logical Unit Description to the Configuration | 101

Configure the Interface Bandwidth | 102

Configure Interface Encapsulation on Logical Interfaces | 103

Understand the Interface Encapsulation on Logical Interfaces | 103

Configure the Encapsulation on a Logical Interface | 104

Display the Encapsulation on a Logical Interface | 104

Configure Interface Encapsulation on PTX Series Routers | 105

Configure a Point-to-Point Connection | 107

Configure a Multipoint Connection | 107

Configure Dynamic Profiles for PPP | 108

Overview of Accounting for the Logical Interface | 108

Accounting Profiles Overview | 109

Configure Accounting for the Logical Interface | 109

Display the Accounting Profile for the Logical Interface | 111

Enable or Disable SNMP Notifications on Logical Interfaces | 112

Disable a Logical Interface | 113

Understanding Interfaces | 114

Network Interfaces | 114

Understanding Interface Naming Conventions | 124

Understanding Management Interfaces | 143

Media MTU and Protocol MTU | 145

MTU Overview | 146

Media MTU Overview | 146

Configure the Media MTU | 147

Protocol MTU | 148

Encapsulation Overhead by Interface Encapsulation Type | 149

Media MTU Sizes by Interface Type | 151

MTU and MACsec | 157

Platform-Specific MTU Behavior | 161

Interface Ranges for Physical Interfaces | 162

Configure Interface Ranges | 163

Expanded Interface Range Statements | 168

Configuration Inheritance Priority | 170

Configuration Inheritance for Member Interfaces | 171

Common Configuration Inheritance | 172

Configuration Group Inheritance | 173

Configuration Expansion Where Interface Range Is Used | 175

Damping Interfaces | 176

Physical Interface Damping Overview | 176

Configure Damping of Shorter Physical Interface Transitions | 184

Configure Damping of Aggregated Ethernet Interface Transitions | 185

Configure Damping of Longer Physical Interface Transitions | 186

Example: Configure Physical Interface Damping | 188

- Requirements | 188
- Overview | 188
- Configuration | 189
- Verification | 190

Protocol Family and Interface Address Properties | 191

Configure the Protocol Family | 192

Assign the Interface Address | 193

Configure Default, Primary, and Preferred Addresses and Interfaces | 195

- Default, Primary, and Preferred Addresses and Interfaces | 195
- Configure the Primary Interface for the Router | 196
- Configure the Primary Address for an Interface | 197
- Configure the Preferred Address for an Interface | 198

Operational Behavior of Interfaces with the Same IPv4 Address | 198

Configure IPCP Options for Interfaces with PPP Encapsulation | 202

Configure Unnumbered Interfaces: Overview | 205

- Configure an Unnumbered Point-to-Point Interface | 205
- Configure an Unnumbered Ethernet or Demux Interface | 206
- Configure a Secondary Address as a Preferred Source Address for Unnumbered Ethernet or Demux Interfaces | 208
- Restrictions for Unnumbered Ethernet Interface Configurations | 208
- Example: Display the Unnumbered Ethernet Interface Configuration | 210
- Example: Display the Configured Preferred Source Address for an Unnumbered Ethernet Interface | 211
- Example: Display the Configuration for the Unnumbered Ethernet Interface as the Next Hop for a Static Route | 213

Protocol MTU | 214

Disable the Removal of Address and Control Bytes | 215

Disable the Transmission of Redirect Messages on an Interface | 216

Apply a Filter to an Interface | 216

Define Interface Groups in Firewall Filters | 217

Apply a Filter to an Interface | 218

Enable Source Class and Destination Class Usage | 223

Source Class and Destination Class Usage Overview | 223

Enable Source Class and Destination Class Usage | 227

Overview | 234

Configure Targeted Broadcast | 236

Configure Targeted Broadcast | 237

Display Targeted Broadcast Configuration Options | 238

2 Other Interfaces

Discard Interfaces | 242

Discard Interface Overview | 242

Discard Interface Configuration | 243

Configure the Discard Interface | 243

Configure an Output Policy | 244

IP Demultiplexing Interfaces | 245

Demultiplexing Interface Overview | 246

Configuring an IP Demultiplexing Interface | 249

Configuring an IP Demux Underlying Interface | 249

Configuring the IP Demux Interface | 252

Configuring MAC Address Validation on Static IP Demux Interfaces | 254

Configuring a VLAN Demultiplexing Interface | 255

Configuring a VLAN Demux Underlying Interface | 255

Configuring the VLAN Demux Interface | 257

Configuring MAC Address Validation on Static VLAN Demux Interfaces | 260

Verifying a Demux Interface Configuration | 261

Loopback Interfaces | 263

Loopback Interface Overview | 264

Loopback Interface Configuration | 265

Configure the Loopback Interface | 265

Example: Configure Two Addresses on the Loopback Interface with Host Routes | 267

Example: Configure Two Addresses on the Loopback Interface with Subnetwork Routes | 267

Example: Configure an IPv4 and an IPv6 Address on the Loopback Interface with Subnetwork Routes | 268

Serial Interfaces | 269

Serial Interfaces Overview | 269

Configure the Serial Line Protocol | 275

Configure the Serial Line Protocol | 275

Serial Interface Default Settings | 276

Configure the Serial Clocking Mode | 280

Configure the Serial Clocking Mode | 280

Invert the Serial Interface Transmit Clock | 281

Configure the DTE Clock Rate | 282

Configure the Serial Signal Handling | 283

Configure the Serial DTR Circuit | 286

Configure Serial Signal Polarities | 287

Configure Serial Loopback Capability | 288

Configure Serial Line Encoding | 290

Configure Serial Interfaces on SRX devices | 291

Basic Serial Interface Configuration | 292

Delete the Serial Interface | 293

Example: Configure serial interface on 8-Port Synchronous Serial GPIM | 293

Verification | 299

3 Monitor and Troubleshooting Interfaces

Monitor Interfaces | 306

Trace Interface Operations Overview | 306

Trace Operations of an Individual Router Interface | 306

Trace Operations of the Interface Process | 307

Troubleshooting Interfaces | 310

Troubleshooting: em0 Management Interface Link is Down | 310

Troubleshooting: fxp0 Management Interface Link is Down | 313

Troubleshooting: Faulty Ethernet Physical Interface on an M Series, an MX Series, or a T Series Router | 315

Check the Cable Connection | 316

Check the Physical Link Status of the Interface | 317

Check the Interface Statistics in Detail | 319

Perform the Loopback Diagnostic Test | 322

Check for Other Possibilities | 324

Enable a Physical Interface | 326

Time Domain Reflectometry on ACX Series Routers Overview | 327

Diagnose a Faulty Twisted-Pair Cable on ACX Series Routers | 330

4 Configuration Statements and Operational Commands

Common Output Fields Description | 335

Improvements to Interface Transmit Statistics Reporting | 345

Junos CLI Reference Overview | 346

About This Guide

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

1

CHAPTER

Device Interfaces

IN THIS CHAPTER

- Overview | 2
- Device Interfaces Overview | 4
- Physical Interface Properties | 54
- Logical Interface Properties | 100
- Understanding Interfaces | 114
- Media MTU and Protocol MTU | 145
- Interface Ranges for Physical Interfaces | 162
- Damping Interfaces | 176
- Protocol Family and Interface Address Properties | 191

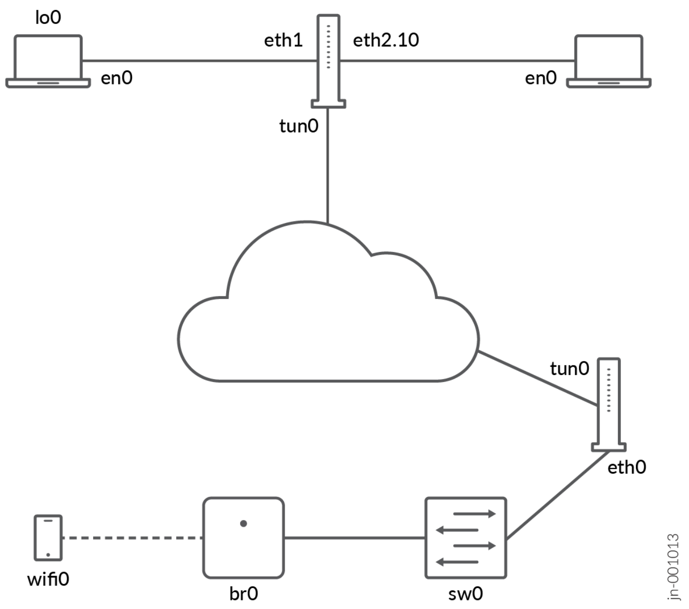
Overview

SUMMARY

Describes the various types of network interfaces and their characteristics.

A network interface—also known as network interface controller, network interface card, or network adapter—is a hardware component that is designed to enable computers to access an interconnection network for communication and synchronization purposes. An interface acts as a port where a host sends or receives packets. A network interface typically provides two distinct kinds of interfaces: one for the computer (host) side and another for the network side. The network interface translates the protocol of the host interface to the network protocol and vice versa, and converts between the different physical media.

Figure 1: Network Interfaces



Types of Interfaces

You can classify a network interfaces as follows:

- Physical Interfaces—A physical interface is a connection between a system and a network. It consists of a software driver and a connector to which you connect network media.

- Logical Interfaces—A logical interface abstracts a main interface and captures its physical characteristics. A VLAN uses logical interfaces to create a virtual network.

- Management Interfaces—Management interfaces are the primary interfaces for accessing the device remotely. You can use the management interface to access the device over the network using utilities such as `ssh` and `telnet`. You can configure the device from anywhere, regardless of its physical location. SNMP can use the management interface to gather statistics from the device.

- Loopback Interfaces—A loopback interface is a virtual, software-based interface on a network device that is always active and reachable. It's used to test and debug network connectivity, and to assign a permanent IP address to a device. The loopback interfaces provide a stable and reliable way to access and manage network devices.

- Serial Interfaces—A serial port is a serial communication interface through which information transfers in or out sequentially one bit at a time.

- Discard Interfaces—The discard interface is a virtual interface that silently discards packets as the packets arrive.

- IP Demultiplexing Interfaces—Demultiplexing (demux) interfaces are logical interfaces that share a common, underlying interface.

See the following table for more information:

Table 1: Switching Interface Basics

Topic	Link
Damping Interfaces	Damping Interfaces
Interface Ranges	Interface Ranges for Physical Interfaces
Loopback Interfaces	Loopback Interfaces
Serial Interfaces	Serial Interfaces
Discard Interfaces	Discard Interfaces
IP Demultiplexing Interfaces	IP Demultiplexing Interfaces

Device Interfaces Overview

IN THIS SECTION

- Device Interfaces Overview | 4
- Types of Interfaces | 5
- Interface Naming Overview | 6
- Interface Descriptors Overview | 24
- Physical Part of an Interface Name | 26
- Displaying Interface Configurations | 27
- Interface Encapsulations Overview | 28
- Understanding Transient Interfaces | 42
- Understanding Services Interfaces | 43
- Understanding Container Interfaces | 44
- Understanding Internal Ethernet Interfaces | 47
- Understanding Interfaces on ACX Series Universal Metro Routers | 49
- TX Matrix Plus and T1600 Router (Routing Matrix) Management Ethernet Interfaces | 53
- T1600 Routers (Routing Matrix) Internal Ethernet Interfaces | 53

The interfaces on a device provide network connectivity to the device. This topic discusses about the various device interfaces supported on Junos OS 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 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.

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

You can configure Junos OS class-of-service (CoS) properties to provide a variety of classes of service for different applications, including multiple forwarding classes for managing packet transmission, congestion management, and CoS-based forwarding.

For more information about configuring CoS properties, see the Junos OS Class of Service User Guide for Routing Devices.

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 47

- 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.
- Container interfaces—Interfaces that support automatic protection switching (APS) on physical SONET links using a virtual container infrastructure.

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

Interface Naming Overview

IN THIS SECTION

- Physical Part of an Interface Name | 7
- Logical Part of an Interface Name | 13
- Separators in an Interface Name | 13
- Channel Part of an Interface Name | 13
- Interface Naming for a Routing Matrix Based on a TX Matrix Router | 14
- Interface Naming for a Routing Matrix Based on a TX Matrix Plus Router | 17
- Chassis Interface Naming | 20
- Examples: Interface Naming | 21

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

The following sections provide interface naming configuration guidelines:

Physical Part of an Interface Name

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

NOTE: The internal 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
pfe-1/0/0               up    up
pfe-1/0/0.16383         up    up   inet
                                   inet6
pfh-1/0/0               up    up
pfh-1/0/0.16383         up    up   inet
[..........]
bcm0                    up    up <----------------
bcm0.0                  up    up   inet 10.0.0.1/8
[..........]
lsi                     up    up
mtun                    up    up
pimd                    up    up
pime                    up    up
tap                     up    up
```

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

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; for more information, see Aggregated Ethernet Interfaces Overview.

- as—Aggregated SONET/SDH interface. This is a virtual aggregated link and has a different naming format from most PICs; for more information, see Configuring Aggregated SONET/SDH Interfaces.

- `at`—ATM1 or ATM2 intelligent queuing (IQ) interface or a virtual ATM interface on a circuit emulation (CE) interface.
- `bcm`—The bcm0 internal Ethernet process is supported on specific Routing Engines for various M series and T series routers. For more information, refer to the link titled *Supported Routing Engines by Chassis* under Related Documentation.
- `cau4`—Channelized AU-4 IQ interface (configured on the Channelized STM1 IQ or IQE PIC or Channelized OC12 IQ and IQE PICs).
- `ce1`—Channelized E1 IQ interface (configured on the Channelized E1 IQ PIC or Channelized STM1 IQ or IQE PIC).
- `ci`—Container interface.
- `coc1`—Channelized OC1 IQ interface (configured on the Channelized OC12 IQ and IQE or Channelized OC3 IQ and IQE PICs).
- `coc3`—Channelized OC3 IQ interface (configured on the Channelized OC3 IQ and IQE PICs).
- `coc12`—Channelized OC12 IQ interface (configured on the Channelized OC12 IQ and IQE PICs).
- `coc48`—Channelized OC48 interface (configured on the Channelized OC48 and Channelized OC48 IQE PICs).
- `cp`—Collector interface (configured on the Monitoring Services II PIC).
- `cstm1`—Channelized STM1 IQ interface (configured on the Channelized STM1 IQ or IQE PIC).
- `cstm4`—Channelized STM4 IQ interface (configured on the Channelized OC12 IQ and IQE PICs).
- `cstm16`—Channelized STM16 IQ interface (configured on the Channelized OC48/STM16 and Channelized OC48/STM16 IQE PICs).
- `ct1`—Channelized T1 IQ interface (configured on the Channelized DS3 IQ and IQE PICs, Channelized OC3 IQ and IQE PICs, Channelized OC12 IQ and IQE PICs, or Channelized T1 IQ PIC).
- `ct3`—Channelized T3 IQ interface (configured on the Channelized DS3 IQ and IQE PICs, Channelized OC3 IQ and IQE PICs, or Channelized OC12 IQ and IQE PICs).
- `demux`—Interface that supports logical IP interfaces that use the IP source or destination address to demultiplex received packets. Only one demux interface (`demux0`) exists per chassis. All demux logical interfaces must be associated with an underlying *logical interface*.
- `dfc`—Interface that supports dynamic flow capture processing on T Series or M320 routers containing one or more Monitoring Services III PICs. Dynamic flow capture enables you to capture packet flows on the basis of dynamic filtering criteria. Specifically, you can use this feature to forward passively

monitored packet flows that match a particular filter list to one or more destinations using an on-demand control protocol.

- `ds`—DS0 interface (configured on the Multichannel DS3 PIC, Channelized E1 PIC, Channelized OC3 IQ and IQE PICs, Channelized OC12 IQ and IQE PICs, Channelized DS3 IQ and IQE PICs, Channelized E1 IQ PIC, Channelized STM1 IQ or IQE PIC, or Channelized T1 IQ).
- `dsc`—Discard interface.
- `e1`—E1 interface (including channelized STM1-to-E1 interfaces).
- `e3`—E3 interface (including E3 IQ interfaces).
- `em`—Management and internal Ethernet interfaces. For M Series routers, MX Series routers, T Series routers, and TX Series routers, you can use the `show chassis hardware` command to display hardware information about the router, including its Routing Engine model. To determine which management interface is supported on your router and Routing Engine combination, see *Understanding Management Ethernet Interfaces* and Supported Routing Engines by Router.
- `es`—Encryption interface.
- `et`—Ethernet interfaces (10-, 25-, 40-, 50-, 100-, 200-, and 400-Gigabit Ethernet interface).
- `fe`—Fast Ethernet interface.
- `fxp`—Management and internal Ethernet interfaces. For M Series routers, MX Series routers, T Series routers, and TX Series routers, you can use the `show chassis hardware` command to display hardware information about the router, including its Routing Engine model. To determine which management interface is supported on your router and Routing Engine combination, see *Understanding Management Ethernet Interfaces* and Supported Routing Engines by Router.
- `ge`—Gigabit Ethernet interface.

NOTE:

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

  show chassis hardware

  ```
  user@host> show chassis hardware
  ..
  ```

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

show configuration interfaces

```
user@host> show configuration interfaces ge-4/0/0
unit 0 {
    family inet {
        address 100.0.0.1/24;
    }
}
```

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

- `gr`—Generic routing encapsulation (GRE) tunnel interface.

- `gre`—Internally generated interface that is configurable only as the control channel for Generalized MPLS (GMPLS). For more information about GMPLS, see the Junos OS MPLS Applications User Guide.

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

- `ip`—IP-over-IP encapsulation tunnel interface.

- `ipip`—Internally generated interface that is not configurable.

- `ixgbe`—The internal Ethernet process ixgbe0 and ixgbe1 are used by the RE-DUO-C2600-16G Routing Engine, which is supported on TX Matrix Plus and PTX5000.

- `iw`—Logical interfaces associated with the endpoints of Layer 2 circuit and Layer 2 VPN connections (pseudowire stitching Layer 2 VPNs). For more information about VPNs, see the Junos OS VPNs Library for Routing Devices.

- `lc`—Internally generated interface that is not configurable.

- `lo`—Loopback interface. The Junos OS automatically configures one loopback interface (`lo0`). The logical interface `lo0.16383` is a nonconfigurable interface for router control traffic.

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

- `stm1`—STM1 interface (configured on the OC3/STM1 interfaces).
- `stm4`—STM4 interface (configured on the OC12/STM4 interfaces).
- `stm16`—STM16 interface (configured on the OC48/STM16 interfaces).
- `t1`—T1 interface (including channelized DS3-to-DS1 interfaces).
- `t3`—T3 interface (including channelized OC12-to-DS3 interfaces).
- `tap`—Internally generated interface that is not configurable.
- `umd`—USB modem interface.
- `vsp`—Voice services interface.
- `vc4`—Virtually concatenated interface.
- `vt`—Virtual loopback tunnel interface.
- `vtep`—Virtual tunnel endpoint interface for VXLANS.
- `xe`—10-Gigabit Ethernet interface. Some older 10-Gigabit Ethernet interfaces use the `ge` media type (rather than `xe`) to identify the physical part of the network device.
- `xt`—Logical interface for Protected System Domains to establish a Layer 2 tunnel connection.

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

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

On PTX1000 routers, the FPC number is always 0.

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

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

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

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

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

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

For more information about interface naming for a routing matrix, see "Interface Naming for a Routing Matrix Based on a TX Matrix Router" on page 14.

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.

Channel Part of an Interface Name

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

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

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

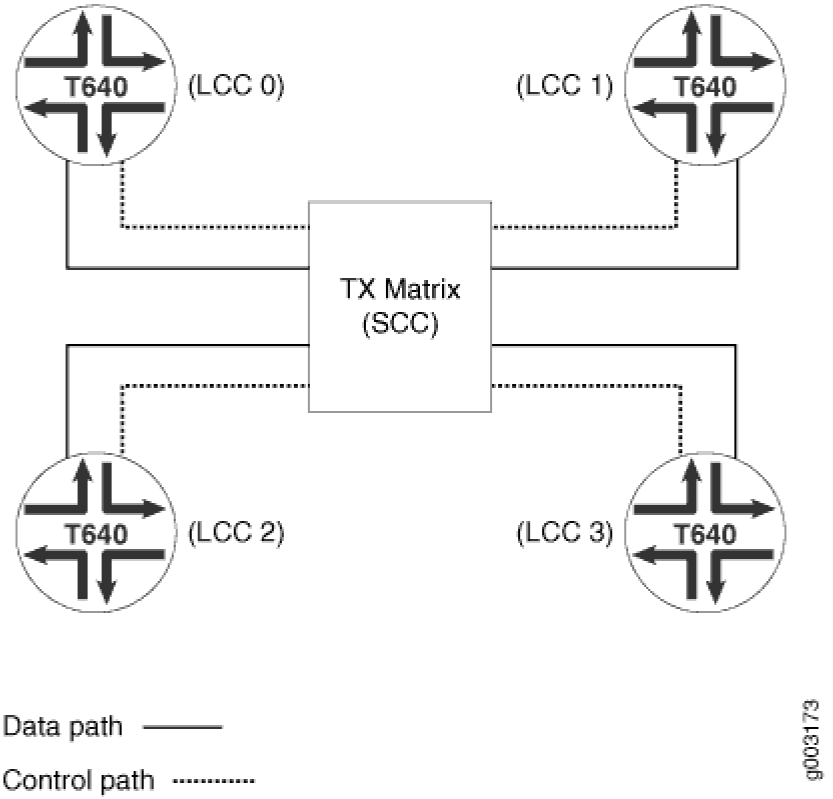
> **NOTE**: For ISDN, the B-channel and D-channel interfaces do not have any configurable parameters. However, when interface statistics are displayed, B-channel and D-channel interfaces have statistical values.

> **NOTE**: In the Junos OS implementation, the term *logical interfaces* generally refers to interfaces you configure by including the `unit` statement at the `[edit interfaces` *`interface-name`*`]` hierarchy level. Logical interfaces have the *`.logical`* descriptor at the end of the interface name, as in `ge-0/0/0.1` or `t1-0/0/0:0.1`, where the logical unit number is `1`.
>
> Although channelized interfaces are generally thought of as logical or virtual, the Junos OS sees T3, T1, and NxDS0 interfaces within a channelized IQ or IQE PIC as physical interfaces. For example, both `t3-0/0/0` and `t3-0/0/0:1` are treated as physical interfaces by the Junos OS. In contrast, `t3-0/0/0.2` and `t3-0/0/0:1.2` are considered logical interfaces because they have the `.2` at the end of the interface names.

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

A routing matrix based on a Juniper Networks TX Matrix router is a multichassis architecture composed of one TX Matrix router and from one to four interconnected T640 routers. From the perspective of the user interface, the routing matrix appears as a single router. The TX Matrix router controls all the T640 routers, as shown in Figure 2 on page 15.

Figure 2: Routing Matrix



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

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

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

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

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

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

- On LCC 3, FPC hardware slots 0 through 7 are configured as 24 through 31.

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

Table 2 on page 16 summarizes the FPC numbering for a T640 router in a routing matrix.

Table 2: FPC Numbering for T640 Routers in a Routing Matrix

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

Table 3 on page 16 lists each FPC hardware slot and the corresponding configuration numbers for LCCs 0 through 3.

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

FPC Numbering	T640 Routers							
	LCC 0							
Hardware Slots	0	1	2	3	4	5	6	7
Configuration Numbers	0	1	2	3	4	5	6	7
	LCC 1							
Hardware Slots	0	1	2	3	4	5	6	7

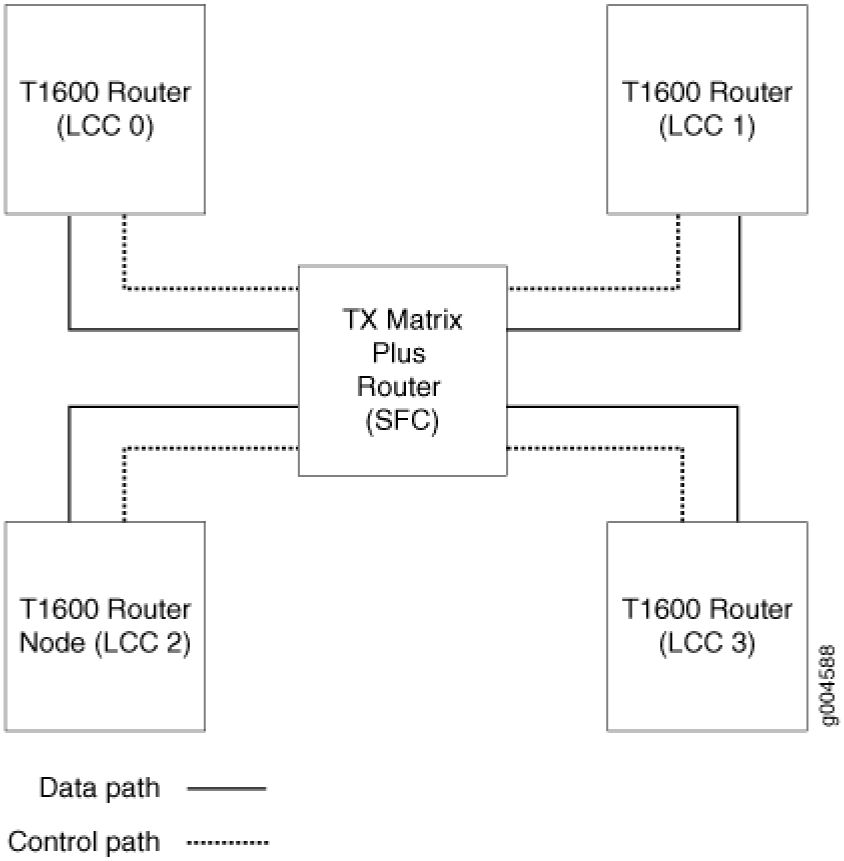
Table 3: One-to-One FPC Numbering for T640 Routers in a Routing Matrix *(Continued)*

FPC Numbering	T640 Routers							
Configuration Numbers	8	9	10	11	12	13	14	15
	LCC 2							
Hardware Slots	0	1	2	3	4	5	6	7
Configuration Numbers	16	17	18	19	20	21	22	23
	LCC 3							
Hardware Slots	0	1	2	3	4	5	6	7
Configuration Numbers	24	25	26	27	28	29	30	31

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

A routing matrix based on a Juniper Networks TX Matrix Plus Router is a multichassis architecture composed of one TX Matrix Plus router and from one to four interconnected T1600 routers. From the perspective of the user interface, the routing matrix appears as a single router. The TX Matrix Plus router controls all the T1600 routers, as shown in Figure 3 on page 18.

Figure 3: Routing Matrix Based on a TX Matrix Plus Router



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

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

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

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

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

- On LCC 0, FPC hardware slots 0 through 7 are configured as 0 through 7.
- On LCC 1, FPC hardware slots 0 through 7 are configured as 8 through 15.

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

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

Table 4 on page 19 summarizes the FPC numbering for a routing matrix based on a TX Matrix Plus router.

Table 4: FPC Numbering for T1600 Routers in a Routing Matrix

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

Table 5 on page 19 lists each FPC hardware slot and the corresponding configuration numbers for LCCs 0 through 3.

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

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

Table 5: One-to-One FPC Numbering for T1600 Routers in a Routing Matrix *(Continued)*

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

Chassis Interface Naming

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

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

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

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

  ```
  [edit chassis]
  lcc lcc-number {
      fpc slot-number { # Use the hardware FPC slot number
          pic pic-number {
              ...
          }
      }
  }
  ```

  For the FPC slot in a T640 router in a routing matrix, specify the actual hardware slot number, as labeled on the T640 router chassis. Do not use the corresponding software FPC configuration numbers shown in Table 3 on page 16.

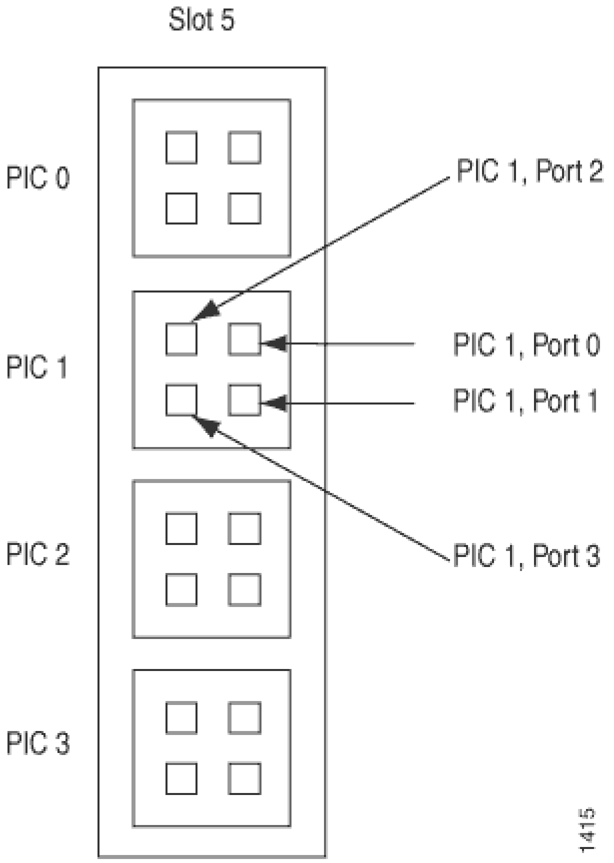
  For the FPC slot in a T1600 router in a routing matrix, specify the actual hardware slot number, as labeled on the T1600 router chassis. Do not use the corresponding software FPC configuration numbers shown in Table 4 on page 19.

For more information about the `[edit chassis]` hierarchy, see the Junos OS Administration Library for Routing Devices.

Examples: Interface Naming

This section provides examples of naming interfaces. For an illustration of where slots, PICs, and ports are located, see Figure 4 on page 22.

Figure 4: Interface Slot, PIC, and Port Locations



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

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

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

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

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

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

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

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

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

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

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

```
so-13/0/0.0
so-13/1/0.0
so-13/2/0.0
so-13/3/0.0
```

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

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

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

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

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)
- Multilink Frame Relay end-to-end (MLFR end-to-end)

- Multilink Frame Relay user-to-network interface network-to-network interface (MLFR UNI NNI)
- Multilink Point-to-Point Protocol (MLPPP)
- Multiprotocol Label Switching (MPLS)
- Trivial Network Protocol (TNP)
- (M Series, T Series, and MX Series routers only) Virtual private LAN service (VPLS)

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

You configure the various interface descriptors as follows:

- You configure the physical interface descriptor by including the `interfaces` *`interface-name`* statement.
- You configure the logical interface descriptor by including the `unit` statement within the `interfaces` *`interface-name`* statement or by including the `.`*`logical`* descriptor at the end of the interface name, as in `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.
- You configure tunnels by including the tunnel statement within the `unit` statement.

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

Physical Part of an Interface Name

IN THIS SECTION

- Interface Names for ACX Series, PTX Series, and QFX Series Devices | 26
- Interface Names for M Series and T Series Routers | 26
- Interface Names for MX Series Routers | 27

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.

NOTE: In the CLI, all PTX3000 PICs are represented as `pic0`. For more information, see PTX3000 PIC Description.

In the physical part of the interface name, a hyphen (-) separates the media type (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]
```

Interface Names for M Series and T Series Routers

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

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

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

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

Interface Names for MX Series Routers

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

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

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

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

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

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

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 6 on page 28 lists encapsulation support by interface type.

Table 6: Encapsulation Support by Interface Type

Interface Type	Physical Interface Encapsulation	*Logical Interface* Encapsulation
`ae`—Aggregated Ethernet interface	`ethernet-ccc`—Ethernet cross-connect `extended-vlan-ccc`—Nonstandard TPID tagging for a cross-connect `extended-vlan-vpls`—Extended VLAN virtual private LAN service `flexible-ethernet-services`—Allows per-unit Ethernet encapsulation configuration. `vlan-ccc`—802.1Q tagging for a cross-connect `ethernet-vpls`—Ethernet virtual private LAN service `vlan-vpls`—VLAN virtual private LAN service	`dix`—Ethernet DIXv2 (RFC 894) `vlan-ccc`—802.1Q tagging for a cross-connect
`as`—Aggregated SONET/SDH interface	`cisco-hdlc`—Cisco-compatible HDLC framing `ppp`—Serial PPP device	NA

Table 6: Encapsulation Support by Interface Type *(Continued)*

Interface Type	Physical Interface Encapsulation	*Logical Interface* Encapsulation
`at`—ATM1 interface	`atm-ccc-cell-relay`—ATM cell relay encapsulation for a cross-connect `atm-pvc`—ATM permanent virtual circuits `ethernet-over-atm`—Ethernet over ATM encapsulation	`atm-ccc-cell-relay`—ATM cell relay for CCC `atm-ccc-vc-mux`—ATM VC for CCC `atm-cisco-nlpid`—Cisco-compatible ATM NLPID encapsulation `atm-nlpid`—ATM NLPID encapsulation `atm-snap`—ATM LLC/SNAP encapsulation `atm-tcc-snap`—ATM LLC/SNAP for a translational cross-connect `atm-tcc-vc-mux`—ATM VC for a translational cross-connect `atm-vc-mux`—ATM VC multiplexing `ether-over-atm-llc`—Ethernet over ATM (LLC/SNAP) encapsulation

Table 6: Encapsulation Support by Interface Type *(Continued)*

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

Table 6: Encapsulation Support by Interface Type *(Continued)*

Interface Type	Physical Interface Encapsulation	*Logical Interface* Encapsulation
`ci`—Container interface	`cisco-hdlc`—Cisco-compatible HDLC framing `ppp`—Serial PPP device	`aps`—SONET interface required for APS configuration.

Table 6: Encapsulation Support by Interface Type *(Continued)*

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

Table 6: Encapsulation Support by Interface Type *(Continued)*

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

Table 6: Encapsulation Support by Interface Type *(Continued)*

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

Table 6: Encapsulation Support by Interface Type *(Continued)*

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

Table 6: Encapsulation Support by Interface Type *(Continued)*

Interface Type	Physical Interface Encapsulation	*Logical Interface* Encapsulation
Ethernet interfaces ge (including Gigabit Ethernet IQ interfaces, xe and 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
ixgbe—10-Gigabit Ethernet internal interfaces	NA	NA
lo—Loopback interface; the Junos OS automatically configures one loopback interface (lo0).	NA	NA
ls—Link services interface	multilink-frame-relay-uni-nni—Multilink Frame Relay UNI NNI (FRF.16) encapsulation	multilink-frame-relay-end-to-end—Multilink Frame Relay end-to-end (FRF.15) multilink-ppp—Multilink PPP

Table 6: Encapsulation Support by Interface Type *(Continued)*

Interface Type	Physical Interface Encapsulation	*Logical Interface* Encapsulation
lsq—Link services IQ interface	multilink-frame-relay-uni-nni—Multilink Frame Relay UNI NNI (FRF.16) encapsulation	multilink-frame-relay-end-to-end—Multilink Frame Relay end-to-end (FRF.15) multilink-ppp—Multilink PPP
lt—Logical tunnel interface	NA	ethernet—Ethernet service ethernet-vpls—Ethernet virtual private LAN service ethernet-ccc—Ethernet cross-connect frame-relay—Frame Relay encapsulation frame-relay-ccc—Frame Relay for a cross-connect vlan—VLAN service vlan-ccc—802.1Q tagging for a cross-connect vlan-vpls—VLAN virtual private LAN service
ml—Multilink interface (including Multilink Frame Relay and MLPPP)	NA	multilink-frame-relay-end-to-end—Multilink Frame Relay end-to-end (FRF.15) multilink-ppp—Multilink PPP

Table 6: Encapsulation Support by Interface Type *(Continued)*

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

Table 6: Encapsulation Support by Interface Type *(Continued)*

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

Table 6: Encapsulation Support by Interface Type *(Continued)*

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

Table 6: Encapsulation Support by Interface Type *(Continued)*

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

Table 6: Encapsulation Support by Interface Type *(Continued)*

Interface Type	Physical Interface Encapsulation	*Logical Interface* Encapsulation
Services interfaces (`cp`, `gr`, `ip`, `mo`, `vt`, `es`, `mo`, `rsp`, and `sp`)	NA	NA
Unconfigurable, internally generated interfaces (`gre`, `ipip`, `learning-chip (lc)`, `lsi`, `tap`, `mt`, `mtun`, `pd`, `pe`, `pimd`, and `pime`)	NA	NA

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

Understanding Transient Interfaces

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

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

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

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

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

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

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

Understanding Services Interfaces

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

- Adaptive Services (AS) PICs—Enable you to provide multiple services on a single PIC by configuring a set of services and applications. The AS PICs offer a special range of services that you configure in one or more service sets.

- ES PIC—Provides a security suite for the IP version 4 (IPv4) and IP version 6 (IPv6) network layers. The suite provides functionality such as authentication of origin, data integrity, confidentiality, replay protection, and nonrepudiation of source. It also defines mechanisms for key generation and exchange, management of security associations, and support for digital certificates.

- Monitoring Services PICs—Enable you to monitor traffic flow and export the monitored traffic. Monitoring traffic enables you to gather and export detailed information about IPv4 traffic flows between source and destination nodes in your network; sample all incoming IPv4 traffic on the monitoring interface and present the data in cflowd record format; perform discard accounting on an incoming traffic flow; encrypt or tunnel outgoing cflowd records, intercepted IPv4 traffic, or both; and direct filtered traffic to different packet analyzers and present the data in its original format. On a Monitoring Services II PIC, you can configure either monitoring interfaces or collector interfaces. A collector interface enables you to combine multiple cflowd records into a compressed ASCII data file and export the file to an FTP server.

- Multilink Services, MultiServices, Link Services, and Voice Services PICs—Enable you to split, recombine, and sequence datagrams across multiple logical data links. The goal of multilink operation is to coordinate multiple independent links between a fixed pair of systems, providing a virtual link with greater bandwidth than any of the members.

- Tunnel Services PIC—By encapsulating arbitrary packets inside a transport protocol, tunneling provides a private, secure path through an otherwise public network. Tunnels connect discontinuous subnetworks and enable encryption interfaces, virtual private networks (VPNs), and Multiprotocol Label Switching (MPLS).

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

Understanding Container Interfaces

IN THIS SECTION

- Understanding Traditional APS Concept | **45**
- Container Interfaces Concept | **45**
- APS Support for Container-Based Interfaces | **46**
- Autocopy of APS Parameters | **46**

Container interfaces provide the following features:

- Automatic protection switching (APS) on SONET/SDH and ATM links are supported using the container infrastructure.

- Container physical interfaces and logical interfaces remain up on switchover.

- APS parameters are auto-copied from the container interface to the member links.

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

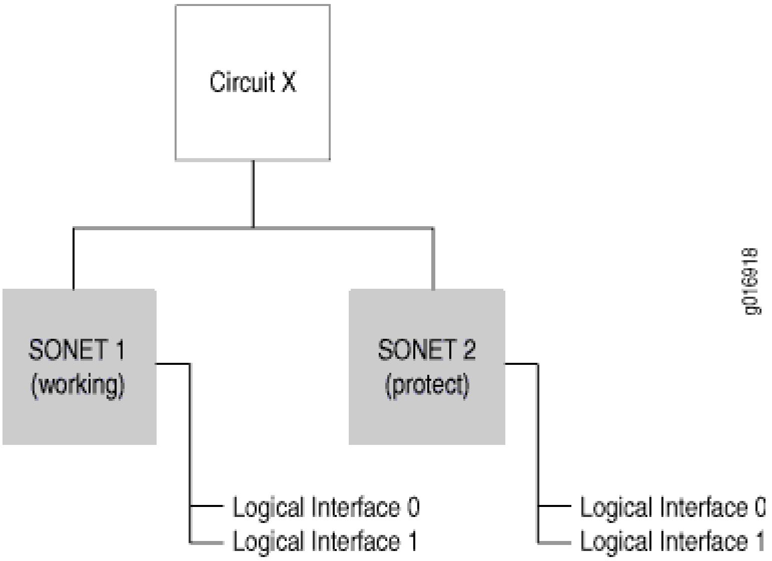
For more information on SONET/SDH configuration, see Configuring Container Interfaces for APS on SONET Links.

Container interfaces features are described in the following sections:

Understanding Traditional APS Concept

Traditional Automatic Protection Switching (APS) is configured on two independent physical SONET/SDH interfaces: one interface is configured as the working circuit and the other is configured as the protect circuit (see Figure 5 on page 45). The circuit, named Circuit X in the figure, is the link between the two SONET interfaces.

Figure 5: APS Interface

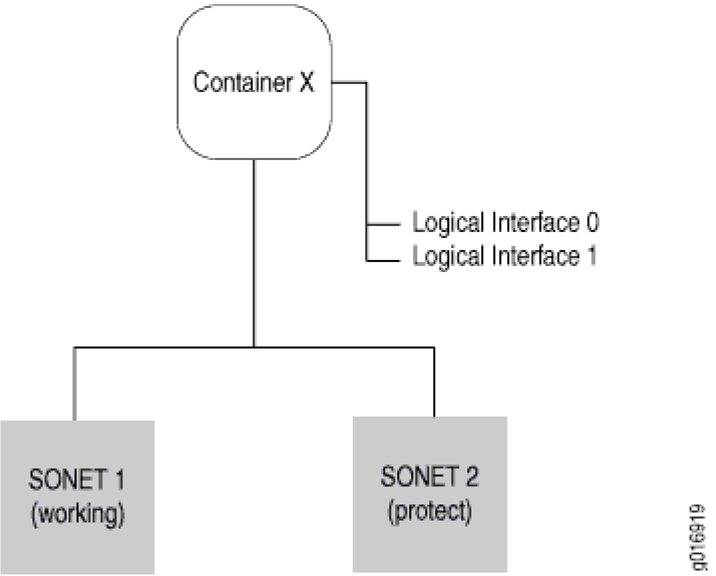


Traditional APS uses routing protocols that run on each individual SONET/SDH interface (since circuit is an abstract construct, instead of being an actual interface). When the working link goes down, the APS infrastructure brings up the protect link and its underlying logical interfaces and brings down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

Container Interfaces Concept

To solve the problem of traffic loss, the Junos OS provides a soft interface construct called a container interface (see Figure 6 on page 46).

Figure 6: Container Interface



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

APS Support for Container-Based Interfaces

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

Autocopy of APS Parameters

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

SEE ALSO

Configuring Container Interfaces for APS on SONET Links

Displaying APS Using a Container Interface with ATM Encapsulation

Understanding Internal Ethernet Interfaces

Within a Juniper device, internal Ethernet interfaces provide communication between the Routing Engine and the Packet Forwarding Engines. Junos OS automatically configures internal Ethernet interfaces when Junos OS boots. Junos OS 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.

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

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

- Most Juniper devices—Junos OS 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.

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

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

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

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

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

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

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

  - The alternate link to the remote Routing Engine is the 10-Gigabit Ethernet port at the `ixgbe1` interface. This second port connects the Routing Engine to the 10-Gigabit Ethernet switch on the remote Control Board, which connects to the 10-Gigabit Ethernet port at the `ixgbe0` interface on the remote Routing Engine.

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

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

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

  - The Routing Engine connects to a Gigabit Ethernet switch on the local Control Board.

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

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

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

  - The alternate link to the remote Routing Engine is at the `em1` interface. This second port connects the Routing Engine to the Gigabit Ethernet switch on the remote Control Board, which connects to the Gigabit Ethernet port at the `bcm0` interface on the remote Routing Engine.

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

Each 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.

SEE ALSO

Supported Routing Engines by Router

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

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

Understanding Interfaces on ACX Series Universal Metro Routers

IN THIS SECTION

- T1 and E1 Time-Division Multiplexing (TDM) Interfaces | **50**
- Inverse Multiplexing for ATM (IMA) | **51**
- Gigabit Ethernet Interfaces | **51**

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

The ACX Series routers support the following:

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

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

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

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

NOTE: Only ACX5048, ACX5096, and ACX5448 routers support 40GbE. The ACX5448 router supports 40GbE channeling to 10GbE.

T1 and E1 Time-Division Multiplexing (TDM) Interfaces

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

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

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

T1 or E1 BITS Interface (ACX2000)

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

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

Inverse Multiplexing for ATM (IMA)

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

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

Gigabit Ethernet Interfaces

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

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

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

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

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

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

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

SEE ALSO

Understanding Encapsulation on an Interface

Configuring Inverse Multiplexing for ATM (IMA) on ACX Series

Device Interfaces Overview | 4

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

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

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

```
user@host> show interfaces ?
...
em0
em0.0
...
```

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

SEE ALSO

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

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

T1600 Routers (Routing Matrix) Internal Ethernet Interfaces

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

SEE ALSO

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

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

Physical Interface Properties

IN THIS SECTION

- Physical Interface Properties Overview | **55**
- Configure the Interface Description | **55**
- How to Specify an Aggregated Interface | **57**
- Configure the Link Characteristics | **57**
- Interface Speed | **58**
- Forward Error Correction (FEC) | **65**
- Interface Aliases | **66**
- Example: Add an Interface Alias Name | **68**
- Clock Source Overview | **73**
- Configure the Clock Source | **74**
- Interface Encapsulation on Physical Interfaces | **75**
- Keepalives | **82**
- Understanding Unidirectional Traffic Flow on Physical Interfaces | **85**
- Enable Unidirectional Traffic Flow on Physical Interfaces | **86**
- Enable SNMP Notifications on Physical Interfaces | **86**
- Accounting for Physical Interfaces | **87**
- Disable a Physical Interface | **91**
- Configure Ethernet Loopback Capability | **93**
- Configure Short Reach Mode on QFX5100-48T | **93**
- Configure Flow Control | **95**
- Set the Mode on an SFP+ or SFP+ MACsec Uplink Module | **96**

- Set the Operational Mode on a 2-Port 40-Gigabit Ethernet QSFP+/100-Gigabit Ethernet QSFP28 Uplink Module | **97**
- Configure the Media Type on Dual-Purpose Uplink Ports | **98**

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, link operational mode, and clock source.

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

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. See *Using DHCP Relay Agent Option 82 Information*.

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`.

```
user-server> snmpwalk host-fxp0.mylab public ifXTable | grep -e '\.23'
 snmpwalk host-fxp0.mylab public ifXTable | grep -e '\.23'
ifMIB.ifMIBObjects.ifXTable.ifXEntry.ifName.23 = et-1/0/1
ifMIB.ifMIBObjects.ifXTable.ifXEntry.ifInMulticastPkts.23 = Counter32: 0
ifMIB.ifMIBObjects.ifXTable.ifXEntry.ifInBroadcastPkts.23 = Counter32: 0
ifMIB.ifMIBObjects.ifXTable.ifXEntry.ifOutMulticastPkts.23 = Counter32: 0
ifMIB.ifMIBObjects.ifXTable.ifXEntry.ifOutBroadcastPkts.23 = Counter32: 0
ifMIB.ifMIBObjects.ifXTable.ifXEntry.ifHCInOctets.23 = Counter64: 0
ifMIB.ifMIBObjects.ifXTable.ifXEntry.ifHCInUcastPkts.23 = Counter64: 0
ifMIB.ifMIBObjects.ifXTable.ifXEntry.ifHCInMulticastPkts.23 = Counter64: 0
ifMIB.ifMIBObjects.ifXTable.ifXEntry.ifHCInBroadcastPkts.23 = Counter64: 0
ifMIB.ifMIBObjects.ifXTable.ifXEntry.ifHCOutOctets.23 = Counter64: 42
ifMIB.ifMIBObjects.ifXTable.ifXEntry.ifHCOutUcastPkts.23 = Counter64: 0
ifMIB.ifMIBObjects.ifXTable.ifXEntry.ifHCOutMulticastPkts.23 = Counter64: 0
ifMIB.ifMIBObjects.ifXTable.ifXEntry.ifHCOutBroadcastPkts.23 = Counter64: 0
ifMIB.ifMIBObjects.ifXTable.ifXEntry.ifLinkUpDownTrapEnable.23 = enabled(1)
ifMIB.ifMIBObjects.ifXTable.ifXEntry.ifHighSpeed.23 = Gauge32: 100
ifMIB.ifMIBObjects.ifXTable.ifXEntry.ifPromiscuousMode.23 = false(2)
ifMIB.ifMIBObjects.ifXTable.ifXEntry.ifConnectorPresent.23 = true(1)
ifMIB.ifMIBObjects.ifXTable.ifXEntry.ifAlias.23 = Backbone connection to PHL01
ifMIB.ifMIBObjects.ifXTable.ifXEntry.ifCounterDiscontinuityTime.23 = Timeticks: (0) 0:00:00.00
```

For information about describing logical units, see "Adding a Logical Unit Description to the Configuration" on page 101.

How to Specify an Aggregated Interface

An aggregated interface is a group of interfaces. To specify an aggregated Ethernet interface, configure `aex` 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 aex]` hierarchy level to complete the association.

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

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

Configure the Link Characteristics

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

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

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

Keep the following in mind:

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

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

Interface Speed

IN THIS SECTION

- Configuring the Interface Speed on Ethernet Interfaces | 58
- Configure the Aggregated Ethernet Link Speed | 60
- Configure the SONET/SDH Interface Speed | 63

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

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

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

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

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

NOTE:

-
- 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.,
-
- 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 the Aggregated Ethernet Link Speed

IN THIS SECTION

- Platform-Specific LAG Behavior | 61

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 (AE) 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)

Use Feature Explorer to confirm platform and release support for specific features.

Review the Platform-Specific LAG Behavior section for notes related to your platform.

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:

Platform-Specific LAG Behavior

Platform	Difference
ACX Series	• ACX7000 Series routers that support LAG can operate in mixed mode. Configure the following on the same aggregated Ethernet interface: • Member links of different modes (WAN and LAN) with the same speed • Member links of different modes (WAN and LAN) with different speeds • ACX7000 Series routers support two modes of LAG configuration: • Maximum AE children 16 - 256 AE bundles • Maximum AE children 64 - 64 AE bundles • ACX7000 Series routers use `ether-options` instead of `gigether-options`.

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.

You can configure Aggregated Ethernet interfaces on the M120 router to operate at one of the following speeds:

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

You can configure aggregated Ethernet links on EX Series switches to operate at one of the following speeds:

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

- `50g`—Links are 50 Gbps.

You can configure aggregated Ethernet links on MX Series, and PTX Series routers and on QFX5100, QFX5120, QFX10002, QFX10008, and QFX10016 switches to operate at one of the following speeds:

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

Configure the SONET/SDH Interface Speed

You can configure the speed on SONET/SDH interfaces in concetenated, nonconcatenated, or channelized (multiplexed) mode.

To configure the SONET/SDH interface speed in concatenated mode:

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

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

2. Configure the interface speed in concatenated mode.

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

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

To configure the SONET/SDH interface speed in nonconcatenated mode:

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

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

2. Configure the interface speed in nonconcatenated mode.

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

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

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

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

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

2. Configure the `no-concatenate` option.

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

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

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 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 | 65
- Overview | 65
- Configure FEC | 66

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 enables or disables FEC based on the plugged-in optics. For instance, Junos OS enables RS-FEC for 100 Gigabit (Gb) SR4 optics and disables RS-FEC for 100 G LR4 optics. You can override the default behavior and explicitly enable or disable RS-FEC.

You can enable or disable RS-FEC for 100-Gigabit Ethernet (GbE) interfaces. After you enable or disable RS-FEC using this statement, this behavior applies to any 100GbE 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 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: PTX5000 routers with FPC-PTX-P1-A and FPC2-PTX-P1A do not support RS-FEC.

On PTX3000 and PTX5000 routers, FPC3-SFF-PTX-1H and FP3-SFF-PTX-1T with PE-10-U-QSFP28 PIC and LR4 optics support RS-FEC only on port 2. For PE-10-U-QSFP28 with LR4 optics, RS-FEC is the default FEC mode on port 2 and NONE is the default FEC mode on ports 0, 1, and 3 through 9. For PE-10-U-QSFP28 with SR4 optics, RS-FEC is enabled by default on all ports. Do not modify the FEC mode on any port, irrespective of the optics installed.

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 gigether-options fec none
   ```

2. To enable an FEC mode:

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

   Alternatively:

   ```
   [edit]
   user@device# delete interfaces interface-name gigether-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 | 67

- Configuration | 67

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.

To suppress the alias in favor of the interface name, use the `display no-interface-alias` parameter along with the show command.

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.

Using the exact value of the interface name for system operations and computations enables backward compatibility with Junos OS releases in which the support for interface aliases is not available.

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
```

On some devices, you can also configure the alias at the `[edit logical-systems` *`logical-system-name`* `interfaces` *`interface-name`* `unit` *`logical-unit-number`*`]` hierarchy level.

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.

Example: Add an Interface Alias Name

IN THIS SECTION

- Requirements | 69
- Overview | 69
- Configuration | 69
- Verification | 72

This example shows how to add an alias to the logical unit of an interface. Using an alias to identify interfaces as they appear in the output for operational commands can allow for more meaningful naming conventions and easier identification. This capability to define interface alias names for physical and logical interfaces is useful in a Junos Node Unifier (JNU) environment that contains the following devices:

- A Juniper Networks MX Series 5G Universal Routing Platform as a controller

- EX Series Ethernet switches, QFX Series devices, and ACX Series Universal Metro Routers as satellite devices

Requirements

This example uses the following hardware and software components:

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

Overview

You can create an alias for each logical unit on a physical interface. The descriptive text you define for the alias is displayed in the output of the `show interfaces` commands. The alias configured for a logical unit of an interface has no effect on how the interface on the router or switch operates—it is only a cosmetic label.

Configuration

IN THIS SECTION

- CLI Quick Configuration | 69
- Add an Interface Alias Name for the Controller Interfaces | 70
- Results | 71

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

CLI Quick Configuration

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

```
set interfaces ae0 unit 0 alias "controller-sat1-downlink1"
set interfaces ae0.0 family inet address 10.0.0.1/24
set interfaces ae1 unit 0 alias "controller-sat1-downlink1"
set interfaces ae0.0 family inet address 192.0.2.128/25
set interfaces ge-0/0/0 vlan-tagging
set interfaces ge-0/0/0 unit 0 alias "ge-to-corp-gw1"
set interfaces ge-0/0/0.0 vlan-id 101
set interfaces ge-0/0/0.0 family inet address 10.1.1.1/23
set interfaces ge-0/1/0 gigether-options 802.3ad ae0
set interfaces ge-0/1/1 gigether-options 802.3ad ae0
set protocols rip group corporate-firewall neighbor ge-to-corp-gw1
```

Add an Interface Alias Name for the Controller Interfaces

Step-by-Step Procedure

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

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

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

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

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

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

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

   ```
   [edit]
   user@host# set interfaces ge-0/0/0 vlan-tagging
   ```

```
user@host# set interfaces ge-0/0/0 unit 0 alias "ge-to-corp-gw1"
 user@host# set interfaces ge-0/0/0.0 vlan-id 101
 user@host# set interfaces ge-0/0/0.0 family inet address 10.1.1.1/23
```

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

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

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

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

Results

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

```
[edit]
        interfaces {
            ae0 {
                unit 0 {
                    alias "controller-sat1-downlink1";
                    family inet {
                        address 10.0.0.1/24;
                    }
                }
                unit 1 {
                    alias "controller-sat1-downlink2";
                    family inet {
                        address 10.0.0.3/24;
                    }
                }
            }
            ge-0/0/0 {
                vlan-tagging;
                unit 0 {
```

```
            alias "ge-to-corp-gw1";
            vlan-id 101;
            family inet {
                address 10.1.1.1/23;
            }
        }
    }
    ge-0/1/0 {
gigether-options {
            802.3ad ae0;
        }
    }
    ge-0/1/1 {
gigether-options {
            802.3ad ae0;
        }
    }
}
protocols rip {
    group corporate-firewall {
        neighbor ge-to-corp-gw1;
    }
}
```

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

Verification

IN THIS SECTION

- Verify the Configuration of the Alias Name for the Controller Interfaces | 73

Use the examples in this section to verify that the alias name is displayed instead of the interface name.

Verify the Configuration of the Alias Name for the Controller Interfaces

Purpose

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

Action

Display information about all RIP neighbors.

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

Meaning

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

SEE ALSO

> *alias*

Clock Source Overview

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

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

The clock source resides on the Control Board (CB) for M120 routers. M7i and M10i routers have a clock source on the Compact Forwarding Engine Board (CFEB) and Enhanced Compact Forwarding Engine Board (CFEB-E).

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

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

Configure the Clock Source

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

To set the clock source as external or internal:

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

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

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

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

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

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

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

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

SEE ALSO

Configuring an External Synchronization Interface

clocking

Interface Encapsulation on Physical Interfaces

IN THIS SECTION

- Encapsulation Capabilities | 75
- Encapsulation Types | 76
- Configure Encapsulation on a Physical Interface | 78
- Display the Encapsulation on a Physical SONET/SDH Interface | 79
- Configure Interface Encapsulation on PTX Series Routers | 80

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 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 encapsulation type `vlan-vpls`, VLAN IDs 1 through 511 are reserved for normal VLANs, and VLAN IDs 512 through 4094 are reserved for VPLS VLANs. For 4-port Fast Ethernet interfaces, you can use VLAN IDs 512 through 1024 for VPLS VLANs.
- For encapsulation types `extended-vlan-ccc` and `extended-vlan-vpls`, all VLAN IDs are valid.
- For Gigabit Ethernet interfaces and Gigabit Ethernet IQ and IQE PICs with SFPs, 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.

> **NOTE**: The 10-port Gigabit Ethernet PIC and the built-in Gigabit Ethernet port on the M7i router do not support flexible Ethernet services encapsulation.

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

Encapsulation Types

The physical interface encapsulation types include:

- ATM CCC cell relay—Connects two remote virtual circuits or ATM physical interfaces with a label-switched path (LSP). Traffic on the circuit is ATM cells.

- ATM PVC—Defined in RFC 2684, *Multiprotocol Encapsulation over ATM Adaptation Layer 5*. When you configure physical ATM interfaces with ATM PVC encapsulation, an RFC 2684-compliant ATM Adaptation Layer 5 (AAL5) tunnel is set up to route the ATM cells over a Multiprotocol Label Switching (MPLS) path that is typically established between two MPLS-capable routers using the Label Distribution Protocol (LDP).

- Cisco-compatible High-Level Data Link Control (HDLC) framing (`cisco-hdlc`)—E1, E3, SONET/SDH, T1, and T3 interfaces can use Cisco HDLC encapsulation. Two related versions are supported:

  - CCC version (`cisco-hdlc-ccc`)—The logical interface does not require an encapsulation statement. When you use this encapsulation type, you can configure the `ccc` family only.

  - TCC version (`cisco-hdlc-tcc`)—Similar to CCC and has the same configuration restrictions, but used for circuits with different media on either side of the connection.

- Ethernet cross-connect—Ethernet interfaces without VLAN tagging can use Ethernet CCC encapsulation. Two related versions are supported:

  - CCC version (`ethernet-ccc`)—Ethernet interfaces with standard Tag Protocol ID (TPID) tagging can use Ethernet CCC encapsulation. When you use this encapsulation type, you can configure the `ccc` family only.

  - TCC version (`ethernet-tcc`)—Similar to CCC, but used for circuits with different media on either side of the connection.

    For 8-port, 12-port, and 48-port Fast Ethernet PICs, TCC is not supported.

- VLAN CCC (`vlan-ccc`)—Ethernet interfaces with VLAN tagging enabled can use VLAN CCC encapsulation. VLAN CCC encapsulation supports TPID 0x8100 only. When you use this encapsulation type, you can configure the `ccc` family only.

  When you configure Ethernet VLAN encapsulation on CCC circuits by using the `encapsulation vlan-ccc` statement at the `[edit interfaces` *`interface-name`*`]` hierarchy level, you can bind a list of VLAN IDs to the interface. To configure a CCC for multiple VLANs, use the `vlan-id-list [` *`vlan-id-numbers`* `]` statement. Configuring this statement creates a CCC for:

  - Each VLAN listed—for example, `vlan-id-list [ 100 200 300 ]`

  - Each VLAN in a range—for example, `vlan-id-list [ 100-200 ]`

  - Each VLAN in a list and range combination—for example, `vlan-id-list [ 50, 100-200, 300 ]`

- Extended VLAN cross-connect—Gigabit Ethernet interfaces with VLAN 802.1Q tagging enabled can use extended VLAN cross-connect encapsulation. (Ethernet interfaces with standard TPID tagging can use VLAN CCC encapsulation.) Two related versions of extended VLAN cross-connect are supported:

  - CCC version (`extended-vlan-ccc`)—Extended VLAN CCC encapsulation supports TPIDs 0x8100, 0x9100, and 0x9901. When you use this encapsulation type, you can configure the `ccc` family only.

  - TCC version (`extended-vlan-tcc`)—Similar to CCC, but used for circuits with different media on either side of the connection.

    For 8-port, 12-port, and 48-port Fast Ethernet PICs, extended VLAN CCC is not supported. For 4-port Gigabit Ethernet PICs, extended VLAN CCC and extended VLAN TCC are not supported.

- Ethernet VPLS (`ethernet-vpls`)—Ethernet interfaces with VPLS enabled can use Ethernet VPLS encapsulation.

- Ethernet VLAN VPLS (`vlan-vpls`)—Ethernet interfaces with VLAN tagging and VPLS enabled can use Ethernet VLAN VPLS encapsulation.

- Extended VLAN VPLS (`extended-vlan-vpls`)—Ethernet interfaces with VLAN 802.1Q tagging and VPLS enabled can use Ethernet Extended VLAN VPLS encapsulation. (Ethernet interfaces with standard TPID tagging can use Ethernet VLAN VPLS encapsulation.) Extended Ethernet VLAN VPLS encapsulation supports TPIDs 0x8100, 0x9100, and 0x9901.

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

- PPP—Defined in RFC 1661, *The Point-to-Point Protocol (PPP) for the Transmission of Multiprotocol Datagrams over Point-to-Point Links*. PPP is the default encapsulation type for physical interfaces. E1, E3, SONET/SDH, T1, and T3 interfaces can use PPP 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
```

NOTE:

- When the encapsulation type is set to `Cisco-compatible Frame Relay` encapsulation, ensure that the LMI type is set to ANSI or Q933-A.
- When `vlan-vpls` encapsulation is set at the physical interface level, commit check will validate that there should not be any `inet` family configured within it.

Display the Encapsulation on a Physical SONET/SDH Interface

IN THIS SECTION

- Purpose | 79
- Action | 80
- Meaning | 80

Purpose

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

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

Action

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

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

Meaning

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

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`

In Junos OS Evolved, the `flexible-ethernet-services` encapsulation is not supported on PTX10003 devices.

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;
                }
            }
        }
    }
}
```

Keepalives

By default, physical interfaces configured with Cisco High-Level Data Link Control (HDLC) or Point-to-Point Protocol (PPP) encapsulation send keepalive packets at 10-second intervals. The Frame Relay term for keepalives is Local Management Interface (LMI) packets; the Junos OS supports both ANSI T1.617 Annex D LMIs and International Telecommunication Union (ITU) Q933 Annex A LMIs. On Asynchronous Transfer Mode (ATM) networks, Operation, Administration, and Maintenance (OAM) cells perform the same function. You configure OAM cells at the logical interface level; for more information, see Defining the ATM OAM F5 Loopback Cell Period.

To disable the sending of keepalives:

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

To explicitly enable the sending of keepalives:

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

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

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

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

To change one or more of the default keepalive values:

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

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

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

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

On interfaces configured with Cisco HDLC or PPP encapsulation, you can include the following three keepalive statements. Note that these statements do not affect Frame Relay encapsulation:

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

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

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

On MX Series routers with Modular Port Concentrators/Modular Interface Cards (MPCs/MICs), the Packet Forwarding Engine on an MPC/MIC processes and responds to Link Control Protocol (LCP) Echo-Request keepalive packets that the PPP subscriber (client) initiates and sends to the router. The mechanism by which LCP Echo-Request packets are processed by the Packet Forwarding Engine instead of by the Routing Engine is referred to as *PPP fast keepalive* For more information about how PPP fast keepalive works on an MX Series router with MPCs/MICs, see the *Junos OS Subscriber Access Configuration Guide*.

SEE ALSO

Defining the ATM OAM F5 Loopback Cell Period

Disabling the Sending of PPPoE Keepalive Messages

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

Configuring Frame Relay Keepalives

Configuring PPP over ATM2 Encapsulation Overview

keepalives

no-keepalives

Understanding Unidirectional Traffic Flow on Physical Interfaces

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

Benefits

- Unidirectional interfaces enable the configuration of a unidirectional link topology. Unidirectional links are useful for applications such as broadband video services where almost all traffic flow is in one direction, from the provider to the user.

- Unidirectional link mode conserves bandwidth by enabling it to be differentially dedicated to transmit and receive interfaces.

- Unidirectional link mode conserves ports for such applications because the transmit-only and receive-only interfaces act independently. Each can be connected to different routers. For example, this can reduce the total number of ports required.

NOTE: Use Feature Explorer to confirm platform and release support for the Unidirectional link mode feature.

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

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

Unidirectional interfaces support IP and IP version 6 (IPv6). Packet forwarding takes place by means of static routes and static Address Resolution Protocol (ARP) entries, which you can configure independently on both unidirectional interfaces.

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

SEE ALSO

https://apps.juniper.net/feature-explorer/feature/3454?fn=Unidirectional%20link%20support

Enable Unidirectional Traffic Flow on Physical Interfaces

Unidirectional link mode makes the traffic flow in only one direction. To enable unidirectional traffic flow 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 `unidirectional` option to create two new, unidirectional (transmit-only and receive-only) physical interfaces subordinate to the original parent interface.

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

Enable SNMP Notifications on Physical Interfaces

By default, Junos OS 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 | 88
- Configure an Accounting Profile for a Physical Interface | 88
- Display the Accounting Profile | 89

Devices running Junos OS 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.

For more information about how to configure an accounting data log file, see the *Configuring Accounting-Data Log Files*.

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. You must specify a `file` statement for the interface profile that has already been configured at the `[edit accounting-options]` hierarchy level. To configure which file to use, use the `file` statement.

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

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. 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. To configure the interval, use the `interval` statement:

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

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
```

Display the Accounting Profile

IN THIS SECTION

- Purpose | 89
- Action | 90
- Meaning | 90

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 | 91
- Example: Disable a Physical Interface | 92

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.

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 192.168.1.10/24;
    }
}
```

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 192.168.1.10/24;
    }
}
```

Configure Ethernet Loopback Capability

To place an interface in loopback mode, include the `loopback` statement:

```
loopback;
```

To return to the default—that is, to disable loopback mode—delete the `loopback` statement from the configuration:

```
[edit]
user@switch# delete interfaces interface-name ether-options loopback
```

To explicitly disable loopback mode, include the `no-loopback` statement:

```
no-loopback;
```

You can include the **loopback** and `no-loopback` statements at the following hierarchy levels:

- `[edit interfaces` *`interface-name`* `aggregated-ether-options]`
- `[edit interfaces` *`interface-name`* `ether-options]`

Configure Short Reach Mode on QFX5100-48T

You can enable short-reach mode for individual and a range of copper-based 10-Gigabit Ethernet interfaces using short cable lengths (less than 10m) on the QFX5100-48T switch. Short-reach mode reduces power consumption up to 5 W on these interfaces.

Use Feature Explorer to confirm platform and release support for specific features.

1. To enable short-reach mode on an individual interface, issue the following command:

   ```
   [edit chassis]
   user@switch# set fpc fpc-slot pic pic-slot port port-number short-reach-mode enable
   ```

   For example, to enable short-reach mode on port 0 on PIC 0, issue the following command:

   ```
   [edit chassis]
   user@switch# set fpc 0 pic 0 port 0 short-reach-mode enable
   ```

2. To enable short-reach mode on a range of interfaces, issue the following command:

   ```
   [edit chassis]
   user@switch# set fpc fpc-slot pic pic-slot port-range port-range-low port-range-high short-reach-mode enable
   ```

   For example, to enable short-reach mode on a range of interfaces between port 0 and port 47 on PIC 0, issue the following command:

   ```
   [edit chassis]
   user@switch# set fpc 0 pic 0 port-range 0 47 short-reach-mode enable
   ```

3. To disable short-reach mode on an individual interface, issue the following command:

   ```
   [edit chassis]
   user@switch# set fpc fpc-slot pic pic-slot port port-number short-reach-mode disable
   ```

   For example, to disable short-reach mode on port 0 on PIC 0, issue the following command:

   ```
   [edit chassis]
   user@switch# set fpc 0 pic 0 port 0 short-reach-mode disable
   ```

4. To disable short-reach mode on a range of interfaces, issue the following command:

   ```
   [edit chassis]
   user@switch# set fpc fpc-slot pic pic-slot port-range port-range-low port-range-high short-reach-mode disable
   ```

For example, to disable short-reach mode on a range of interfaces between port 0 and port 47 on PIC 0, issue the following command:

```
[edit chassis]
user@switch# set fpc 0 pic 0 port-range 0 47 short-reach-mode disable
```

Configure Flow Control

By default, the router or switch imposes flow control to regulate the amount of traffic sent out on a Fast Ethernet, Tri-Rate Ethernet copper, GbE, and 10-Gigabit Ethernet interface. Flow control is not supported on the 4-port Fast Ethernet PIC. This is useful if the remote side of the connection is a Fast Ethernet or GbE switch.

You can disable flow control if you want the router or switch to permit unrestricted traffic. To disable flow control, include the `no-flow-control` statement:

```
no-flow-control;
```

To explicitly reinstate flow control, include the `flow-control` statement:

```
flow-control;
```

You can include these statements at the following hierarchy levels:

- `[edit interfaces` *`interface-name`* `aggregated-ether-options]`
- `[edit interfaces` *`interface-name`* `ether-options]`
- `[edit interfaces` *`interface-name`* `fastether-options]`
- `[edit interfaces` *`interface-name`* `gigether-options]`

On the Type 5 FPC, to prioritize control packets in case of ingress oversubscription, you must ensure that the neighboring peers support MAC flow control. If the peers do not support MAC flow control, then you must disable flow control.

Set the Mode on an SFP+ or SFP+ MACsec Uplink Module

You can use SFP+ and SFP+ MACsec uplink modules either for two SFP+ transceivers or four SFP transceivers. You configure the operational mode on the module to match the transceiver type. For SFP+ transceivers, configure the 10-gigabit operational mode, and for SFP transceivers, you configure the 1-gigabit operational mode.

By default, the SFP+ uplink module operates in the 10-gigabit mode and supports only SFP+ transceivers. If you have not changed the module from the default setting and you want to use SFP+ transceivers, you do not need to configure the operational mode.

Use MACsec to confirm platform and release support for specific features.

To set the operational mode of an SFP+ or SFP+ MACsec uplink module:

1. Change the operating mode to the appropriate mode for the transceiver type you want to use by using one of the following commands:

   ```
   [edit]
   user@switch# set chassis fpc 0 pic 1 sfpplus pic-mode 1g
   ```

   ```
   [edit]
   user@switch# set chassis fpc 0 pic 1 sfpplus pic-mode 10g
   ```

2. (SFP+ uplink module only) If the switch is running:

   - Junos OS Release 10.1 or later, the changed operating mode takes effect immediately unless a port on the SFP+ uplink module is a Virtual Chassis port (VCP). If any port on the SFP+ uplink module is a VCP, the changed operating mode does not take effect until the next reboot of the switch.

     During the operating mode change, the Packet Forwarding Engine is restarted. In a Virtual Chassis configuration, this means that the Flexible PIC Concentrator connection with the primary device is dropped and then reconnected.

   - Junos OS Release 10.0 or earlier, reboot the switch.

You can see whether the operating mode has been changed to the new mode you configured by issuing the `show chassis pic fpc-slot` *`slot-number`* `pic-slot 1` command.

Set the Operational Mode on a 2-Port 40-Gigabit Ethernet QSFP+/100-Gigabit Ethernet QSFP28 Uplink Module

You can configure the 2-port 4-Gigabit Ethernet QSFP+/100-Gigabit Ethernet QSFP28 uplink module on EX4300-48MP switches to operate either two 40-Gigabit Ethernet ports or two 100-Gigabit Ethernet ports. By default, the uplink module operates only the two 40-Gbps ports.

The uplink module on EX4300-48MP switches supports MACsec. See *Understanding Media Access Control Security (MACsec)* for more information.

The uplink module does not support configuring VCPs.

To set the operational mode on this uplink module:

1. Install the two-port 40GbE QSFP+/100GbE QSFP28 uplink module only in PIC slot 2 on the switch. Insert the uplink module in the chassis and check whether it is detected by issuing the `show chassis hardware` command.
2. Change the operational mode to 100-Gigabit Ethernet mode, by issuing the following command on the first port (port 0). The port then recognizes the 100-Gigabit speed and disables the adjacent 40GbE port. The adjacent 40GbE port is disabled only when port 0 is loaded with 100GbE optics.

```
[edit]
user@switch# set chassis fpc 0 pic 2 port 0 speed 100G
```

3. You can change the operational mode to 100-Gigabit Ethernet mode on the second (port 1) by using the following command. This command overrides the `set chassis fpc 0 pic 2 port 0 speed 100G` command to change the operational mode to 100GbE mode.

```
[edit]
user@switch# run request chassis system-mode mode-2x100G
```

4. Optional: Check whether the operational mode has been changed to the new mode you configured by issuing the `show chassis pic fpc-slot` *0* `pic-slot 2` command.

If you configure both the ports on the uplink module to operate at 100-Gbps speed, the four built-in QSFP+ ports on the switch are disabled.

Starting with Junos OS Release 19.1R1, you can channelize the 100GbE to four independent 25GbE ports by using breakout cables. You can configure only port 0 of the uplink module as 25GbE port. Issue the command `set chassis fpc 0 pic 2 port 0 channel-speed 25g` to channelize the 100GbE uplink port to four 25GbE uplink ports.

Starting with Junos OS Release 19.3R1, you can configure the 2-port 40-Gigabit Ethernet QSFP+/100-Gigabit Ethernet QSFP28 uplink module on EX4300-48MP switches to operate either two 40-Gigabit Ethernet ports or two 100-Gigabit Ethernet ports.

You can also channelize the 40-Gigabit Ethernet interfaces to four independent 10-Gigabit Ethernet interfaces using breakout cables. To channelize the 100-Gigabit Ethernet interfaces to operate as four independent 25-Gigabit Ethernet, specify the port number and channel speed

1. To configure the 100-Gigabit Ethernet uplink port to operate as a 25-Gigabit Ethernet interface, specify the port number and channel speed by using the following command:

   ```
   [edit chassis fpc 0 pic 2]
   user@switch# set port port-number channel-speed speed
   ```

   For example, to configure port 0 to operate as a 25-Gigabit Ethernet port:

   ```
   [edit chassis fpc 0 pic 2]
   user@switch# set port 0 channel-speed 25g
   ```

2. Review your configuration and issue the `commit` command.

   ```
   [edit]
   user@switch# commit
   commit complete
   ```

If you configure both the ports on the uplink module to operate at 100-Gbps speed, the four QSFP+ ports on the switch are disabled.

Configure the Media Type on Dual-Purpose Uplink Ports

EX2200-C switches and ACX1000 routers provide two dual-purpose uplink ports. Each dual uplink port is a single interface that offers a choice of two connections: an RJ-45 connection for a copper Ethernet cable and an SFP connection for a fiber-optic Ethernet cable. You can choose to use either connection, but only one connection can be active at a time.

By default, if you plug a transceiver into the SFP connector, the port becomes a fiber-optic Gigabit Ethernet port, even if a copper Ethernet cable is plugged into the RJ-45 connection as well. If a transceiver is not plugged into the SFP connector, the port defaults to a copper 10/100/1000 Ethernet port.

You can constrain the use of the port to one connection type by configuring the media type for the port to be either copper or fiber. When you configure a media type on the port, the port will no longer accept the alternate connection type. For example, if you configure the uplink port as a fiber port and then plug a copper Ethernet cable into the RJ-45 connector, the interface will not come up.

To configure the media type for an uplink port:

```
user@switch# set interfaces interface-name media-type (Dual-Purpose Uplink Ports) media-type
```

For example, to set the media type for uplink port **ge-0/1/0** to copper:

```
user@switch# set interfaces ge-0/1/0 media-type copper
```

NOTE: When you change the media type setting for a dual-purpose uplink port, it can take up to 6 seconds for the interface to appear in operational commands.

Change History Table

Feature support is determined by the platform and release you are using. Use Feature Explorer to determine if a feature is supported on your platform.

Release	Description
19.3R1	Starting with Junos OS Release 19.3R1, you can configure the 2-port 40-Gigabit Ethernet QSFP+/100-Gigabit Ethernet QSFP28 uplink module on EX4300-48MP switches to operate either two 40-Gigabit Ethernet ports or two 100-Gigabit Ethernet ports.
19.1R1	Starting with Junos OS Release 19.1R1, in the 2-port 40-Gigabit Ethernet QSFP+/1-port 100-Gigabit Ethernet QSFP28 uplink module of EX4300-48MP switches, you can channelize the 100-Gigabit four independent 25-Gigabit Ethernet ports by using breakout cables.
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 `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.
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.

Logical Interface Properties

IN THIS SECTION

- Logical Interface Properties Overview | 100
- Specify the Logical Interface Number | 101
- Add a Logical Unit Description to the Configuration | 101
- Configure the Interface Bandwidth | 102
- Configure Interface Encapsulation on Logical Interfaces | 103
- Configure Interface Encapsulation on PTX Series Routers | 105
- Configure a Point-to-Point Connection | 107
- Configure a Multipoint Connection | 107
- Configure Dynamic Profiles for PPP | 108
- Overview of Accounting for the Logical Interface | 108
- Enable or Disable SNMP Notifications on Logical Interfaces | 112
- Disable a Logical Interface | 113

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 statements at the following hierarchy levels:

- `[edit interfaces interface-name]`
- `[edit logical-systems logical-system-name 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. For more information, see "Interface Naming Overview" on page 6.

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

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

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

You can include this statement at the following hierarchy levels:

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

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

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 following hierarchy levels:

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

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

> **NOTE**: You can configure the extended DHCP relay to include the interface description in the option 82 Agent Circuit ID suboption. See DHCP Relay Agent Information Option (Option 82).

For information about describing physical interfaces, see "Configure the Interface Description" on page 55.

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 levels:

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

`rate` is the peak rate, in 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 | 103
- Configure the Encapsulation on a Logical Interface | 104
- Display the Encapsulation on a Logical Interface | 104

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. For more information, see *Configuring VLAN and Extended VLAN Encapsulation*.

- You can create an ATM cell-relay circuit by configuring an entire ATM physical device or an individual virtual circuit (VC). When you configure an entire device, only cell-relay encapsulation is the only encapsulation type allowed on the logical interfaces. For more information, see Configuring an ATM1 Cell-Relay Circuit Overview.

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*]` or `[edit logical-systems *logical-system-name* interfaces *interface-name* unit *logical-unit-number*]` hierarchy level.

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

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

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

Display the Encapsulation on a Logical Interface

IN THIS SECTION

- Purpose | **105**
- Action | **105**
- Meaning | **105**

Purpose

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

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

Action

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

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

Meaning

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

RELATED DOCUMENTATION

encapsulation (Logical Interface)

Configuring VLAN and Extended VLAN Encapsulation

Configuring an ATM1 Cell-Relay Circuit Overview

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

In Junos OS Evolved, the flexible-ethernet-services encapsulation is not supported on PTX10003 devices.

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;
                }
            }
        }
    }
}
```

Configure a Point-to-Point Connection

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

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

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

You can include this statement at the following hierarchy levels:

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

Configure a Multipoint Connection

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

```
multipoint;
```

You can include this statement at the following hierarchy levels:

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

Configure Dynamic Profiles for PPP

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

After dynamic profiles are created, the profiles reside in a profile library on the router. You can then use the `dynamic-profile` statement to attach profiles to interfaces. To assign a dynamic profile to a PPP interface, you can include the `dynamic-profile` statement at the `[edit interfaces interface-name unit logical-unit-number ppp-options]` hierarchy level:

```
[edit interfaces interface-name unit logical-unit-number ppp-options]
dynamic-profile profile-name;
```

To monitor the configuration, issue the `show interfaces interface-name` command.

For information about dynamic profiles, see *Dynamic Profiles Overview* in the *Junos Subscriber Access Configuration Guide*.

For information about creating dynamic profiles, see *Configuring a Basic Dynamic Profile* in the *Junos Subscriber Access Configuration Guide*.

For information about assigning a dynamic profile to a PPP interface, see *Attaching Dynamic Profiles to Static PPP Subscriber Interfaces* in the *Junos Subscriber Access Configuration Guide*.

For information about using dynamic profiles to authenticate PPP subscribers, see *Configuring Dynamic Authentication for PPP Subscribers*.

NOTE: Dynamic profiles for PPP subscribers are supported only on PPPoE interfaces for this release.

Overview of Accounting for the Logical Interface

IN THIS SECTION

- Accounting Profiles Overview | **109**
- Configure Accounting for the Logical Interface | **109**

- Display the Accounting Profile for the Logical Interface | 111

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. For more information, see the Junos OS Network Management Administration Guide for Routing Devices.

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

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. For more information about how to configure an accounting-data log file, see the *Configuring Accounting-Data Log Files*.

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

1. To configure which statistics 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. For more information, see Configuring Accounting-Data Log Files.

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

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

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

4. To configure the interfaces on which the accounting needs to be performed, apply the interface profile to a 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
    ```

SEE ALSO

Accounting Options Overview

Configuring Accounting-Data Log Files

Display the Accounting Profile for the Logical Interface

IN THIS SECTION

- Purpose | **111**
- Action | **111**
- Meaning | **112**

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—ge-1/0/1
- Logical unit number—1
- Interface profile —`if_profile`
- File name—`if_stats`
- Interval—15 minutes

Action

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

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

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

  ```
  interface-profile if_profile {
      interval 15;
  ```

```
    file if_stats {
        fields {
            input-bytes;
            output-bytes;
            input-packets;
            output-packets;
            input-errors;
            output-errors;
        }
    }
}
```

Meaning

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

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 levels:

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

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

When an interface is disabled, a route (pointing to the reserved target “`REJECT`”) with the IP address of the interface and a 32–bit subnet mask is installed in the routing table. See *Routing Protocols*.

Example: 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;
        }
}
```

Understanding Interfaces

IN THIS SECTION

- Network Interfaces | 114
- Understanding Interface Naming Conventions | 124
- Understanding Management Interfaces | 143

Junos OS supports different types of interfaces on which the devices function. The following topics provide information of types of interfaces used, the naming conventions and the usage of management interfaces by Juniper Networks.

Network Interfaces

IN THIS SECTION

- Network Interfaces for EX Series | 115
- Special Interfaces for EX Series | 116

- Network Interfaces for EX4600, NFX Series, QFX Series, QFabric System | 119
- Special Interfaces for EX4600, NFX Series, QFX Series, QFabric System | 121
- Network Interfaces for OCX Series | 122
- Special Interfaces for OCX Series | 123

Juniper Networks devices have two types of interfaces: network interfaces and special interfaces. This topic provides brief information about these interfaces. For additional information, see the Junos OS Network Interfaces Library for Routing Devices.

Network Interfaces for EX Series

Network interfaces connect to the network and carry network traffic. Table 7 on page 115 lists the types of network interfaces supported on EX Series switches.

Table 7: Network Interfaces Types and Purposes for EX Series

Type	Purpose
Aggregated Ethernet interfaces	All EX Series switches allow you to group Ethernet interfaces at the physical layer to form a single link layer interface. This group is also known as a *link aggregation group (LAG)* or *bundle*. These aggregated Ethernet interfaces help to balance traffic and increase the uplink bandwidth. See *Understanding Aggregated Ethernet Interfaces and LACP for Switches*.
LAN access interfaces	Use these EX Series switch interfaces to connect the following to the network: • PC • Laptop • File server • Printer When you power on an EX Series switch and use the factory-default configuration, the software automatically configures interfaces in access mode for each of the network ports. The default configuration also enables autonegotiation for both speed and link mode.

Table 7: Network Interfaces Types and Purposes for EX Series *(Continued)*

Type	Purpose
Power over Ethernet (PoE) interfaces	EX Series switches provide PoE network ports with various switch models. Use these ports to connect VoIP telephones, wireless access points, video cameras, and point-of-sale devices to safely receive power from the same access ports that are used to connect personal computers to the network. PoE interfaces are enabled by default in the factory configuration. See Understanding PoE on EX Series Switches.
Trunk interfaces	You can connect EX Series access switches to a distribution switch or customer-edge (CE) switches or routers. To use a port for this type of connection, you must explicitly configure the network interface for trunk mode. You must also configure the interfaces from the distribution switch or CE switch to the access switches for trunk mode.

Special Interfaces for EX Series

Table 8 on page 116 lists the types of special interfaces supported on EX Series switches.

Table 8: Special Interfaces Types and Purposes for EX Series

Type	Purpose
Console port	Each EX Series switch has a serial port, labeled **CON** or **CONSOLE**, for connecting tty-type terminals to the switch using standard PC-type tty cables. The console port does not have a physical address or IP address associated with it. However, it is an interface since it provides access to the switch. On an EX3300 *Virtual Chassis*, an EX4200 Virtual Chassis, or an EX4500 Virtual Chassis, you can access the primary device and configure all members of the Virtual Chassis through any member's console port. For more information about the console port in a Virtual Chassis, see Understanding Global Management of a Virtual Chassis.
Loopback	All EX Series switches have this software-only virtual interface that is always up. The loopback interface provides a stable and consistent interface and IP address on the switch.

Table 8: Special Interfaces Types and Purposes for EX Series *(Continued)*

Type	Purpose
Management interface	The Juniper Networks Junos operating system (Junos OS) for EX Series switches automatically creates the switch's management Ethernet interface, me0. The management Ethernet interface provides an out-of-band method for connecting to the switch. To use me0 as a management port, you must configure its logical port, me0.0, with a valid IP address. You can connect to the management interface over the network using utilities such as SSH or Telnet. SNMP can use the management interface to gather statistics from the switch. (The management interface me0 is analogous to the fxp0 interfaces on routers running Junos OS.) See *Understanding Management Interfaces*.
Integrated Routing and Bridging (IRB) Interface or *Routed VLAN Interface* (RVI)	EX Series switches use an integrated routing and bridging (IRB) interface or Routed VLAN Interface (RVI) to route traffic from one broadcast domain to another and to perform other Layer 3 functions such as traffic engineering. These functions are typically performed by a router interface in a traditional network. The IRB interface or RVI functions as a logical router, eliminating the need for having both a switch and a router. Configure these interfaces as part of a broadcast domain or Virtual Private LAN Service (VPLS) routing instance for L3 traffic to be routed from. See Understanding Integrated Routing and Bridging.

Table 8: Special Interfaces Types and Purposes for EX Series *(Continued)*

Type	Purpose
Virtual Chassis port (VCP) interfaces	Virtual Chassis ports (VCPs) are used to interconnect switches in a *Virtual Chassis*: - EX3300 switches—Port 2 and port 3 of the SFP+ uplink ports are preconfigured as VCPs and can be used to interconnect up to six EX3300 switches in an EX3300 Virtual Chassis. See *Setting an Uplink Port on an EX Series or QFX Series Switch as a Virtual Chassis Port*. - EX4100, EX4100-24MP, EX4100-48MP, and EX4100-F switches—Each EX4100, EX4100-24MP, EX4100-48MP, or EX4100-F switch has dedicated VCP ports. You cannot use any other ports on EX4100 switches as VCPs. See EX4100/EX4100-F Switches in a Virtual Chassis. - EX4200 and EX4500 switches—Each EX4200 switch or each EX4500 switch with a Virtual Chassis module installed has two dedicated VCPs on its rear panel. These ports can be used to interconnect up to ten EX4200 switches in an EX4200 Virtual Chassis, up to ten EX4500 switches in an EX4500 Virtual Chassis, and up to ten switches in a mixed EX4200 and EX4500 Virtual Chassis. When you power on switches that are interconnected in this manner, the software automatically configures the VCP interfaces for the dedicated ports that have been interconnected. These VCP interfaces are not configurable or modifiable. See *Understanding the High-Speed Interconnection of the Dedicated Virtual Chassis Ports Connecting EX4200, EX4500, and EX4550 Member Switches*. You can also interconnect EX4200 and EX4500 switches by using uplink module ports. Using uplink ports allows you to connect switches over longer distances than you can by using the dedicated VCPs. To use the uplink ports as VCPs, you must explicitly configure the uplink module ports on the members you want to connect as VCPs. See *Setting an Uplink Port on an EX Series or QFX Series Switch as a Virtual Chassis Port* . - EX4300 switches—All QSFP+ ports are configured as VCPs by default. See *Understanding EX Series Virtual Chassis*. You can also interconnect EX4300 switches into a Virtual Chassis by using SFP+ uplink module ports as VCPs. Using uplink ports as VCPs allows you to connect switches over longer distances than you can by using the QSFP+ ports as VCPs. To use the uplink ports as VCPs, you must explicitly configure the uplink module ports on the members you want to connect as VCPs. See *Setting an Uplink Port on an EX Series or QFX Series Switch as a Virtual Chassis Port*.

Table 8: Special Interfaces Types and Purposes for EX Series *(Continued)*

Type	Purpose
	- EX8200 switches—EX8200 switches can be connected to an XRE200 External Routing Engine to create an EX8200 Virtual Chassis. The XRE200 External Routing Engine has dedicated VCPs that connect to ports on the internal Routing Engines of the EX8200 switches and can connect to another XRE200 External Routing Engine for redundancy. These ports require no configuration. . You can also connect two members of an EX8200 Virtual Chassis so that they can exchange Virtual Chassis Control Protocol (VCCP) traffic. To do so, you explicitly configure network ports on the EX8200 switches as VCPs.
Virtual management Ethernet (VME) interface	EX3300, EX4200, EX4300, and EX4500 switches have a VME interface. This is a *logical interface* that is used for Virtual Chassis configurations and allows you to manage all the members of the Virtual Chassis through the primary device. For more information about the VME interface, see Understanding Global Management of a Virtual Chassis. EX8200 switches do not use a VME interface. An EX8200 Virtual Chassis is managed through the management Ethernet (me0) interface on the XRE200 External Routing Engine.

Network Interfaces for EX4600, NFX Series, QFX Series, QFabric System

Network interfaces connect to the network and carry network traffic. Table 9 on page 119 lists the types of network interfaces supported.

Table 9: Network Interfaces Types and Purposes for EX4600, NFX Series, QFX Series, QFabric System

Type	Purpose
Aggregated Ethernet interfaces	Group Ethernet interfaces at the physical layer to form a single link-layer interface, also known as a *link aggregation group (LAG)* or *bundle*. These aggregated Ethernet interfaces help to balance traffic and increase the uplink bandwidth.

Table 9: Network Interfaces Types and Purposes for EX4600, NFX Series, QFX Series, QFabric System *(Continued)*

Type	Purpose
Channelized Interfaces	Depending on the device and software package, 40-Gbps QSFP+ ports can be configured to operate as the following types of interfaces: • 10-Gigabit Ethernet interfaces (*xe*) • 40-Gigabit Ethernet interfaces (*et* and *xle*) • 40-Gigabit data plane uplink interfaces (*fte*) When an *et* port is channelized to four *xe* ports, a colon is used to signify the four separate channels. For example, on a QFX3500 standalone switch with port 2 on PIC 1 configured as four 10-Gigabit Ethernet ports, the interface names are *xe-0/1/2:0*, *xe-0/1/2:1*, *xe-0/1/2:2*, and *xe-0/1/2:3* **NOTE**: You cannot configure channelized interfaces to operate as Virtual Chassis ports.
Ethernet Interfaces	Configure Gigabit Ethernet, 10-Gigabit Ethernet, 40-Gigabit Ethernet interfaces to connect to other servers, storage, and switches. You can configure 40-Gigabit data plane uplink ports to connect a Node device to an Interconnect devices as well as for Virtual Chassis ports (VCPs).
Fibre Channel interfaces	Use Fibre Channel interfaces to connect the switch to a Fibre Channel over Ethernet (FCoE) forwarder or a Fibre Channel switch in a storage area network (SAN). You can configure Fibre Channel interfaces only on ports 0 through 5 and 42 through 47 on QFX3500 devices. Fibre Channel interfaces do not forward Ethernet traffic. See *Overview of Fibre Channel*.
LAN access interfaces	Use these interfaces to connect to other servers, storage, and switches. When you power on a QFX Series product and use the factory-default configuration, the software automatically configures interfaces in access mode for each of the network ports.
Multichassis aggregated Ethernet (MC-AE) interfaces	Group a LAG on one standalone switch with a LAG on another standalone switch to create a MC-AE. The MC-AE provides load balancing and redundancy across the two standalone switches.

Table 9: Network Interfaces Types and Purposes for EX4600, NFX Series, QFX Series, QFabric System *(Continued)*

Type	Purpose
Tagged-access mode interfaces	Use tagged-access interfaces to connect a switch to an access layer device. Tagged-access interfaces can accept VLAN-tagged packets from multiple VLANs.
Trunk interfaces	Use trunk interfaces to connect to other switches or routers. To use a port for this type of connection, you must explicitly configure the port interface for trunk mode. The interfaces from the switches or routers must also be configured for trunk mode. In this mode, the interface can be in multiple VLANs and accept tagged packets from multiple devices. Trunk interfaces typically connect to other switches and to routers on the LAN.
Virtual Chassis ports (VCPs)	You can use Virtual Chassis ports to send and receive Virtual Chassis Control Protocol (VCCP) traffic, and to create, monitor, and maintain the Virtual Chassis. On QFX3500, QFX3600, QFX5100, QFX5110, QFX5200, and EX4600 standalone switches, you can configure 40-Gigabit Ethernet QSFP+ uplink ports (non-channelized) or fixed SFP+ 10-Gigabit Ethernet ports as VCPs by issuing the `request virtual-chassis-vc-port-set` CLI command. QFX5110 switches also support configuring 100-Gigabit QSFP28 ports as VCPs.

Special Interfaces for EX4600, NFX Series, QFX Series, QFabric System

Table 10 on page 121 lists the types of special interfaces supported.

Table 10: Special Interfaces Types and Purposes supported on EX4600, NFX Series, QFX Series, QFabric System

Type	Purpose
Console port	Each device has a serial console port, labeled **CON** or **CONSOLE**, for connecting tty-type terminals to the switch. The console port does not have a physical address or IP address associated with it. However, it is an interface in the sense that it provides access to the switch.
Loopback interface	A software-only virtual interface that is always up. The loopback interface provides a stable and consistent interface and IP address on the switch.

Table 10: Special Interfaces Types and Purposes supported on EX4600, NFX Series, QFX Series, QFabric System *(Continued)*

Type	Purpose
Management interface	The management Ethernet interface provides an out-of-band method for connecting to a standalone switch and QFabric system. **NOTE**: On OCX Series switches, the em0 management interface always has the status `up` in `show` command outputs, even if the physical port is empty. The me0 interface is a virtual interface between Junos and the host operating system, therefore its status is independent from the status of the physical port.
Routed VLAN interfaces (*RVI* and IRB interfaces)	Layer 3 routed VLAN interfaces (called RVI in the original CLI, and called IRB in Enhanced Layer 2 Software) route traffic from one broadcast domain to another and perform other Layer 3 functions such as traffic engineering. These functions are typically performed by a router interface in a traditional network. The RVI or IRB functions as a logical router, eliminating the need for having both a switch and a router. The RVI or IRB must be configured as part of a broadcast domain or virtual private LAN service (VPLS) routing instance for Layer 3 traffic to be routed out of it.

Network Interfaces for OCX Series

Network interfaces connect to the network and carry network traffic. Table 11 on page 122 lists the types of network interfaces supported.

Table 11: Network Interfaces Types and Purposes for OCX Series

Type	Purpose
Aggregated Ethernet interfaces	Group Ethernet interfaces at the physical layer to form a single link-layer interface, also known as a *link aggregation group (LAG)* or *bundle*. These aggregated Ethernet interfaces help to balance traffic and increase the uplink bandwidth.
Ethernet Interfaces	Configure Gigabit Ethernet, 10-Gigabit Ethernet, 40-Gigabit Ethernet interfaces to connect to other servers, storage, and switches.

Special Interfaces for OCX Series

Table 12 on page 123 lists the types of special interfaces supported.

Table 12: Special Interfaces Types and Purposes for OCX Series

Type	Purpose
Console port	Each device has a serial console port, labeled **CON** or **CONSOLE**, for connecting tty-type terminals to the switch. The console port does not have a physical address or IP address associated with it. However, it is an interface in the sense that it provides access to the switch.
Loopback interface	A software-only virtual interface that is always up. The loopback interface provides a stable and consistent interface and IP address on the switch.
Management interface	The management Ethernet interface provides an out-of-band method for connecting to a standalone switch and QFabric system. **NOTE**: On OCX Series switches, the em0 management interface always has the status `up` in `show` command outputs, even if the physical port is empty. The me0 interface is a virtual interface between Junos and the host operating system, therefore its status is independent from the status of the physical port.

SEE ALSO

- EX2200 Switches Hardware Overview
- EX3200 System Overview
- EX3300 Switches Hardware Overview
- *EX4200 Switches Hardware Overview*
- *EX4300 Switches Hardware Overview*
- EX4500 Switches Hardware Overview
- EX6210 Switch Hardware Overview
- EX8208 Switch Hardware Overview
- EX8216 Switch Hardware Overview
- Understanding Layer 3 Logical Interfaces
- Understanding Layer 3 Subinterfaces

Understanding Interface Naming Conventions

IN THIS SECTION

- Physical Part of an Interface Name for EX Series | 124
- Logical Part of an Interface Name for EX Series | 127
- Wildcard Characters in Interface Names for EX Series | 127
- Physical Part of an Interface Name for QFX series, NFX Series, EX4600, QFabric System | 127
- Logical Part of an Interface Name on a Switch Running QFabric Software Package for QFX series, NFX Series, EX4600, QFabric System | 141
- Logical Part of a Channelized Interface Name on a Switch Running Enhanced Layer 2 Software for QFX series, NFX Series, EX4600, QFabric System | 142
- Wildcard Characters in Interface Names for QFX series, NFX Series, EX4600, QFabric System | 142
- Physical Part of an Interface Name for OCX1100 | 142
- Wildcard Characters in Interface Names for OCX1100 | 143

The EX Series, QFX Series, NFX Series, OCX1100, QFabric System, and EX4600 devices use a naming convention for defining the interfaces that are similar to that of other platforms running under Juniper Networks Junos OS. This topic provides brief information about the naming conventions used for interfaces on the QFX Series and on EX4600 switches.

For detailed information on interface naming like physical part, logical part, and channel part of the interfaces, see Interface Naming Overview.

This topic describes:

Physical Part of an Interface Name for EX Series

Network interfaces in Junos OS are specified as follows:

type-fpc / *pic* / *port*

EX Series switches apply this convention as follows:

- *type*—EX Series interfaces use the following media types:
  - ge—Gigabit Ethernet interface
  - xe—10 Gigabit Ethernet interface

  - et—40 Gigabit Ethernet interface

- *fpc*—Flexible PIC Concentrator. EX Series interfaces use the following convention for the FPC number in interface names:

  - On an EX2200 switch, an EX2300, an EX3200 switch, a standalone EX3300 switch, a standalone EX3400 switch, a standalone EX4200 switch, a standalone EX4300 switch, a standalone EX4500, and a standalone EX4550 switch, FPC refers to the switch itself. The FPC number is **0** by default on these switches.

  - On an EX3300 *Virtual Chassis*, an EX3400 Virtual Chassis, an EX4200 Virtual Chassis, an EX4300 Virtual Chassis, an EX4500 Virtual Chassis, an EX4550 Virtual Chassis, or a mixed Virtual Chassis, the FPC number indicates the member ID of the switch in the Virtual Chassis.

  - On EX4100 and EX4100-F switches, the FPC number ranges from **0** to **9**. On a standalone EX4100 or EX4100-F switch, FPC refers to the switch. The FPC number is **0** by default on the standalone switches.

  - On EX4100 and EX4100-F Virtual Chassis, the FPC number indicates the member ID of the switch in the Virtual Chassis.

  - On an EX6200 switch and a standalone EX8200 switch, the FPC number indicates the slot number of the line card that contains the physical interface. On an EX6200 switch, the FPC number also indicates the slot number of the Switch Fabric and Routing Engine (SRE) module that contains the uplink port.

  - On an EX8200 Virtual Chassis, the FPC number indicates the slot number of the line card on the Virtual Chassis. The line card slots on Virtual Chassis member 0 are numbered 0 through 15; on member 1, they are numbered 16 through 31, and so on.

  - On EX9251 switch, the FPC number is always **0**.

  - The EX9253 switch does not have actual FPCs—the line cards are the FPC equivalents on the switch. In FPC (n), n is a value in the range of 0-1. The value corresponds to the line card slot number in which the line card is installed.

  - On an EX29204 switch, switch does not have actual FPCs—the line cards are the FPC equivalents on the switch. The value ranges from 0-2, and it corresponds to the line card slot number in which the line card is installed.

- *pic*—EX Series interfaces use the following convention for the PIC (*Physical Interface Card*) number in interface names:

  - On EX2200, EX2300, EX3200, EX3300, EX4200, EX4500 switch, and EX4550 switches, the PIC number is **0** for all built-in interfaces (interfaces that are not uplink ports).

  - On EX2200, EX2300, EX3200, EX3300, and EX4200 switches, the PIC number is **1** for uplink ports.

  - On EX3400 switches, the PIC number is **0** for built-in network ports, **1** for built-in QSFP+ ports (located on the rear panel of the switch), and **2** for uplink module ports.

  - On EX4100 and EX4100-F switches, the PIC number ranges from **0** to **2**. The PIC number is **0** for built-in network ports, **1** for SFP28/SFP+ dedicated Virtual Chassis ports, and **2** for SFP/SFP+ uplink ports.

  - On EX4300 switches, the PIC number is **0** for built-in network ports, **1** for built-in QSFP+ ports (located on the rear panel of the switch), and **2** for uplink module ports.

  - On EX4500 switches, the PIC number is **1** for ports on the left-hand uplink module and **2** for ports on the right-hand uplink module.

  - On EX4550 switches, the PIC number is **1** for ports in the expansion module or Virtual Chassis module installed in the module slot on the front panel of the switch and **2** for those in the expansion module or Virtual Chassis module installed in the module slot on the rear panel of the switch.

  - On EX6200 and EX8200 switches, the PIC number is always **0**.

  - On EX9251 and EX9253 switches, the PIC number is **0** for built-in network ports, **1** for built-in QSFP+ ports (located on the rear panel of the switch).

  - On EX9204 switches, the PIC number ranges from 0-3.

- *port*—EX Series interfaces use the following convention for port numbers:

  - On EX2200, EX2300, EX3200, EX3300, EX3400, EX4200, EX4300, EX4500, and EX4550 switches, built-in network ports are numbered from left to right. On models that have two rows of ports, the ports on the top row start with **0** followed by the remaining even-numbered ports, and the ports on the bottom row start with **1** followed by the remaining odd-numbered ports.

  - Uplink ports in EX2200, EX3200, EX3300, EX3400, EX4200, EX4300, EX4500, and EX4550 switches are labeled from left to right, starting with **0**.

  - On EX4100 and EX4100-F switches, the uplink ports are labeled from 0 to 3. The Virtual Chassis ports are also labeled from 0 to 3. The downlink ports are labeled from 0 to 47 (for EX4100-48P, EX4100-48T, EX4100-F-48P, and EX4100-F-48T switches) and from 0 to 23 (for EX4100-24P, EX4100-24T, EX4100-F-24P and EX4100-F-24T switches).

  - On EX6200 and EX8200 switches, the network ports are numbered from left to right on each line card. On line cards that have two rows of ports, the ports on the top row start with **0** followed by the remaining even-numbered ports, and the ports on the bottom row start with **1** followed by the remaining odd-numbered ports.

- Uplink ports on an SRE module in an EX6200 switch are labeled from left to right, starting with **0**.
- EX9251 Switch has eight 10-Gigabit Ethernet ports and four rate-selectable ports that you can configure as 100-Gigabit Ethernet ports or 40-Gigabit Ethernet ports; each rate-selectable port can be configured as four 10-Gigabit Ethernet ports by using a breakout cable. The 10-Gigabit Ethernet ports support SFP+ transceivers and rate-selectable ports support QSFP28 and QSFP+ transceivers.
- EX9253 contains six built-in QSFP+ ports, each of which can house QSFP+ pluggable transceivers and 12 built-in QSFP28 ports, each of which can house QSFP28 pluggable transceivers.

Logical Part of an Interface Name for EX Series

The logical unit part of the interface name corresponds to the logical unit number, which can be a number from 0 through 16384. In the virtual part of the name, a period (.) separates the port and logical unit numbers: *type-fpc/pic/port.logical-unit-number*. For example, if you issue the `show ethernet-switching interfaces` command on a system with a default VLAN, the resulting display shows the logical interfaces associated with the VLAN:

```

Interface   State    VLAN members          Blocking
ge-0/0/0.0  down     remote-analyzer       unblocked
ge-0/0/1.0  down     default               unblocked
ge-0/0/10.0 down     default               unblocked
```

Wildcard Characters in Interface Names for EX Series

In the `show interfaces` and `clear interfaces` commands, you can use wildcard characters in the ***interface-name*** option to specify groups of interface names without having to type each name individually. You must enclose all wildcard characters except the asterisk (*) in square brackets [ ].

Physical Part of an Interface Name for QFX series, NFX Series, EX4600, QFabric System

Interfaces in Junos OS are specified as follows:

device-name:type-fpc/pic/port

The convention is as follows (and platform support depends on the Junos OS release in your installation):

- *device-name*—(QFabric systems only) The *device-name* is either the serial number or the alias of the QFabric system component, such as a Node device, Interconnect device, or QFabric infrastructure. The name can contain a maximum of 128 characters and cannot contain any colons.
- *type*—The QFX Series and EX4600 device interfaces use the following media types:
  - **fc**—Fibre Channel interface
  - **ge**—Gigabit Ethernet interface
  - **xe**—10-Gigabit Ethernet interface
  - *sxe*—10-Gigabit Service interface. *sxe* is an internal interface and user must not configure this interface. It supports L2 and L3 configurations like VLANs and IP address.
  - **xle**—40-Gigabit Ethernet interface (QFX3500, QFX3600, and QFX5100 switches running a QFabric software package)
  - **et**—25-Gigabit Ethernet interface (QFX5120 and QFX5200 switches)
  - **et**—40-Gigabit Ethernet interface (QFX3500, QFX3600, QFX5100, QFX5200, QFX10000, and EX4600 switches running Enhanced Layer 2 Software)
  - **et**—100-Gigabit Ethernet interface (QFX5200 and QFX10000 switches running Enhanced Layer 2 Software)
  - **fte**—40-Gigabit data plane uplink interface (QFX3500, QFX3600, and QFX5100 switches running a QFabric software package)
  - **me**—Management interface
  - **em**—Management interface on QFX5100 and EX4600 switches.
- *fpc*—Flexible PIC Concentrator. QFX Series interfaces use the following convention for the FPC number in interface names:
  - On QFX3500, QFX3600, QFX5100 devices running a QFabric software package, and QFX10002 switches, the FPC number is always 0.

    The FPC number indicates the slot number of the line card that contains the physical interface.
  - On QFX3500, QFX3600, QFX5100, QFX5200, EX4600, QFX10002, QFX10008, and QFX10016 switches running Enhanced Layer 2 Software, the member ID of a member in a Virtual Chassis determines the FPC number.

    > **NOTE**: Every member in a Virtual Chassis must have a unique member ID, otherwise the Virtual Chassis will not be created.

  - On standalone QFX5100, EX4600, and QFX10002 switches, the FPC number is always 0.

- *pic*—QFX Series and EX4600 device interfaces use the following convention for the PIC (*Physical Interface Card*) number in interface names:

Table 13: Naming Conventions for PICs

Device with Software Package	Convention
QFX3500 switch with QFabric software package	PIC 0 can support 48 ports, PIC 1 can support 16 10-Gigabit Ethernet ports, and PIC 2 can support 4 40-Gigabit Ethernet ports.
QFX3500 switch with Enhanced Layer 2 software	PIC **0** can support 48 ports, and PIC **1** can support 16 10-Gigabit Ethernet ports, and 4 40-Gigabit Ethernet ports.
QFX3500 Node device with a QFabric software package	PIC **0** can support 48 ports and PIC **1** can support four 40-Gigabit data plane uplink ports.
QFX3600 switch with a QFabric software package	PIC **0** can support 64 10-Gigabit Ethernet ports, and PIC **1** can support 16 40-Gigabit Ethernet ports.
QFX3600 switch with Enhanced Layer 2 software	PIC **0** can support 64 10-Gigabit Ethernet ports and can also support 16 40-Gigabit Ethernet ports.
QFX3600 Node device running a QFabric software package	PIC **0** can support 56 10-Gigabit Ethernet ports, and PIC **1** can support 8 40-Gigabit data plane uplink ports, and up to 14 40-Gigabit Ethernet ports.
QFX5100-48S switch with Enhanced Layer 2 software	PIC **0** provides six 40-Gbps QSFP+ ports and 48 10-Gigabit Ethernet interfaces.
EX4600 device with Enhanced Layer 2 software	PIC 0 provides 4 40-Gbps QSFP+ ports and 24 10-Gigabit Ethernet interfaces. There are two expansion bays (PIC 1 and PIC 2), and you can insert QFX-EM-4Q expansion modules and EX4600-EM-8F expansion modules. The QFX-EM-4Q expansion module provide 4 40-Gbps QSFP+ ports. The EX4600-EM-8F expansion module provides 8 10-Gbps SFP+ ports. You can insert any combination of expansion modules. For example, you can insert two EX4600-EM-8F expansion modules, two QFX-EM-4Q expansion modules, or one of each.

Table 13: Naming Conventions for PICs *(Continued)*

Device with Software Package	Convention
QFX5100-48S switch with a QFabric software package	PIC **1** provides six 40-Gbps QSFP+ ports, and PIC **0** provides 48 10-Gigabit Ethernet interfaces.
QFX5100-24Q switch with Enhanced Layer 2 software	PIC **0** provides 24 40-Gbps QSFP+ ports. PIC 1 and PIC 2 can each contain a QFX-EM-4Q expansion module, and each expansion module provides 4 40-Gbps QSFP+ ports
QFX5100-96S switch with Enhanced Layer 2 software	PIC **0** provides 96 10-Gigabit Ethernet interfaces and 8 40-Gbps QSFP+ ports .
QFX5110-48S switch with Enhanced Layer 2 software	PIC **0** can support 48 10-Gigabit Ethernet ports labeled 0 through 47, and 4 QSFP28 ports labeled 48 through 51. Ports 0 through 47 support either 1-Gbps small form-factor pluggable (SFP) or 10-Gbps small form-factor pluggable plus (SFP+) transceivers. You can also use SFP+ DAC cables and 10-Gbps active optical cables (AOC) in any access port. The default 100-Gigabit Ethernet ports can be configured as 40-Gigabit Ethernet, and in this configuration can either operate as dedicated 40-Gigabit Ethernet ports or can be channelized to 4 independent 10-Gigabit Ethernet ports using copper or fiber breakout cables.
QFX5200-32C switch with Enhanced Layer 2 software	PIC **0** provides 32 QSFP28 ports. The 100-Gigabit Ethernet ports can be channelized to two 50-Gigabit Ethernet or four 25-Gigabit Ethernet ports. The default 100-Gigabit Ethernet ports can be configured as 40-Gigabit Ethernet and operate as 40-Gigabit Ethernet or be channelized to four 10-Gigabit Ethernet ports.
QFX10002-36Q switch with Enhanced Layer 2 software	PIC **0** provides 144 10-Gigabit Ethernet interfaces, and 36 40-Gbps QSFP+ ports, and 12 100-Gigabit Ethernet interfaces.
QFX10002-72Q switch with Enhanced Layer 2 software	PIC **0** provides 288 10-Gigabit Ethernet interfaces, and 72 40-Gbps QSFP+ ports, and 24 100-Gigabit Ethernet interfaces.
QFX10008 switch with Enhanced Layer 2 software	PIC **0** provides one-thousand, one-hundred fifty two 10-Gigabit Ethernet interfaces, two-hundred eighty-eight 40-Gbps QSFP+ ports, or two-hundred forty 100-Gigabit Ethernet interfaces.

Table 13: Naming Conventions for PICs *(Continued)*

Device with Software Package	Convention
QFX10016 switch with Enhanced Layer 2 software	PIC 0 provides two-thousand, three-hundred and four 10-Gigabit Ethernet interfaces, five-hundred seventy-six 40-Gbps QSFP+ ports, or four-hundred eighty 100-Gigabit Ethernet interfaces.

- *port*—Interfaces use the following convention for port numbers:

Table 14: Naming Conventions for PORTs

Device with Software Package	Convention
QFX3500 switch with a QFabric software package	There are 48 network access ports (10-Gigabit Ethernet) labeled 0 through 47 on PIC 0 and, 16 network access ports labeled 0 through 15 on PIC 1, and four 40-Gbps QSFP+ ports labeled Q0 through Q3 on PIC 2. You can use the QSFP+ ports to connect the Node device to Interconnect devices. By default, the 40-Gbps QSFP+ ports are configured to operate as 10-Gigabit Ethernet ports. You can use QSFP+ to four SFP+ copper breakout cables to connect the 10-Gigabit Ethernet ports to other servers, storage, and switches. Optionally, you can choose to configure the QSFP+ ports as 40-Gigabit Ethernet ports (see *Configuring the QSFP+ Port Type on QFX3500 Standalone Switches*).
QFX3500 switch with Enhanced Layer 2 software	There are 48 network access ports labeled 0 through 47 on PIC 0 and 4 40-Gbps QSFP+ ports labeled Q0 through Q3 on PIC 1. See *Channelizing Interfaces on QFX3500, QFX3600, QFX5100, QFX10002, QFX10008, QFX10016, and EX4600 Switches* for information on how to configure and channelize the 40-Gbps QSFP+ ports.

Table 14: Naming Conventions for PORTs *(Continued)*

Device with Software Package	Convention
QFX3600 switch with a QFabric software package	There are 64 network access ports (10-Gigabit Ethernet) labeled Q0 through Q15 on PIC 0, and there are 16 network access ports (40-Gigabit Ethernet) labeled Q0 through Q15 on PIC **1**. By default, all the QSFP+ ports are configured to operate as 40-Gigabit Ethernet ports. Optionally, you can choose to configure the QSFP+ ports as 10-Gigabit Ethernet ports (see *Configuring the Port Type on QFX3600 Standalone Switches*) and use QSFP+ to four SFP+ copper breakout cables to connect the 10-Gigabit Ethernet ports to other servers, storage, and switches.
QFX3600 Node device with a QFabric software package	PIC **0** can support up to 56 10-Gigabit Ethernet ports labeled Q2 through Q15, and PIC **1** can support up to 8 40-Gigabit data plane uplink ports labeled Q0 through Q7, and up to 14 40-Gigabit Ethernet ports labeled Q2 through Q15. On a QFX3600 Node device, by default, four 40-Gbps QSFP+ ports (labeled Q0 through Q3) are configured for uplink connections between your Node device and your Interconnect devices, and twelve 40-Gbps QSFP+ ports (labeled Q4 through Q15) use QSFP+ to four SFP+ copper breakout cables to support up to 48 10-Gigabit Ethernet ports for connections to either endpoint systems (such as servers and storage devices) or external networks. Optionally, you can choose to configure the first eight ports (Q0 through Q7) for uplink connections between your Node device and your Interconnect devices, and ports Q2 through Q15 for 10-Gigabit Ethernet or 40-Gigabit Ethernet connections to either endpoint systems or external networks (see *Configuring the Port Type on QFX3600 Node Devices*).
QFX3600 switch with Enhanced Layer 2 software	PIC **0** can support 64 network access ports (10-Gigabit Ethernet ports) labeled Q0 through Q15 and 16 40-Gigabit Ethernet ports labeled Q0 through Q15. See *Channelizing Interfaces on QFX3500, QFX3600, QFX5100, QFX10002, QFX10008, QFX10016, and EX4600 Switches* for information on how to configure and channelize the 40-Gbps QSFP+ ports.
QFX5100-48S switch with Enhanced Layer 2 software	PIC **0** can support 48 network access ports (10-Gigabit Ethernet ports) labeled 0 through 47 and 6 40-Gbps QSFP+ ports labeled 48 through 53. See *Channelizing Interfaces on QFX3500, QFX3600, QFX5100, QFX10002, QFX10008, QFX10016, and EX4600 Switches* for information on how to configure and channelize the 40-Gbps QSFP+ ports.

Table 14: Naming Conventions for PORTs *(Continued)*

Device with Software Package	Convention
EX4600 switch with Enhanced Layer 2 software	PIC **0** can support 24 network access ports (10-Gigabit Ethernet ports) labeled 0 through 23 and 4 40-Gbps QSFP+ ports labeled 24 through 27. There are two expansion bays (PIC 1 and PIC 2), and you can insert QFX-EM-4Q expansion modules and EX4600-EM-8F expansion modules. The QFX-EM-4Q expansion module provide 4 40-Gbps QSFP+ ports. The EX4600-EM-8F expansion module provides 8 10-Gbps SFP+ ports. You can insert any combination of expansion modules. For example, you can insert two EX4600-EM-8F expansion modules, two QFX-EM-4Q expansion modules, or one of each. See *Channelizing Interfaces on QFX3500, QFX3600, QFX5100, QFX10002, QFX10008, QFX10016, and EX4600 Switches* for information on how to configure and channelize the 40-Gbps QSFP+ ports.
QFX5100-48S switch with a QFabric software package	PIC **0** can support 48 network access ports (10-Gigabit Ethernet ports) labeled 0 through 47, and PIC 1 can support 6 40-Gbps QSFP+ ports labeled 0 through 5. See *Configuring the QSFP+ Port Type on QFX5100 Devices* for information on how to configure the port mode of 40-Gbps QSFP+ ports.
QFX5100-24Q switch with Enhanced Layer 2 software	PIC **0** can support 24 40-Gbps QSFP+ ports labeled 0 through 23. PIC 1 and PIC 2 each support 4 40-Gbps QSFP+ port, for a total of eight 40-Gbps QSFP + ports. See *Channelizing Interfaces on QFX3500, QFX3600, QFX5100, QFX10002, QFX10008, QFX10016, and EX4600 Switches* for information on how to configure and channelize the 40-Gbps QSFP+ ports. **NOTE**: You cannot channelize the 40-Gbps QSFP+ ports provided in the two QFX-EM-4Q expansion modules. Also, even though there is a total of 128 physical ports, only 104 logical ports can be channelized. You can configure different system modes to achieve varying levels of port density on the QFX5100-24Q and QFX5100-96S switches. Depending on the system mode you configure, there are restrictions on which ports you can channelize. If you channelize ports that are restricted, the configuration is ignored. See *Configuring the System Mode* for information on how to configure the system mode.

Table 14: Naming Conventions for PORTs *(Continued)*

Device with Software Package	Convention
QFX5100-96S switch with Enhanced Layer 2 software	PIC **0** can support 96 10-Gigabit Ethernet ports labeled 0 through 95, and 8 40-Gbps QSFP+ ports labeled 96 through 103. See *Channelizing Interfaces on QFX3500, QFX3600, QFX5100, QFX10002, QFX10008, QFX10016, and EX4600 Switches* for information on how to configure and channelize the 40-Gbps QSFP+ ports. **NOTE**: You can only channelize the 40-Gbps QSFP+ ports provided in ports 96 and 100, because only 104 logical ports can be channelized. You can configure different system modes to achieve varying levels of port density on the QFX5100-24Q and QFX5100-96S switches. Depending on the system mode you configure, there are restrictions on which ports you can channelize. If you channelize ports that are restricted, the configuration is ignored. See *Configuring the System Mode* for information on how to configure the system mode.
QFX5110-48S switch with Enhanced Layer 2 software	PIC **0** can support 48 10-Gigabit Ethernet ports labeled 0 through 47, and 4 QSFP28 ports labeled 48 through 51. These data ports (0 through 47) support either 1-Gbps small form-factor pluggable (SFP) or 10-Gbps small form-factor pluggable plus (SFP+) transceivers. You can also use SFP+ DAC cables and 10-Gbps active optical cables (AOC) in any access port.The default 100-Gigabit Ethernet ports can be configured as 40-Gigabit Ethernet, and in this configuration can either operate as dedicated 40-Gigabit Ethernet ports or can be channelized to 4 independent 10-Gigabit Ethernet ports using copper or fiber breakout cables.
QFX5200-32C switch with Enhanced Layer 2 software	There is support for both quad small-form-factor pluggable (QSFP+) and 28-Gbps QSFP+ (QSFP28) transceivers in the 32 QSFP28 sockets. The QSFP28 ports are configured as 100-Gigabit Ethernet ports by default, but can also be configured to speeds of 50, 40, 25, or 10 Gigabit Ethernet. The 100 Gigabit Ethernet ports can be channelized using breakout cables either to 2 independent downstream 50 Gigabit Ethernet or to 4 independent 25 Gigabit Ethernet ports. The default 100 Gigabit Ethernet ports can also be configured as 40 Gigabit Ethernet and in this configuration can either operate as dedicated 40 Gigabit Ethernet ports or can be channelized to 4 independent 10 Gigabit Ethernet ports using breakout cables. See *Channelizing Interfaces on QFX5200-32C Switches* for information on how to configure and channelize the interfaces.

Table 14: Naming Conventions for PORTs *(Continued)*

Device with Software Package	Convention
QFX10002-36Q switch with Enhanced Layer 2 software	There are 36 quad small-form factor pluggable plus (QSFP+) ports that support 40-Gigabit Ethernet optical transceivers. Out of these 36 ports, 12 ports are QSFP28 capable, which are dual speed 40- or 100-Gigabit Ethernet optical transceivers. Each QSFP28 socket can be configured to support: • 100-Gigabit Ethernet using 28-Gbps QSFP28 optical transceivers. When a QSFP28 transceiver is inserted into the ports marked with a fine black line underneath the socket and the port is configured for 100-Gigabit Ethernet, the two adjacent ports are disabled and the QSFP28 is enabled for 100-Gigabit Ethernet. • 40-Gigabit Ethernet using QSFP+ optical transceivers. • 10-Gigabit Ethernet using breakout cables. When configured for channelization, a breakout cable converts the 40-Gigabit Ethernet port into 4 independent 10-Gigabit Ethernet ports. Any of the 36 ports 0 through 35 can be configured as either uplink or access ports. See *Channelizing Interfaces on QFX3500, QFX3600, QFX5100, QFX10002, QFX10008, QFX10016, and EX4600 Switches* for information on how to configure and channelize the 40-Gbps QSFP+ ports. Each of the 12 QSFP28 ports support: • 100-Gigabit Ethernet QSFP28 transceivers • 40-Gigabit Ethernet QSFP+ transceivers Each of the 36 QSFP+ ports support: • 40-Gigabit Ethernet QSFP+ transceivers • Access ports

Table 14: Naming Conventions for PORTs *(Continued)*

Device with Software Package	Convention
QFX10002-72Q switch with Enhanced Layer 2 software	There are 72 quad small-form factor pluggable plus (QSFP+) ports that support 40-Gigabit Ethernet optical transceivers. Out of these 72 ports, 24 ports are QSFP28 capable, which are dual speed 40- or 100-Gigabit Ethernet optical transceivers. Each QSFP28 socket can be configured to support: • 100-Gigabit Ethernet using 28-Gbps QSFP28 optical transceivers. When a QSFP28 transceiver is inserted into the ports marked with a fine black line underneath the socket and the port is configured for 100-Gigabit Ethernet, the two adjacent ports are disabled and the QSFP28 is enabled for 100-Gigabit Ethernet. • 40-Gigabit Ethernet using QSFP+ optical transceivers. • 10-Gigabit Ethernet using breakout cables. When configured for channelization, a breakout cable converts the 40-Gigabit Ethernet port into 4 independent 10-Gigabit Ethernet ports. Any of the 72 ports 0 through 71 can be configured as either uplink or access ports. See *Channelizing Interfaces on QFX3500, QFX3600, QFX5100, QFX10002, QFX10008, QFX10016, and EX4600 Switches* for information on how to configure and channelize the 40-Gbps QSFP+ ports. Each of the 24 QSFP28 ports support: • 100-Gigabit Ethernet QSFP28 transceivers Each of the 72 QSFP+ ports support: • 40-Gigabit Ethernet QSFP+ transceivers Each of the 36 QSFP+ ports support: • 40-Gigabit Ethernet QSFP+ transceivers • Access ports • Uplink ports

Table 14: Naming Conventions for PORTs *(Continued)*

Device with Software Package	Convention
On a QFX10008 switch with Enhanced Layer 2 software, there are two line cards available: QFX10008 with Line Card QFX10000-36Q (ELS)	QFX10000-36Q, a 36-port 40-Gigabit Ethernet quad small form-factor pluggable plus transceiver (QSFP+) or 12-port 100GbE QSFP28 line card The QFX10000-36Q line cards supports Each QSFP28 socket can be configured to support: • 100-Gigabit Ethernet using QSFP28 optical transceivers. When a QSFP28 transceiver is inserted into the ports marked with a fine black line underneath the socket and the port is configured for 100-Gigabit Ethernet, the two adjacent ports are disabled and the QSFP28 socket is enabled for 100-Gigabit Ethernet. • 40-Gigabit Ethernet using QSFP+ optical transceivers. • 10-Gigabit Ethernet using breakout cabling and attached optical transceivers. When configured for channelization, the system converts the 40-Gigabit Ethernet port into 4 independent 10-Gigabit Ethernet ports. Any of the 36 ports 0 through 35 can be configured as either uplink or access ports. See *Channelizing Interfaces on QFX3500, QFX3600, QFX5100, QFX10002, QFX10008, QFX10016, and EX4600 Switches* for information on how to configure and channelize the 40-Gbps QSFP+ ports. Each of the 12 QSFP28 ports supports: • 100-Gigabit Ethernet QSFP28 transceivers • 40-Gigabit Ethernet QSFP+ transceivers Each of the 12 QSFP28 ports supports: • 100-Gigabit Ethernet QSFP28 transceivers • 40-Gigabit Ethernet QSFP+ transceivers Each of the 36 QSFP+ ports support: • 40-Gigabit Ethernet QSFP+ transceivers

Table 14: Naming Conventions for PORTs *(Continued)*

Device with Software Package	Convention
	- Access ports - Uplink ports
QFX10008 with Line Card QFX10000-30C and QFX10000-30C-M (ELS)	QFX10000-30C and QFX10000-30C-M, a 30-port 100-Gigabit or 40-Gigabit Ethernet QSFP28 line card - The QFX10000-30C and QFX10000-30C-M line cards support: Thirty 28-Gbps QSFP+ Pluggable Solution (QSFP28) cages that support either 40-Gigabit Ethernet or 100-Gigabit Ethernet optical transceivers. The QFX10000-30C and QFX10000-30C-M ports auto detect the type of transceiver installed and set the configuration to the appropriate speed. Each QSFP28 socket can be configured to support: - 100-Gigabit Ethernet using QSFP28 optical transceivers. When a QSFP28 transceiver is inserted into the ports marked with a fine black line underneath the socket and the port is configured for 100-Gigabit Ethernet, the two adjacent ports are disabled and the QSFP28 socket is enabled for 100-Gigabit Ethernet. - 40-Gigabit Ethernet using QSFP+ optical transceivers. See *Channelizing Interfaces on QFX3500, QFX3600, QFX5100, QFX10002, QFX10008, QFX10016, and EX4600 Switches* for information on how to configure and channelize the 40-Gbps QSFP+ ports. Each of the 30 QSFP28 ports supports: - 100-Gigabit Ethernet QSFP28 transceivers - 40-Gigabit Ethernet QSFP+ transceivers - Access ports - Uplink ports

Table 14: Naming Conventions for PORTs *(Continued)*

Device with Software Package	Convention
On a QFX10016 switch running Enhanced Layer 2 software, there are 16 slots, which you can populate with two types line cards: QFX10016 with Line Card QFX10000-36Q (ELS)	• QFX10000-36Q, a 36-port 40-Gigabit Ethernet quad small form-factor pluggable plus transceiver (QSFP+) or 12-port 100GbE QSFP28 line card The QFX10000-36Q line card consists of 36 quad small form-factor pluggable plus (QSFP+) ports that support 40-Gigabit Ethernet optical transceivers. Out of these 36 ports, 12 ports are QSFP28 capable. The QSFP+ ports are dual speed and can support either 40-Gigabit or 100-Gigabit Ethernet optical transceivers. The line card can support 10-Gigabit Ethernet by channelizing the 40-Gigabit ports. Channelization is supported on fiber break-out cable using standard structured cabling techniques. With 100-Gigabit Ethernet using QSFP28 optical transceivers, when a QSFP28 transceiver is inserted into the ports marked with a fine black line underneath the socket and the port is configured for 100-Gigabit Ethernet, the two adjacent ports are disabled and the QSFP28 socket is enabled for 100-Gigabit Ethernet. You can use 40-Gigabit Ethernet using QSFP+ optical transceivers. With 10-Gigabit Ethernet using breakout cabling and attached optical transceivers, when configured for channelization, the system converts the 40-Gigabit Ethernet port into 4 independent 10-Gigabit Ethernet ports. Any of the 36 ports 0 through 35 can be configured as either uplink or access ports. Each of the 12 QSFP28 ports supports: • 100-Gigabit Ethernet QSFP28 transceivers • 40-Gigabit Ethernet QSFP+ transceivers Each of the 36 QSFP+ ports supports: • 40-Gigabit Ethernet QSFP+ transceivers • Access ports You can use 40-Gigabit Ethernet QSFP+ transceivers in any downstream port.

Table 14: Naming Conventions for PORTs *(Continued)*

Device with Software Package	Convention
	- Uplink ports You can configure all the QSFP+ ports as uplinks. Every second and sixth port in a 6XQSFP cage on a QFX10000-36Q supports 100-Gigabit Ethernet using QSFP28 transceivers. These 100-Gigabit Ethernet ports work either as 100-Gigabit Ethernet or as 40-Gigabit Ethernet, but are recognized as 40-Gigabit Ethernet by default. When a 40-Gigabit Ethernet transceiver is inserted into a 100-Gigabit Ethernet port, the port recognizes the 40-Gigabit Ethernet port speed. When a 100-Gigabit Ethernet transceiver is inserted into the port and enabled in the CLI, the port recognizes the 100-Gigabit Ethernet speed and disables two adjacent 40-Gigabit Ethernet ports. You can also use an 100-Gigabit Ethernet transceiver and run it at 40-Gigabit Ethernet by using the CLI to set the port speed to 40-Gigabit Ethernet. The 40-Gigabit Ethernet ports can operate independently, be channelized into four 10-Gigabit Ethernet ports, or bundled with the next two consecutive ports and channelized into twelve 10-Gigabit Ethernet ports as a port range. Only the first and fourth port in each 6XQSFP cage are available to channelize a port range. The port range must be configured using the set chassis fpc pic port channel-speed command. For example, to channelize the first switch port, use the `set chassis fpc 0 pic 0port 1channel-speed 10g` command.

Table 14: Naming Conventions for PORTs *(Continued)*

Device with Software Package	Convention
QFX10016 with Line Card QFX10000-30C and QFX10000-30C-M (ELS)	QFX10000-30C and QFX10000-30C-M, a 30-port 100-Gigabit or 40-Gigabit Ethernet QSFP28 line card. The QFX10000-30C and QFX10000-30C-M ports auto detect the type of transceiver installed and set the configuration to the appropriate speed. Each QSFP28 socket supports: • 100-Gigabit Ethernet using QSFP28 optical transceivers. When a QSFP28 transceiver is inserted into any of the ports, the QSFP28 socket is enabled for 100-Gigabit Ethernet. • 40-Gigabit Ethernet using QSFP+ optical transceivers. When a QSFP+ transceiver is inserted into any of the ports, the QSFP+ socket is enabled for 40-Gigabit. Any of the 30 ports 0 through 29 can be configured as either uplink or access ports, and of the 30 QSFP28 ports supports: • 100-Gigabit Ethernet QSFP28 transceivers • 40-Gigabit Ethernet QSFP+ transceivers

Logical Part of an Interface Name on a Switch Running QFabric Software Package for QFX series, NFX Series, EX4600, QFabric System

The logical unit part of the interface name corresponds to the logical unit number, which can be a number from 0 through 16384. In the virtual part of the name, a period (.) separates the port and logical unit numbers: *device-name* (QFabric systems only): *type-fpc/pic/port.logical-unit-number*. For example, if you issue the **show ethernet-switching interfaces** command on a system with a default VLAN, the resulting display shows the logical interfaces associated with the VLAN:

```
Interface                State   VLAN members          Blocking
node-device1:xe-0/0/1.0  down     remote-analyzer       unblocked
node-device1:xe-0/0/2.0  down     default               unblocked
node-device1:xe-0/0/3.0  down     default               unblocked
```

When you configure aggregated Ethernet interfaces, you configure a *logical interface*, which is called a or a . Each LAG can include up to eight Ethernet interfaces, depending on the switch model.

Logical Part of a Channelized Interface Name on a Switch Running Enhanced Layer 2 Software for QFX series, NFX Series, EX4600, QFabric System

Channelizing enables you to configure four 10-Gigabit Ethernet interfaces from a 40-Gigabit Ethernet QSFP+ interface. By default, a 40-Gigabit Ethernet QSFP+ interface is named *et-fpc/pic/port*. The resulting 10-Gigabit Ethernet interfaces appear in the following format: *xe-fpc/pic/port:channel*, where channel can be a value of 0 through 3.

For example, if an *et* interface named `et-0/0/3` is channelized to four 10-Gigabit Ethernet interfaces, the resulting 10-Gigabit Ethernet interface names will be `xe-0/0/3:0`, `xe-0/0/3:1`, `xe-0/0/3:2`, and `xe-0/0/3:3`:

```
Interface               Admin Link Proto    Local        Remote
xe-0/0/3:0               up    down
xe-0/0/3:1               up    down
xe-0/0/3:2               up    down
xe-0/0/3:3               up    down
```

Wildcard Characters in Interface Names for QFX series, NFX Series, EX4600, QFabric System

In the **show interfaces** and **clear interfaces** commands, you can use wildcard characters in the ***interface-name*** option to specify groups of interface names without having to type each name individually. You must enclose all wildcard characters except the asterisk (*) in square brackets [ ].

Physical Part of an Interface Name for OCX1100

Interfaces in Junos OS are specified as follows:

type-fpc/pic/port

The convention is as follows:

- *type*—The OCX Series device interfaces use the following media types:
  - **xe**—10-Gigabit Ethernet interface
  - **et**—40-Gigabit Ethernet interface
  - **em**—Management interface
- *fpc*—Flexible PIC Concentrator. OCX Series interfaces use the following convention for the FPC number in interface names:

  - On standalone OCX Series switches, the FPC number is always **0**.

    The FPC number indicates the slot number of the line card that contains the physical interface.

- *pic*—The OCX Series interfaces use the following convention for the PIC (*Physical Interface Card*) number in interface names:

  - PIC **0** provides six 40-Gbps QSFP+ ports and 48 10-Gigabit Ethernet interfaces.

- *port*—Interfaces use the following convention for port numbers:

  - PIC **0** can support 48 network access ports (10-Gigabit Ethernet ports) labeled 1 through 48 and 6 40-Gbps QSFP+ ports labeled 49 through 54.

Wildcard Characters in Interface Names for OCX1100

In the **show interfaces** and **clear interfaces** commands, you can use wildcard characters in the ***interface-name*** option to specify groups of interface names without having to type each name individually. You must enclose all wildcard characters except the asterisk (*) in square brackets [ ].

SEE ALSO

Interfaces Overview for Switches

Channelizing Interfaces on QFX3500, QFX3600, QFX5100, QFX10002, QFX10008, QFX10016, and EX4600 Switches

Understanding Management Interfaces

Understanding Port Ranges and System Modes

Configuring the System Mode

Configuring Gigabit Ethernet Interfaces (CLI Procedure)

Configuring Gigabit Ethernet Interfaces for EX Series Switches with ELS support

Junos OS Network Interfaces Library for Routing Devices

Rear Panel of a QFX3500 Device

Front Panel of a QFX3600 Device

Understanding Management Interfaces

You use management interfaces to access devices remotely. Typically, a management interface is not connected to the in-band network, but is connected to a device in the internal network. Through a management interface, you can access the device over the network using utilities such as **ssh** and **telnet** and configure it from anywhere, regardless of its physical location. As a security feature, users cannot log in as **root** through a management interface. To access the device as **root**, you must use the console port. You can also use **root** to log in using SSH.

> **NOTE**: Before you can use management interfaces, you must configure the logical interfaces with valid IP addresses. Juniper Networks does not support configuring two management interfaces in the same subnet.

Management interface port ranges vary based on device type (and platform support depends on the Junos OS release in your installation):

- QFX3500 devices:

  The valid port range for a management interface (**me**) on a QFX3500 device is between 0 and 6, with a total of seven available ports. On a QFX3500 standalone switch, however, you can only configure me0 and me1 as management interfaces. The management interfaces are labeled **C0** and **C1**, and they correspond to me0 and me1. On a QFX3500 Node device, the RJ-45 management interfaces and SFP management interfaces correspond to **me5** and **me6**

- QFX3600 devices:

  There are two RJ-45 management interfaces (labeled **C0** and **C1**) and two SFP management interfaces (labeled **C0S** and **C1S**). On a QFX3600 standalone switch, the RJ-45 management interfaces and SFP management interfaces correspond to **me0** and **me1**. On a QFX3600 Node device, the RJ-45 management interfaces and SFP management interfaces correspond to **me5** and **me6**. Each pair of management interfaces correspond to one Ethernet interface—for example, both RJ-45 management interfaces (labeled **C0** and **C0s**) can correspond to me0, and both SFP management interfaces (labeled **C1** and **C1S**) can correspond to me1. By default, both RJ-45 management interfaces are active. If you insert an SFP interface into the SFP management port (**C0S**, for example), the SFP interface would become the active management interface, and the corresponding RJ-45 management interface (**C0**) is disabled.

  > **NOTE**: On a QFX3600 device, you can use either the RJ-45 or the SFP management interfaces, but not both at the same time.

- On QFX5100, QFX5200, and EX4600 switches, there is one RJ-45 management interface (labeled **C0** and one SFP management interface (labeled **C1**), and they correspond to em0 and em1. You can use both management interfaces simultaneously.

- On QFX10002 and QFX10008 switches, there is one RJ-45 management interface (labeled **MGMT** and one SFP management interface (labeled **MGMT**), and they correspond to em0 and em1. Although the CLI permits you to configure two management Ethernet interfaces within the same subnet, only one interface is usable and supported.

- On QFX10008 and QFX10016 switches, if you are using em1 for management purpose, then you cannot directly access the backup RE em1 from external network. Indirectly you can access the backup RE from external network, by following these steps:

  - Login to primary RE using SSH/Telnet to its em1.

  - Access backup RE using the following command:

  ```
  user@host>request routing-engine login other-routing-engine
  ```

- On OCX Series switches:

  There is one RJ-45 management interface (labeled **MGMT**), which corresponds to em0. The em0 interface always has the status `up` in show command outputs, even if the physical port is empty. The me0 interface is a virtual interface between Junos and the host operating system, therefore its status is independent from the status of the physical port.

- QFabric system:

  On a QFabric system, there are management interfaces on the Node devices, Interconnect devices, and Director devices. However, you cannot access the management interfaces on the Node devices or Interconnect devices directly. You can only manage and configure these devices using the Director device. You can connect to the management interface over the network using utilities such as SSH.

  For information on how to use management interfaces on a QFabric system, see *Performing the QFabric System Initial Setup on a QFX3100 Director Group* and *Gaining Access to the QFabric System Through the Default Partition*.

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

- MTU Overview | 146
- Media MTU Overview | 146
- Configure the Media MTU | 147
- Protocol MTU | 148

- Encapsulation Overhead by Interface Encapsulation Type | 149
- Media MTU Sizes by Interface Type | 151
- MTU and MACsec | 157
- Platform-Specific MTU Behavior | 161

MTU Overview

A maximum transmission unit (MTU) is the largest data unit that can be forwarded on a link without fragmentation. If a packet exceeds the MTU for the interface or protocol it passes through, the device fragments the packet. When a packet is larger than the MTU, the device either drops the packet or fragments it and transmits the fragments. Fragmentation slows down the network and can lead to packet loss.

Some protocols such as IS-IS do not support fragmentation. With these protocols, if a packet exceeds the MTU for a link, the device drops the packet.

Configure the media MTU for a physical interface and the MTU for a protocol to avoid packet loss and optimize traffic over your network.

Use Feature Explorer to confirm platform and release support for specific features.

Review the "Platform-Specific MTU Behavior" on page 161 section for notes related to your platform.

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 Layer 3 (L3) MTU. In some cases, the L3 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 depends on the Layer 2 (L2) overhead and is calculated as follows:

```
Default media MTU = Default protocol MTU + L2 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.
- A SONET/SDH interface operating in concatenated mode has a “c” added to the rate descriptor. For example, a concatenated OC48 interface is referred to as OC48c.
- 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:

- Generic routing encapsulation (gr-)
- IP-IP (ip-)
- Loopback (lo-)
- Link services (ls-)
- Multilink services (ml-)
- Multicast (pe-, pd-)

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. Review the "Platform-Specific MTU Behavior" on page 161 section for notes related to your platform.

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

- Overview | **148**
- Protocol MTU for MPLS | **149**

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.

You can configure the protocol MTU on all tunnel interfaces except virtual tunnel (VT) interfaces. Junos OS sets the MTU size for VT interfaces to unlimited by default.

Protocol MTU for MPLS

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

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 15: Encapsulation Overhead by Encapsulation Type

Interface Encapsulation	Encapsulation Overhead (Bytes)
802.1Q/Ethernet 802.3	21
802.1Q/Ethernet Subnetwork Access Protocol (SNAP)	26
802.1Q/Ethernet version 2	18
ATM Cell Relay	4
ATM permanent virtual connection (PVC)	12
Cisco HDLC	4

Table 15: Encapsulation Overhead by Encapsulation Type *(Continued)*

Interface Encapsulation	Encapsulation Overhead (Bytes)
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

Media MTU Sizes by Interface Type

IN THIS SECTION

- Media MTU Sizes by Interface Type for M7i and M10i Routers with CFEB | 151
- Media MTU Sizes by Interface Type for M7i Routers with CFEB-E, M10i Routers with CFEB-E, and M320 and M120 Routers | 152
- Media MTU Sizes for MX Series Routers | 154
- Media MTU Sizes by Interface Type for ACX Series Routers and EX and QFX Series Switches | 156
- Media MTU Sizes by Interface Type for PTX Series Packet Transport Routers | 156
- Media MTU Sizes by Interface Type for JRR200 Series Routers | 157

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. Use this topic to determine the MTU values you can configure on your device.

Media MTU Sizes by Interface Type for M7i and M10i Routers with CFEB

Table 16: Media MTU Sizes by Interface Type for M7i and M10i Routers with CFEB

Interface Type	Default Media MTU (Bytes)	Maximum MTU (Bytes)	Default IP Protocol MTU (Bytes)
Adaptive Services (MTU size not configurable)	9192	N/A	N/A
ATM	4482	9192	4470
E1/T1	1504	9192	1500
E3/T3	4474	9192	4470

Table 16: Media MTU Sizes by Interface Type for M7i and M10i Routers with CFEB *(Continued)*

Interface Type	Default Media MTU (Bytes)	Maximum MTU (Bytes)	Default IP Protocol MTU (Bytes)
Fast Ethernet	1514	1533 (4-port) 1532 (8-port) 1532 (12-port) **NOTE**: The maximum MTU for two 100Base-TX Fast Ethernet port FIC is 9192 bytes.	1500 (IPv4), 1497 (ISO)
Gigabit Ethernet	1514	9192 **NOTE**: The maximum MTU for one Gigabit Ethernet port FIC is 9192 bytes.	1500 (IPv4), 1497 (ISO)
Serial	1504	9192	1500 (IPv4), 1497 (ISO)
SONET/SDH	4474	9192	4470

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

Table 17: Media MTU Sizes by Interface Type for M7i Routers with CFEB-E, M10i Routers with CFEB-E, and M320 and M120 Routers

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

Table 17: Media MTU Sizes by Interface Type for M7i Routers with CFEB-E, M10i Routers with CFEB-E, and M320 and M120 Routers *(Continued)*

Interface Type	Default Media MTU (Bytes)	Maximum MTU (Bytes)	Default IP Protocol MTU (Bytes)
Channelized E1 IQ	1504	4500	1500
Channelized OC12 IQ	4474	9192	4470
Channelized STM1 IQ	4474	9192	4470
DS3	4471	4500	4470
E1	1504	4500	1500
E3 IQ	4471	4500	4470
Fast Ethernet	1514	1533 (4-port) 1532 (8-, 12- and 48-port)	1500 (IPv4), 1497 (ISO)
Gigabit Ethernet	1514	9192	1500 (IPv4), 1497 (ISO)
SONET/SDH	4474	9192	4470
T1	1504	4500	1500
CT3 IQ (excluding M120)	4474	9192	4470

Media MTU Sizes for MX Series Routers

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

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

Table 19: Media MTU Sizes for MX Series Routers by MPC

MPC	Maximum MTU (Bytes)
MPC1	9500 (Ethernet interfaces)
MPC2	9500 (Ethernet interfaces)
MPC2E	9500 (Ethernet interfaces)
MPC2E-NG. MPC3E-NG	9500
MPC3E	9500 (Ethernet interfaces)

Table 19: Media MTU Sizes for MX Series Routers by MPC *(Continued)*

MPC	Maximum MTU (Bytes)
MPC4E	9500 (Ethernet interfaces)
MPC5E	9500 (Ethernet interfaces)
MPC6E	9500 (Ethernet interfaces)
MPC7E (MPC7E-MRATE and MP7E-10G)	16,000
MPC8E (MX2K-MPC8E)	16,000
MPC9E (MX2K-MPC9E)	16,000
MPC10E-10C-MRATE	16,000 (Junos OS Release 19.2R1 and later)
MPC10E-15C-MRATE	16,000 (Junos OS Release 19.1R1 and later)
MX2K-MPC11E	16,000 (Junos OS Release 19.3R2 and later)
MX10003 MPC (MX10003-LC2103)	16,000 (Junos OS Release 17.3R1 and later)

Table 20: Media MTU Sizes for MX Series Routers by Platform

Platform	Maximum MTU (Bytes)
MX5, MX10, MX40, MX80	9192
MX204	16,000 (Junos OS Release 17.4R1 and later)
MX304	16000
MX10000	16000

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

Table 21: Media MTU Sizes by Interface Type for ACX Series Routers

Interface Type	Switch	Default Media MTU (Bytes)	Maximum MTU (Bytes)	Default IP Protocol MTU (Bytes)
Gigabit Ethernet and 10-Gigabit Ethernet	ACX1000, ACX2000, ACX4000, ACX5048, ACX5096 line of routers, and ACX500	1514	9216	1500 (IPv4), 1497 (ISO)
Gigabit Ethernet and 10-Gigabit Ethernet	ACX5448 series and ACX710 Series	1514	10000	1500 (IPv4), 1497 (ISO)
Gigabit Ethernet and 10-Gigabit Ethernet	ACX7000 Series	1514	9996	1500 (IPv4), 1497 (ISO)

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

Table 22: Media MTU Sizes by Interface Type for PTX Series Packet Transport Routers

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

Table 22: Media MTU Sizes by Interface Type for PTX Series Packet Transport Routers *(Continued)*

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

Media MTU Sizes by Interface Type for JRR200 Series Routers

Table 23: Media MTU Sizes by Interface Type for JRR200 Series Routers

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

MTU and MACsec

IN THIS SECTION

- Overview of Automatic MTU Adjustment for MACsec | **158**
- Configure Automatic MTU Adjustment for MACsec | **158**
- Behavior of Automatic MTU Adjustment for MACsec | **159**

Media Access Control security (MACsec) is a Layer 2 (L2) security protocol that provides point-to-point security. MACsec adds a header to packets passing through interfaces where MACsec is enabled. If a packet is near the protocol MTU limit, and the MTU is not adjusted to account for the MACsec header, the packet can exceed the interface MTU when the MACsec header is added. In that case, the device drops the packet. Before enabling MACsec, you must ensure your protocol MTU is large enough to accommodate the additional 32 bytes of MACsec overhead.

NOTE: The MACsec header can be smaller than 32 bytes when there is no Secure Channel Identifier (SCI) field. We recommend assuming the MACsec header is 32 bytes to ensure the device transmits the MACsec packet.

Overview of Automatic MTU Adjustment for MACsec

This feature ensures the interface and protocol MTU are adjusted properly to account for the MACsec overhead when the MTU is left as the default. Without this feature, you (the network administrator) need to adjust the interface and protocol MTU manually.

When MACsec is enabled on a physical interface or a logical interface and a custom MTU has not been set, you can configure your device to automatically adjust the MTU to include the MACsec header for that interface. If the device is using the default interface MTU when this feature is enabled, the device automatically increases the interface MTU to accommodate the MACsec header. When MACsec is enabled on a specific logical interface, the protocol families under that logical interface use an adjusted MTU that accommodates the MACsec header.

This feature is not supported on aggregated Ethernet interfaces or link aggregation groups (LAGs) directly, but it is supported on physical interfaces that are members of aggregated Ethernet interfaces. If you enable MACsec on one member interface of an aggregated Ethernet interface, the device copies the automatically adjusted MTU to all members of the aggregated Ethernet interface. Note that the LAG flaps when you add or remove the only MACsec-enabled interface to or from a LAG.

Configure Automatic MTU Adjustment for MACsec

NOTE: When the media MTU or protocol MTU changes, even automatically, it causes an interface to be deleted and added again. This causes the link to flap.

Automatic MTU adjustment is disabled by default. To enable automatic MTU adjustment for MACsec:

1. Configure MACsec at both the `[edit interfaces` *`interface-name`*`]` and the `[edit security macsec interfaces` *`interface-name`*`]` hierarchy levels. See Configuring MACsec for more information.

2. Configure the `enable-auto-mtu-update` statement at the `[edit security macsec]` hierarchy level.

```
[edit]
user@device# set security macsec enable-auto-mtu-update
```

Behavior of Automatic MTU Adjustment for MACsec

Factors that affect the behavior of the MTU automatic adjustment include:

- Where MACsec is configured. MACsec can be configured at the physical interface (IFD) level or the logical interface (IFL) level.
- Whether the MTU is for an interface or for a protocol.
- For the protocol MTU, whether the protocol belongs to a Layer 2 (L2) or Layer 3 (L3) protocol family.

If you have manually configure the MTU, the device uses the configured MTU instead and does not automatically update the MTU. The following tables show how devices that support this feature automatically adjust the MTU when the MTU has not been configured.

Table 24: Automatic MTU Adjustment for MACsec for L3 Protocol Families

MACsec Enabled At:	IFD MTU Configured?	IFD MTU (in bytes)	Protocol MTU Configured?	Protocol MTU (in bytes)
Physical interface (IFD) level	No	IFD MTU + 32	No	(Adjusted IFD MTU) – (32 + L2 overhead)
Physical interface (IFD) level	No	IFD MTU + 32	Yes	Uses configured protocol MTU
Logical interface (IFL) level	No	IFD MTU remains unchanged.	No	(IFD MTU) – (32 + L2 overhead)
Logical interface (IFL) level	No	IFD MTU remains unchanged.	Yes	Uses configured protocol MTU

This feature functions differently for L2 protocol families such as CCC, VPLS, BRIDGE, or TCC:

Table 25: Automatic MTU Adjustment for MACsec for L2 Protocol Families (Junos OS)

MACsec Enabled At:	IFD MTU Configured?	Where L2 Protocol Is Configured	Protocol MTU Configured?	Protocol MTU Behavior
Physical interface (IFD) level	No	Any logical interface under the physical interface uses an L2 protocol	No	The device skips protocol MTU adjustment for all logical interfaces under that physical interface hierarchy.
Logical interface (IFL) level	No	Only the logical interface where MACsec is enabled uses an L2 protocol	No	The device skips MTU adjustment only for the protocol configured under that logical interface.

Table 26: Automatic MTU Adjustment for MACsec for L2 Protocol Families (Junos OS Evolved)

MACsec Enabled At:	IFD MTU Configured?	IFD MTU (in bytes)	Protocol MTU Configured?	Protocol MTU (in bytes)
Physical interface (IFD) level	No	IFD MTU + 32	No	Original IFD MTU (Adjusted IFD MTU - 32 = IFD MTU + 32 - 32)
Physical interface (IFD) level	No	IFD MTU + 32	Yes	Uses configured protocol MTU
Logical interface (IFL) level	No	IFD MTU remains unchanged.	No	Same as the IFD MTU
Logical interface (IFL) level	No	IFD MTU remains unchanged.	Yes	Uses configured protocol MTU

Platform-Specific MTU Behavior

Use Feature Explorer to confirm platform and release support for specific features.

Use the following table to review platform-specific behavior for your platform:

Platform	Difference
ACX Series	• ACX Series routers that support protocol MTU need to explicitly configure MTU at the family level for IPv4 and IPv6 make MTU exception work in egress. Follow the guidelines below while configuring MTUs. If you configure MTUs: • If you configure MTUs for both inet and inet6 families, inet MTU gets precedence. • If you configure MTU only at inet level, the same value applies to inet6 as well. • If you configure MTU only for inet6 level, the same value applies to inet as well.
MX Series	• MX304, MX960, MX2020, MX10003, MX10008: When MACsec is enabled on the interfaces of these devices, you can enable the device to automatically increase the MTU as described in the **Automatic MTU Adjustment for MACsec** section above. • MX204, MX240, MX301, MX304, MX480, MX960, MX2008, MX2010, MX2020, MX10004 and MX10008: When changing the media MTU or protocol MTU, the physical interface does NOT flap. However, all protocol sessions (such as BGP, OSPF, IS-IS, etc.) configured on that interface will flap as they are reset during the MTU change operation.

Change History Table

Feature support is determined by the platform and release you are using. Use Feature Explorer to determine if a feature is supported on your platform.

Release	Description
25.2R1	In Junos OS Release 25.2R1 and Junos OS Evolved 25.4R1, when MACsec is enabled on an interface, you can enable the device to automatically increase the MTU for that interface by configuring the `enable-auto-mtu-update` statement at the `[edit security macsec]` hierarchy level.
19.3R2	Starting in Junos OS Release 19.3R2, the maximum configurable MTU size for MX2K-MPC11E is 16,000 bytes.
19.2R1	Starting in Junos OS Release 19.2R1, the maximum configurable MTU size for MPC10E-10C-MRATE is 16,000 bytes.
19.1R1	Starting in Junos OS Release 19.1R1, the maximum configurable MTU size for MPC10E-15C-MRATE is 16,000 bytes.
17.4R1	Starting in Junos OS Release 17.4R1, the MTU size for MX204 is 16,000 bytes.
17.3R1	Starting in Junos OS Release 17.3R1, the MTU size for MX10003 MPC is 16,000 bytes.

Interface Ranges for Physical Interfaces

IN THIS SECTION

- Configure Interface Ranges | 163
- Expanded Interface Range Statements | 168
- Configuration Inheritance Priority | 170
- Configuration Inheritance for Member Interfaces | 171
- Common Configuration Inheritance | 172
- Configuration Group Inheritance | 173
- Configuration Expansion Where Interface Range Is Used | 175

Junos OS 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.

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

Configure Interface Ranges

IN THIS SECTION

- Supported Hierarchies | **166**

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 supports interface ranges for the following interface types:

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

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.

Supported Hierarchies

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

- `ethernet-switching-options analyzer` *`name`* `input [egress | ingress ] interface`
- `ethernet-switching-options analyzer` *`name`* `output interface`
- `ethernet-switching-options bpdu-block interface`
- `ethernet-switching-options interfaces ethernet-switching-options voip interface`
- `ethernet-switching-options redundant-trunk-group group g1 interface`
- `ethernet-switching-options secure-access-port interface`
- `poe interface vlans pro-bng-mc1-bsd1 interface`
- `protocols dot1x authentication interface`
- `protocols dvmrp interface`
- `protocols esis interface`

- `protocols gvrp interface`
- `protocols igmp interface`
- `protocols igmp-snooping vlan` *`name`* `interface`
- `protocols igmp-host client` *`num`* `interface`
- `protocols isis interface`
- `protocols layer2-control bpdu-block interface`
- `protocols layer2-control mac-rewrite interface`
- `protocols ldp interface`
- `protocols link-management peer control-channel`
- `protocols link-management peer lmp-control-channel interface`
- `protocols link-management te-link` *`name`* `interface`
- `protocols lldp interface`
- `protocols lldp-med interface`
- `protocols mld interface`
- `protocols mld-host client` *`num`* `interface`
- `protocols mpls interface`
- `protocols mstp interface`
- `protocols mstp msti` *`id`* `interface`
- `protocols mstp msti vlan` *`id`* `interface`
- `protocols oam ethernet link-fault-management interface`
- `protocols oam ethernet lmi interface`
- `protocols ospf area` *`id`* `interface`
- `protocols pim interface`
- `protocols rip group` *`name`* `neighbour`
- `protocols ripng group` *`name`* `neighbour`
- `protocols router-advertisement interface`

- `protocols router-discovery interface`
- `protocols rstp interface`
- `protocols rsvp interface`
- `protocols sflow interfaces`
- `protocols snmp interface`
- `protocols stp interface`
- `protocols vstp interface`
- `protocols vstp vlan` *`name`* `interface`

Expanded Interface Range Statements

The OS 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.

A sample 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 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 OS reads this configuration.

In this example, Junos OS 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;
     }
}
```

Verify 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;
}
```

SEE ALSO

Using Wildcards with Configuration Groups

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 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 | 176
- Configure Damping of Shorter Physical Interface Transitions | 184
- Configure Damping of Aggregated Ethernet Interface Transitions | 185
- Configure Damping of Longer Physical Interface Transitions | 186
- Example: Configure Physical Interface Damping | 188

Physical Interface Damping Overview

IN THIS SECTION

- Damping Overview for Shorter Physical Interface Transitions | 177
- Damping Overview for Longer Physical Interface Transitions | 178

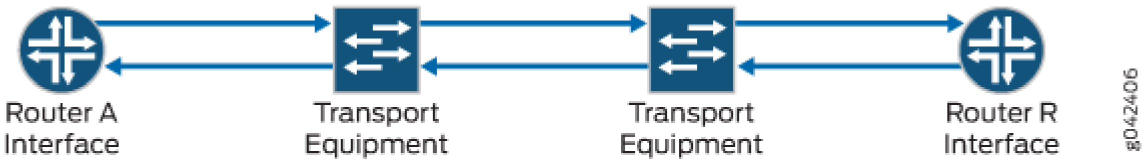
Physical interface damping limits the advertisement of the up-and-down transitions (flapping) on an interface. Each time a transition occurs, the interface state is changed, which generates an advertisement to the upper-level routing protocols. Damping helps reduce the number of these advertisements.

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

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

Figure 7 on page 177 is used to describe these types of interface flaps and the damping configuration that you can use in each case.

Figure 7: Two Router Interfaces Connected Through Transport Equipment



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.

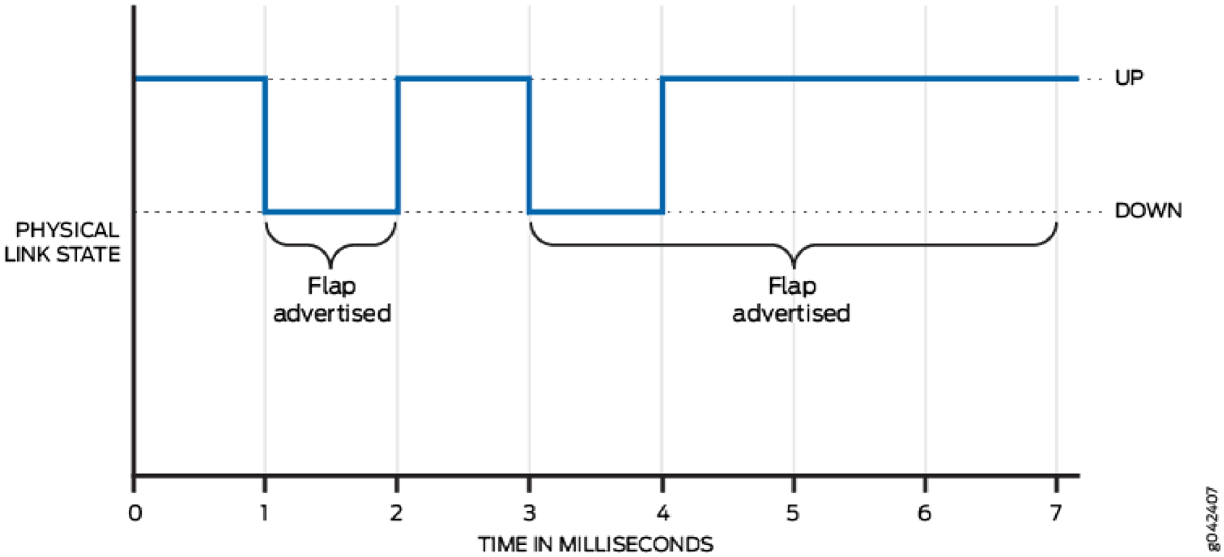
The types of interface damping depend upon the transition time length.

Damping Overview for Shorter Physical Interface Transitions

Figure 7 on page 177 shows two routers with two transport devices between them. If a redundant link between the two transport devices fails, Junos OS performs link switching. Link switching takes a number of milliseconds. As shown in Figure 8 on page 178, 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. 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.

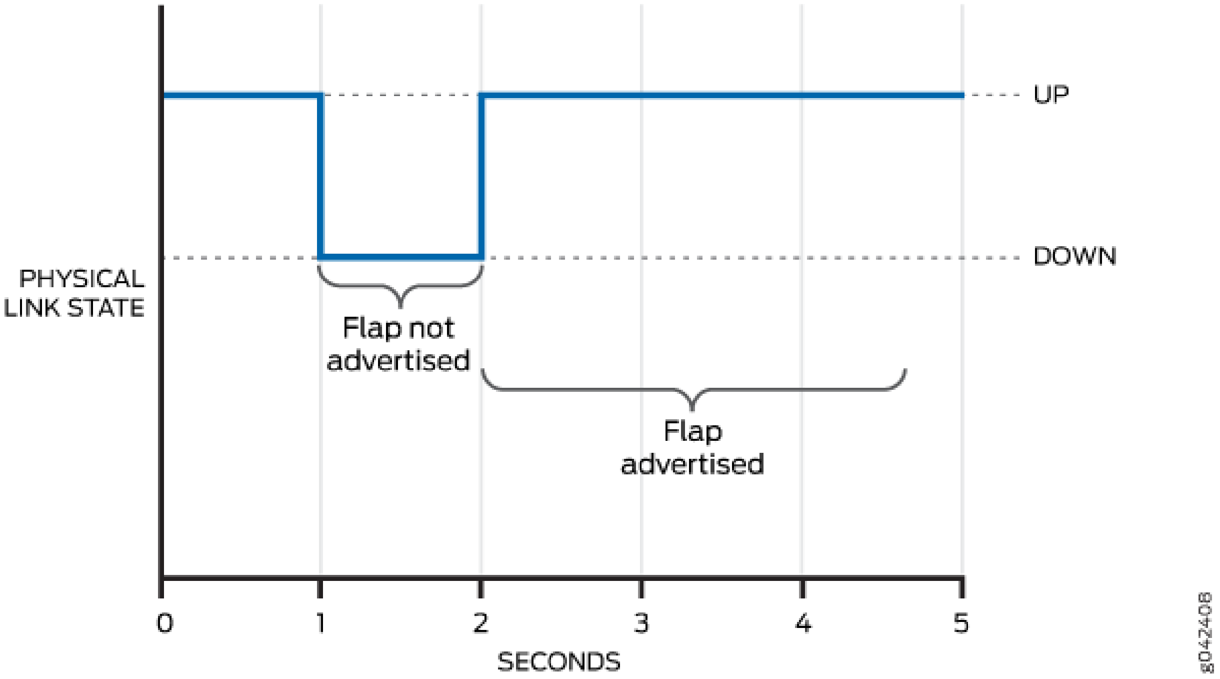
Figure 8: Multiple Flaps of Short Duration (Milliseconds)



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 9 on page 179. 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 9: 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 adds a penalty to the interface penalty counter. If at some point the accumulated penalty exceeds the suppress level, Junos OS places the interface in the suppress state. In this case, Junos OS 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 10 on page 181 and Figure 11 on page 183 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 10 on page 181 and Figure 11 on page 183 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 10 on page 181 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 10: Physical-Level Link Is Down When the Penalty Falls Below the Reuse Level

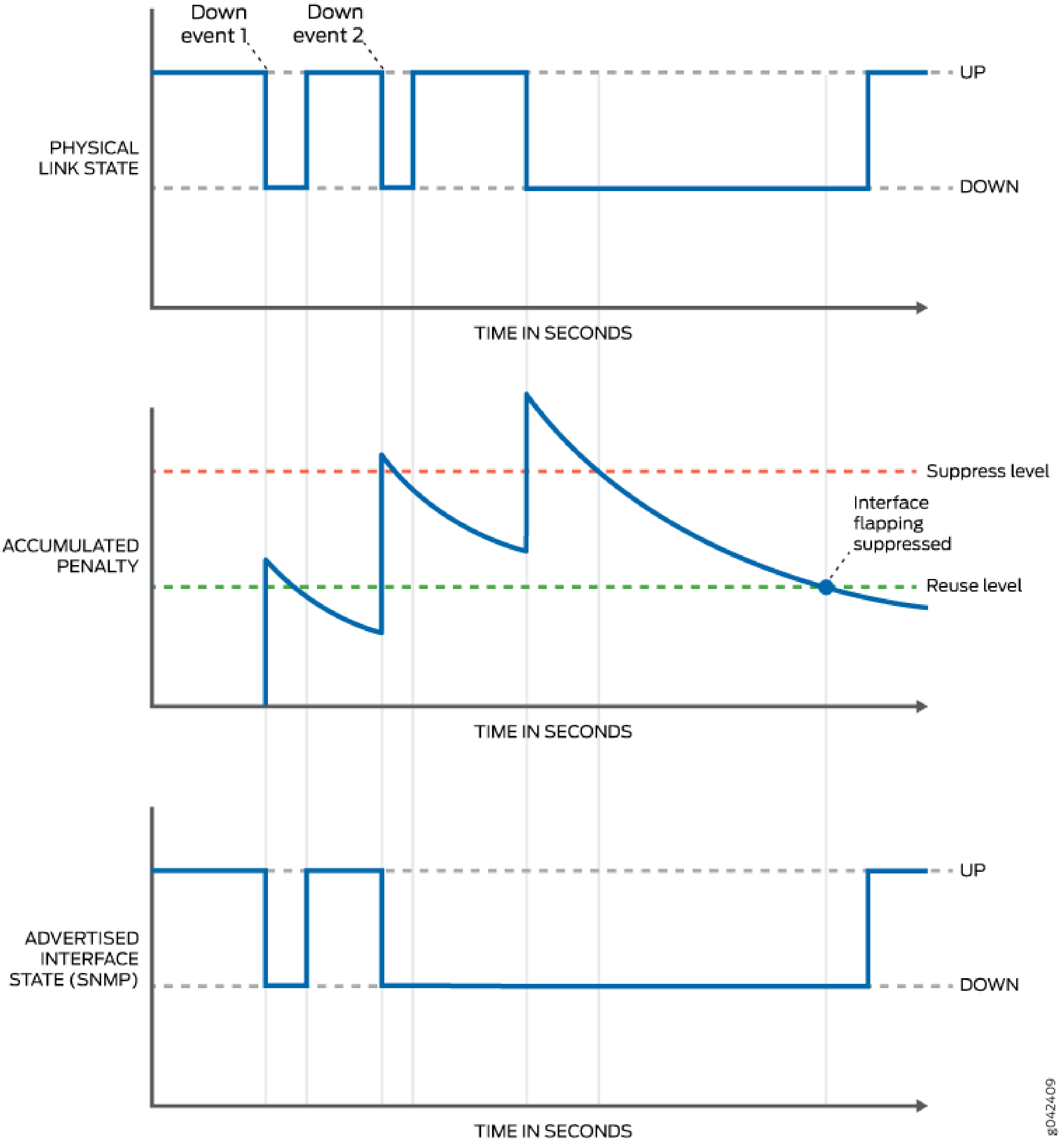
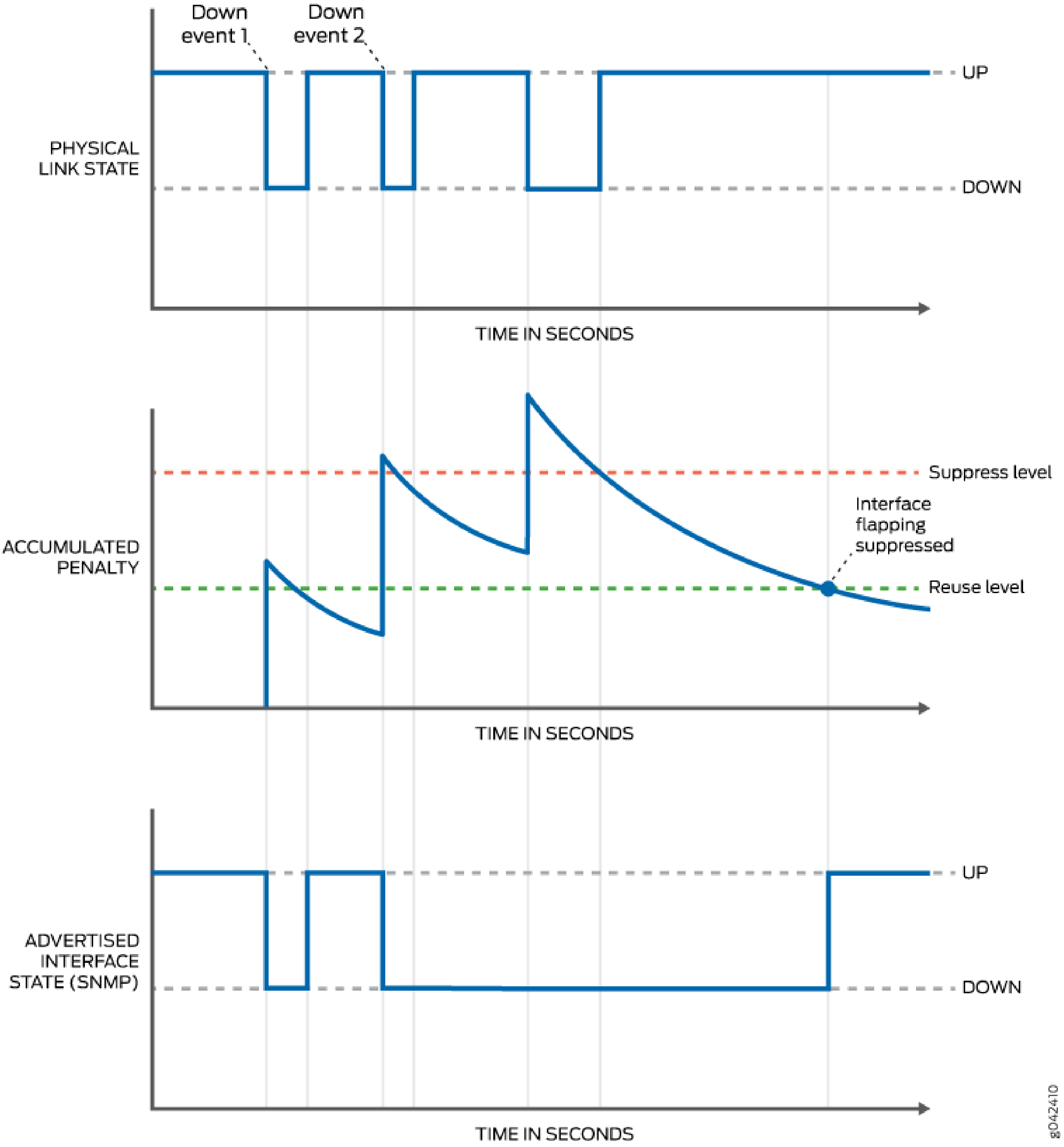


Figure 11 on page 183 shows the penalty dropping below the reuse level when the physical link is up. The system is notified of a state change immediately.

Figure 11: Physical-Level Link Is Up When the Penalty Falls Below the Reuse Level



NOTE: The QFX10002-72Q and QFX10002-36Q switches do not support `hold-time down` of less than 1 second on 100G interfaces. The recommended `hold-time down` is 3 seconds.

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. In some situations, you might want to damp interface transitions.

For example, you may want to configure damping on an interface that is connected to an add/drop multiplexer (ADM) or wavelength-division multiplexer (WDM), or to protect against SDH framer holes.

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 ms:

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

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

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

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

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

For most Ethernet interfaces, Junos OS 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.

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

Configure Damping of Aggregated Ethernet 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. In some situations, you might want to damp interface transitions.

For example, you may want to configure damping on an interface that is connected to an add/drop multiplexer (ADM) or wavelength-division multiplexer (WDM), or to protect against SDH framer holes.

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 aggregated ethernet 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 aex
   ```

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

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

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

For most Ethernet interfaces, Junos OS 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.

You can specify the hold-time value on aggregated ethernet interfaces. When you configure hold-timer for ae- interfaces, we recommend not to configure the hold-time for member links.

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.

Use Physical interface damping to confirm platform and release support for specific features.

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. It does not matter how unstable the interface has been.

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. The decay-half cycle is the interval after which the accumulated interface penalty counter is reduced by half if the interface remains stable.

   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
```

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.

Example: Configure Physical Interface Damping

IN THIS SECTION

- Requirements | 188
- Overview | 188
- Configuration | 189
- Verification | 190

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

Requirements

This example uses the following hardware and software components:

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

Overview

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

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

- Nearly instantaneous multiple flaps of short duration (ms). 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 you configure a hold-down timer and the interface goes from up to down, the system waits until the interface remains down for the hold-down timer period. Then the system advertises the interface as down. Similarly, when a hold-up timer is configured and an interface goes from down to up, it is not advertised as being up until it has remained up for the hold-up timer period.
- Periodic flaps of long duration (seconds). For longer periodic interface flaps, you configure interface damping with the `damping` statement on the interface. This damping method uses an exponential back-

off algorithm to suppress interface up-and-down event reporting to the upper-level protocols. Each time an interface goes down, a penalty is added to the interface penalty counter. If at some point the accumulated penalty exceeds the suppress level, the interface is placed in the suppress state, and further interface state up transitions are not reported to the upper-level protocols.

Configuration

IN THIS SECTION

- CLI Quick Configuration | 189
- Procedure | 189
- Results | 190

CLI Quick Configuration

To quickly configure this example, copy the following commands and paste into a TXT. Remove any line breaks and change any details to match your network configuration. Copy and paste the commands into the CLI at the `[edit]` hierarchy level.

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

Procedure

Step-by-Step Procedure

To configure damping on the PTX Series Packet Transport Router:

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

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

2. Commit the configuration:

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

Results

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

```
user@router# show interfaces
xe-6/0/0 {
    damping {
        half-life 11;
        max-suppress 2222;
        reuse 3333;
        suppress 4444;
        enable;
    }
```

Verification

IN THIS SECTION

- Verify Interface Damping on xe-6/0/0 | 190

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

Verify Interface Damping on xe-6/0/0

Purpose

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

Action

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

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

Meaning

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

Protocol Family and Interface Address Properties

IN THIS SECTION

- Configure the Protocol Family | 192
- Assign the Interface Address | 193
- Configure Default, Primary, and Preferred Addresses and Interfaces | 195
- Operational Behavior of Interfaces with the Same IPv4 Address | 198
- Configure IPCP Options for Interfaces with PPP Encapsulation | 202
- Configure Unnumbered Interfaces: Overview | 205
- Protocol MTU | 214
- Disable the Removal of Address and Control Bytes | 215

- Disable the Transmission of Redirect Messages on an Interface | 216
- Apply a Filter to an Interface | 216
- Enable Source Class and Destination Class Usage | 223
- Overview | 234
- Configure Targeted Broadcast | 236

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.

In addition to the common protocol suites, Junos OS protocol families sometimes use the following protocol suites. For more information, see *family*.

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

To configure the protocol family, complete the minimum configuration tasks under the `[edit interfaces interface-name unit logical-unit-number family family]` hierarchy.

Table 27: Protocol Family Configuration Tasks

Task	Find Details Here
Configure MTU.	Configure the Media MTU
Configure the unit and family so that the interface can transmit and receive multicast traffic only.	*Restricting Tunnels to Multicast Traffic*
Disable the sending of redirect messages by the router.	*Protocol Redirect Messages*
Assign an address to an interface.	Assign the Interface Address

SEE ALSO

family

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.

NOTE: The Point-to-Point Protocol (PPP) address is taken from the loopback interface address that has the primary attribute. When the loopback interface is configured as an unnumbered interface, it takes the primary address from the donor interface.

To assign an address to an interface, perform the following steps:

1. Configure the interface address at the `[edit interfaces` *`interface-name`* `unit` *`logical-unit-number`* `family` *`family`*`]` hierarchy level.

    - To configure an 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
```

NOTE:

- Juniper Networks routers and switches support /31 destination prefixes when used in point-to-point Ethernet configurations; however, they are not supported by many other devices, such as hosts, hubs, routers, or switches. You must determine if the peer system also supports /31 destination prefixes before configuration.

- You can configure the same IPv4 address on multiple physical interfaces. When you assign the same IPv4 address to multiple physical interfaces, the operational behavior of those interfaces differs, depending on whether they are implicitly or explicitly point-to-point.

- By default, all interfaces are assumed to be point-to-point (PPP) interfaces. For all interfaces except aggregated Ethernet, Fast Ethernet, and Gigabit Ethernet, you can explicitly configure an interface to be a point-to-point connection.

- If you configure the same IP address on multiple interfaces in the same routing instance, Junos OS applies the configuration randomly on one of the interfaces. The other interfaces will remain without an IP address.

- To configure an 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] specify the remote address of the connection for the encrypted, PPP-encapsulated, and tunnel interfaces.

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

Configure Default, Primary, and Preferred Addresses and Interfaces

IN THIS SECTION

- Default, Primary, and Preferred Addresses and Interfaces | 195
- Configure the Primary Interface for the Router | 196
- Configure the Primary Address for an Interface | 197
- Configure the Preferred Address for an Interface | 198

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.

NOTE: You can explicitly mark an interface's IP as primary and preferred using a configuration statement. If an interface is only assigned a single IP that address is considered the primary and preferred address by default. When assigned multiple IP addresses, none of which are explicitly configured as primary, the numerically lowest IP address is uses as the primary address on that interface.

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.

3. When there are multiple interfaces with "primary" and "preferred" addresses, the interface with the lowest interface index is selected, and the primary address is used. In the case that none of the interface's IP addresses are explicitly marked with the `primary` statement, the numerically lowest address on that interface is used as the system default address.

4. Any remaining interface with an IP address may be selected. This includes the router's management or internal interfaces. For this reason, it's recommended that you assign a loopback address, or explicitly configure a primary interface, to control default address selection.

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.

Primary Interface Selection Process

When no interface is explicitly configured as primary using the `primary` statement, the router automatically selects a primary interface based on the following criteria:

1. By default, the multicast-capable interface with the lowest interface index is chosen as the primary interface.

2. If no multicast-capable interface exists, the point-to-point interface with the lowest interface index is chosen.

3. Otherwise, any interface with an IP address is selected. In practice, this means that, on the router, the `fxp0` or `em0` interface is selected by default.

For example, in a router with the following interfaces:

- `fxp0.0` (management interface with IP address 192.168.1.1/24)

- `ge-0/0/0.0` (multicast-capable interface with IP address 10.1.1.1/24)

- `ge-0/0/1.0` (multicast-capable interface with IP address 10.1.2.1/24)

- `so-0/1/0.0` (point-to-point interface with IP address 172.16.1.1/30)

Without explicit configuration, `ge-0/0/0.0` would be selected as the primary interface because it's a multicast-capable interface with the lowest interface index.

To ensure predictable routing behavior, it's recommended to explicitly configure a primary interface using the `primary` statement rather than relying on the automatic selection process. Using a loopback interface (lo0) as the primary interface is a common best practice since it's not tied to any physical interface and remains available regardless of link status.

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

```
primary;
```

You can include this statement at the following hierarchy levels:

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

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

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 so-0/0/0.0 255.255.255.255` command is the primary address on interface `so-0/0/0.0`. The primary address flag 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`*`]`
- `[edit logical-systems` *`logical-system-name`* `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`*`]`
- `[edit logical-systems` *`logical-system-name`* `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.

NOTE: For all interfaces, except aggregated Ethernet, Fast Ethernet, and Gigabit Ethernet, you can explicitly configure an interface to be a point-to-point connection.

If you configure the same IP address on multiple interfaces in the same routing instance, 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
ge-0/1/0 {
     unit 0 {
        family inet {
            address 203.0.113.1/24;
        }
    }
}
```

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

The sample output shown below for the above configuration reveals that only `ge-0/1/0.0` was assigned the same IPv4 address `203.0.113.1/24` and its `link` state was `up`, while `ge-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 ge*
Interface               Admin Link Proto    Local                 Remote
        ge-0/1/0                up    up
        ge-0/1/0.0              up    up   inet     203.0.113.1/24
                                   multiservice
        ge-0/1/1                up    down
        ge-3/0/0                up    down
        ge-3/0/1                up    up
        ge-3/0/1.0              up    up   inet
                                   multiservice
```

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

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

The following sample output (for the preceding configuration) reveals that both `so-0/0/0.0` and `so-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 interface is up at any given time (following a redundancy scheme outside of the Junos OS devices scope), because both being up may cause adverse effects.

show interfaces terse

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

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

```
[edit interfaces]
user@host# show
ge-0/0/1  {
    vlan-tagging;
         unit 0 {
        vlan-id 1;
            family inet {
            address 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
gr-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 interface gets successfully configured on P2P interfaces when you try to configure the same IPv4 address for multiple instances of different interfaces.

show interfaces terse

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

Configure IPCP Options for Interfaces with PPP Encapsulation

For interfaces with PPP encapsulation, you can configure IPCP to negotiate IP address assignments and to pass network-related information such as Windows Name Service (WINS) and Domain Name System (DNS) servers, as defined in RFC 1877, *PPP Internet Protocol Control Protocol Extensions for Name Server Addresses*.

When you enable a PPP interface, you can configure an IP address, enable the interface to negotiate an IP address assignment from the remote end, or allow the interface to be unnumbered. You can also assign a destination profile to the remote end. The destination profile includes PPP properties, such as primary and secondary DNS and NetBIOS Name Servers (NBNSs). These options are described in the following sections:

> **NOTE**: The Junos OS does not request name servers from the remote end; the software does, however, send name servers to the remote end if requested.

Before you begin

You must configure the PPP encapsulation on the interface before configuring the IPCP option. The following PPP encapsulation types are supported on the logical interface:

- `atm-mlppp-llc`
- `atm-ppp-llc`
- `atm-ppp-vc-mux`
- `multilink-ppp`

For more information about PPP encapsulation, see "Configuring Interface Encapsulation on Logical Interfaces" on page 103 and Configuring ATM Interface Encapsulation

- To configure an IP address for the interface, include the `address` statement in the configuration. For more information, see *Configuring the Interface Address*.

  If you include the `address` statement in the configuration, you cannot include the `negotiate-address` or `unnumbered-address` statement in the configuration.

  When you include the `address` statement in the interface configuration, you can assign PPP properties to the remote end.

  > **NOTE**: The option to negotiate an IP address is not allowed in MLFR and MFR encapsulations.

- To enable the interface to obtain an IP address from the remote end, include the `negotiate-address` statement at the `[edit interfaces` *`interface-name`* `unit` *`logical-unit-number`* `family inet]` hierarchy level.

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

NOTE: If you include the `negotiate-address` statement in the configuration, you cannot include the `address` or `unnumbered-address` statement in the configuration.

- To configure an interface to be unnumbered, include the `unnumbered-address` and `destination` statements in the configuration.

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

NOTE:

- The `unnumbered-address` statement enables the local address to be derived from the specified interface. The interface name must include a logical unit number and must have a configured address (see *Configuring the Interface Address*). Specify the IP address of the remote interface with the `destination` statement.

- If you include the `unnumbered-address` statement in the configuration, you cannot include the `address` or `negotiate-address` statement in the interface configuration.

- To assign PPP properties to the remote end include the `destination-profile` statement:

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

```
[edit interfaces interface-name unit  logical-unit-number family inet unnumbered-address
interface-name]
user@host# set destination-profile name
```

NOTE:

- You can assign PPP properties to the remote end, after you include the `address` or `unnumbered-address` statement in the interface configuration.

- You define the profile at the `[edit access group-profile name ppp]` hierarchy level. For more information, see *Configuring the Group Profile for L2TP and PPP*.

SEE ALSO

> *Configuring the Group Profile for L2TP and PPP*

Configure Unnumbered Interfaces: Overview

IN THIS SECTION

- Configure an Unnumbered Point-to-Point Interface | **205**
- Configure an Unnumbered Ethernet or Demux Interface | **206**
- Configure a Secondary Address as a Preferred Source Address for Unnumbered Ethernet or Demux Interfaces | **208**
- Restrictions for Unnumbered Ethernet Interface Configurations | **208**
- Example: Display the Unnumbered Ethernet Interface Configuration | **210**
- Example: Display the Configured Preferred Source Address for an Unnumbered Ethernet Interface | **211**
- Example: Display the Configuration for the Unnumbered Ethernet Interface as the Next Hop for a Static Route | **213**

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:

- For interfaces with Point-to-Point Protocol (PPP) encapsulation, you can configure an unnumbered interface by including the unnumbered-interface statement in the configuration. For more information, see "Configuring IPCP Options for Interfaces with PPP Encapsulation" on page 202.
- 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" on page 265.

  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 referred to as the *borrower interface*. The interface from which the IP address is borrowed is referred to as the *donor interface*. In the `unnumbered-address` statement, *`interface-name`* specifies the donor interface. For an unnumbered Ethernet interface, the donor interface can be an Ethernet, ATM, SONET, or loopback interface that has a logical unit number and configured IP address and is not itself an unnumbered interface. For an unnumbered IP 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.

- For information about host routes, see the MPLS Applications User Guide.

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.

  For information about these statement options, see *Configuring the Interface Address*.
- 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;
     }
  }
}
```

For more information about configuring the same address on multiple interfaces, see *Configuring the Interface Address*.

Example: Display the Unnumbered Ethernet Interface Configuration

IN THIS SECTION

- Purpose | 210
- Action | 211
- Meaning | 211

Purpose

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

- Unnumbered interface —ge-1/0/0
- Donor interface —ge-0/0/0
- Donor interface address —4.4.4.1/24

The unnumbered interface “borrows” an IP address from the donor interface.

Action

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

```
interfaces {
    ge-0/0/0 {
        unit 0 {
            family inet {
                address 4.4.4.1/24;
            }
        }
    }
    ge-1/0/0 {
        unit 0 {
            family inet {
                unnumbered-address ge-0/0/0.0;
            }
        }
    }
}
```

Meaning

The sample configuration works correctly on M and T Series routers. For unnumbered interfaces on MX Series routers, you must also configure static routes on an unnumbered Ethernet interface by including the `qualified-next-hop` statement at the `[edit routing-options static route destination-prefix]` hierarchy level to specify the unnumbered Ethernet interface as the next-hop interface for a configured static route.

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

IN THIS SECTION

- Purpose | 212
- Action | 212
- Meaning | 212

Purpose

To display the configuration of preferred source address for an unnumbered interface at the `[edit interfaces` *`interface-name`* `unit` *`logical-unit-number`* `family inet]` hierarchy level:

- Unnumbered interface —ge-4/0/0
- Donor interface —lo0
- Donor interface primary address—2.2.2.1/32
- Donor interface secondary address—3.3.3.1/32

Action

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

```
interfaces {
    lo0 {
        unit 0 {
            family inet {
                address 2.2.2.1/32;
                address 3.3.3.1/32;
            }
        }
    }
}
interfaces {
    ge-4/0/0 {
        unit 0 {
            family inet {
                unnumbered-address lo0.0 preferred-source-address 3.3.3.1;
            }
        }
    }
}
```

Meaning

The loopback interface `lo0` is the donor interface from which an unnumbered Ethernet interface `ge-4/0/0` “borrows” an IP address.

The example shows one of the loopback interface’s secondary addresses, 3.3.3.1, as the preferred source address for the unnumbered Ethernet interface.

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

IN THIS SECTION

- Purpose | 213
- Action | 213
- Meaning | 214

Purpose

To display the unnumbered interface configured as the next hop for the static route at the `[edit interfaces` *`interface-name`* `unit` *`logical-unit-number`* `family inet]` hierarchy level:

- Unnumbered interface —ge-0/0/0
- Donor interface —lo0
- Donor interface primary address—5.5.5.1/32
- Donor interface secondary address—6.6.6.1/32
- Static route—7.7.7.1/32

Action

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

```
interfaces {
    ge-0/0/0 {
        unit 0 {
            family inet {
                unnumbered-address lo0.0;
            }
        }
    }
}
```

```
lo0  {
        unit 0 {
            family inet {
                address 5.5.5.1/32;
                address 6.6.6.1/32;
            }
        }
    }
```

- The following configuration enables the kernel to install a static route to address 7.7.7.1/32 with a next hop through unnumbered interface ge-0/0/0.0.

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

Meaning

In this example, `ge-0/0/0` is the unnumbered interface. A loopback interface, `lo0`, is the donor interface from which `ge-0/0/0` “borrows” an IP address. The example also configures a static route to `7.7.7.1/32` with a next hop through unnumbered interface `ge-0/0/0.0`.

Protocol MTU

IN THIS SECTION

- Overview | 215
- Protocol MTU for MPLS | 215

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.

You can configure the protocol MTU on all tunnel interfaces except virtual tunnel (VT) interfaces. Junos OS sets the MTU size for VT interfaces to unlimited by default.

Protocol MTU for MPLS

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

Disable the Removal of Address and Control Bytes

For Point-to-Point Protocol (PPP) CCC-encapsulated interfaces, the address and control bytes are removed by default before the packet is encapsulated into a tunnel.

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]`
- `[edit logical-systems` *`logical-system-name`* `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 levels:

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

To disable the sending of protocol redirect messages for the entire router or switch, include the `no-redirects` statement at the `[edit system]` hierarchy level.

SEE ALSO

> *no-redirects*

Apply a Filter to an Interface

IN THIS SECTION

- Define Interface Groups in Firewall Filters | **217**
- Apply a Filter to an Interface | **218**

Define Interface Groups in Firewall Filters

IN THIS SECTION

- Filter-Based Forwarding on the Output Interface | 217

When applying a firewall filter, you can define an interface to be part of an *interface group*. Packets received on that interface are tagged as being part of the group. You can then match these packets using the `interface-group` match statement, as described in the Routing Policies, Firewall Filters, and Traffic Policers User Guide.

To define the interface to be part of an interface group, include the `group` statement:

```
group filter-group-number;
```

You can include this statement at the following hierarchy levels:

- `[edit interfaces interface-name unit logical-unit-number family family filter]`
- `[edit logical-systems logical-system-name interfaces interface-name unit logical-unit-number family family filter]`

NOTE: The number 0 is not a valid interface group number.

Filter-Based Forwarding on the Output Interface

If port-mirrored packets are to be distributed to multiple monitoring or collection interfaces, based on 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.

For more information about FBF configuration, see the Junos OS Routing Protocols Library for Routing Devices. For more information about port mirroring, see the Junos OS Services Interfaces Library for Routing Devices.

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*]`

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

In the `family` statement, the protocol family can be `ccc`, `inet`, `inet6`, `mpls`, or `vpls`.

In the `group` statement, specify the interface group number to associate with the filter.

In the `input` statement, list the name of one firewall filter to be evaluated when packets are received on the interface.

In the `input-list` statement, list the names of filters to evaluate when packets are received on the interface. You can include up to 16 filter names.

In the `output` statement, list the name of one firewall filter to be evaluated when packets are transmitted on the interface.

NOTE: Output filters do not work for broadcast and multicast traffic, including VPLS traffic (except in MX Series routers with MPC/MIC interfaces), as shown in "Apply a Filter to an Interface" on page 218.

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

NOTE: On an MX Series router, you cannot apply as an output filter a firewall filter configured at the `[edit firewall filter family ccc]` hierarchy level. You can apply firewall filters configured for the `family ccc` statement as input filters only.

In the `output-list` statement, list the names of filters to evaluate when packets are transmitted on the interface. You can include up to 16 filter names.

You can use the same filter one or more times. On M Series routers (except the M320 and M120 routers), if you apply a firewall filter or policer to multiple interfaces, the filter or policer acts on the sum of traffic entering or exiting those interfaces.

On T Series, M120, and M320 routers, interfaces are distributed among multiple packet forwarding components. Therefore, on these routers, if you apply a firewall filter or policer to multiple interfaces, the filter or policer acts on the traffic stream entering or exiting each interface, regardless of the sum of traffic on the multiple interfaces.

For more information on Understanding Ethernet Frame Statistics, see the *MX Series Layer 2 Configuration Guide*.

If you apply the filter to interface lo0, it is applied to packets received or transmitted by the Routing Engine. You cannot apply MPLS filters to the management interface (fxp0 or em0) or the loopback interface (lo0).

Filters applied at the [set interfaces lo0 unit 0 family any filter input] hierarchy level are not installed on T4000 Type 5 FPCs.

For more information about firewall filters, see the Routing Policies, Firewall Filters, and Traffic Policers User Guide. For more information about MPLS filters, see the MPLS Applications User Guide.

Example: Input Filter for VPLS Traffic

For M Series and T Series routers only, apply an input filter to VPLS traffic. Output filters do not work for broadcast and multicast traffic, including VPLS traffic.

Note that on MX Series routers with MPC/MIC interfaces, the VPLS filters on the egress route is applicable to broadcast, multicast, and unknown unicast traffic.

```
[edit interfaces]
fe-2/2/3 {
    vlan-tagging;
    encapsulation vlan-vpls;
    unit 601 {
        encapsulation vlan-vpls;
        vlan-id 601;
        family vpls {
            filter {
                input filter1;  # Works for multicast destination MAC address
                output filter1;  # Does not work for multicast destination MAC address
            }
        }
    }
}
[edit firewall]
family vpls {
    filter filter1 {
        term 1 {
            from {
                destination-mac-address {
                    01:00:0c:cc:cc:cd/48;
                }
            }
            then {
                discard;
            }
        }
        term 2 {
            then {
                accept;
            }
        }
    }
}
```

Example: Filter-Based Forwarding at the Output Interface

The following example illustrates the configuration of filter-based forwarding at the output interface. In this example, the packet flow follows this path:

1. A packet arrives at interface `fe-1/2/0.0` with source and destination addresses `10.50.200.1` and `10.50.100.1`, respectively.

2. The route lookup in routing table `inet.0` points to egress interface `so-0/0/3.0`.

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

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

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

```
}
so-2/0/0 {
    unit 0 {
        family inet {
            address 10.50.20.2/25;
        }
    }
}
[edit firewall]
filter fbf {
    term 0 {
        from {
            source-address {
                10.50.200.0/25;
            }
        }
        then routing-instance fbf;
    }
    term d {
        then count d;
    }
}
[edit routing-instances]
fbf {
    instance-type forwarding;
    routing-options {
        static {
            route 10.50.100.0/25 next-hop so-2/0/0.0;
        }
    }
}
[edit routing-options]
interface-routes {
    rib-group inet fbf-group;
}
static {
    route 10.50.100.0/25 next-hop 10.50.10.1;
}
rib-groups {
    fbf-group {
        import-rib [inet.0 fbf.inet.0];
    }
}
```

Enable Source Class and Destination Class Usage

IN THIS SECTION

- Source Class and Destination Class Usage Overview | **223**
- Enable Source Class and Destination Class Usage | **227**

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.

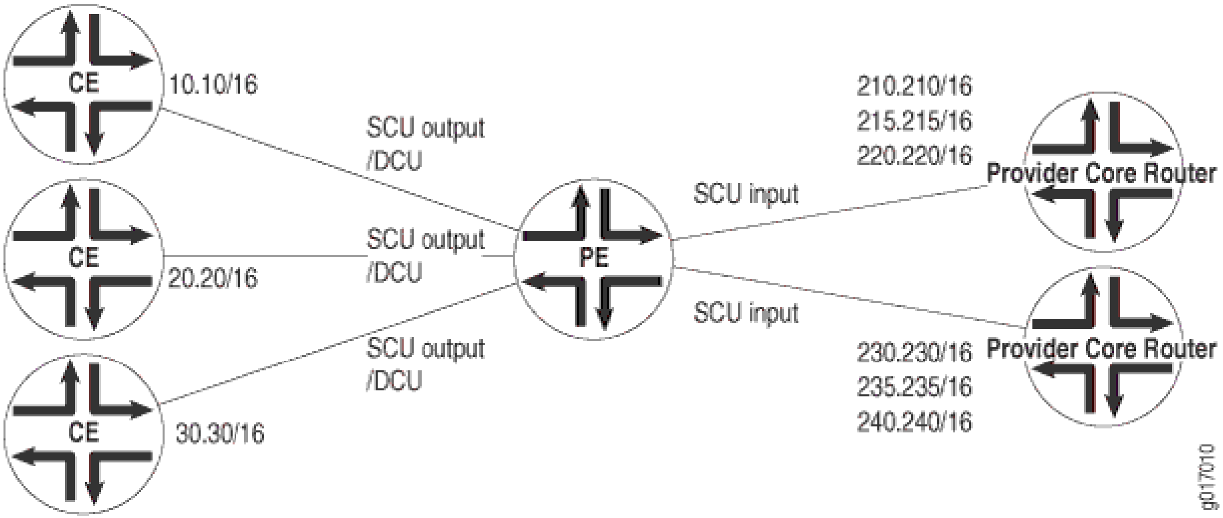
Destination class usage (DCU) counts packets from customers by performing lookup of the IP destination address. DCU makes it possible to track traffic originating from the customer edge and destined for specific prefixes on the provider core router.

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 12: 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)]
- [edit logical-systems *logical-system-name* interfaces *interface-name* unit *logical-unit-number* family (inet | inet6 | mpls)]

For SCU to work, you must configure at least one input interface and at least one output interface.

The ability to count a single packet for both SCU and DCU accounting depends on the underlying physical interface.

- For traffic over 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`.

On T Series, M120, and M320 routers, the source class and destination classes are not carried across the router fabric. The implications of this are as follows:

- On T Series, M120, and M320 routers, SCU and DCU accounting is performed before the packet enters the fabric.

- On M7i, M10i, M120, and M320 routers, on MX Series routers with non-MPC, and on T Series routers, SCU and DCU accounting is performed before output filters are evaluated. Consequently, if a packet matches a firewall filter match condition, the packet is included in SCU or DCU accounting; the packet is counted for any term action (including the `discard` action).

- On M120, M320, and T Series routers, the `destination-class` and `source-class` statements are supported at the `[edit firewall family` *`family-name`* `filter` *`filter-name`* `term` *`term-name`* `from]` hierarchy level only for the filter applied to the forwarding table. On M7i, M10i, and MX Series routers, these statements are supported.

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.

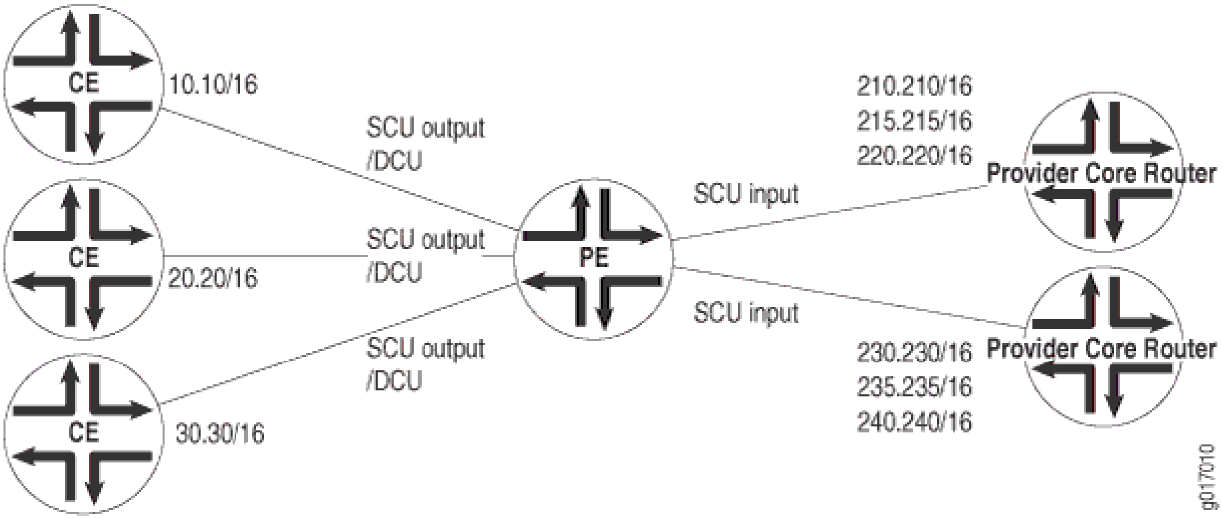
In Junos OS Release 9.3 and later, you can configure SCU accounting for Layer 3 VPNs configured with the `vrf-table-label` statement. Include the `source-class-usage` statement at the `[edit routing-instances` *routing-instance-name* vrf-table-label] hierarchy level. The source-class-usage statement at this hierarchy level is supported only for the virtual routing and forwarding (VRF) instance type.

NOTE: 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. For more information, see the Junos OS VPNs Library for Routing Devices.

For a complete discussion about source and destination class accounting profiles, see the Junos OS Network Management Administration Guide for Routing Devices. For more information about MPLS, see the MPLS Applications User Guide.

Enable Source Class and Destination Class Usage

Figure 13: 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 {
    so-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 `so-0/0/2` in the OSPF process.

2.

```
Router A
[edit]
interfaces {
    so-0/0/2 {
        unit 0 {
            family inet {
                address 10.255.50.2/24;
            }
        }
    }
    lo0 {
        unit 0 {
            family inet {
                address 10.255.192.10/32;
            }
        }
    }
}
protocols {
    ospf {
        area 0.0.0.0 {
            interface so-0/0/2.0;
            interface lo0.0;
        }
    }
}
```

3. 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 {
    so-0/0/1 {
        unit 0 {
            family inet {
                accounting {
                    source-class-usage {
                        input;
                        output;
                    }
                }
                address 10.255.50.1/24;
            }
        }
    }
    so-0/0/3 {
        unit 0 {
            family inet {
                accounting {
                    source-class-usage {
                        input;
                        output;
                    }
                }
                address 10.255.10.3/24;
            }
        }
    }
    lo0 {
        unit 0 {
            family inet {
                address 10.255.6.111/32;
            }
        }
    }
}
protocols {
    ospf {
        area 0.0.0.0 {
            interface so-0/0/1.0;
            interface so-0/0/3.0;
        }
    }
}
routing-options {
    forwarding-table {
        export scu-policy;
    }
}
policy-options {
    policy-statement scu-policy {
        term 0 {
            from {
                route-filter 10.255.192.0/24 orlonger;
            }
            then source-class scu-class-a;
        }
        term 1 {
            from {
                route-filter 10.255.165.0/24 orlonger;
            }
            then source-class scu-class-b;
        }
    }
}
```

4. 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 `so-0/0/4` in the OSPF process.

   ```
   Router B
   interfaces {
       so-0/0/4 {
           unit 0 {
               family inet {
                   address 10.255.10.4/24;
               }
           }
       }
       lo0 {
           unit 0 {
               family inet {
                   address 10.255.165.226/32;
               }
           }
       }
   }
   protocols {
       ospf {
           area 0.0.0.0 {
               interface so-0/0/4.0;
               interface lo0.0;
           }
       }
   }
   ```

5. 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.

   In Junos OS Release 9.3 and later, you can configure SCU accounting for Layer 3 VPNs configured with the `vrf-table-label` statement. Include the `source-class-usage` statement at the `[edit routing-instances` *`routing-instance-name`* `vrf-table-label]` hierarchy level. The `source-class-usage` statement at this hierarchy level is supported only for the virtual routing and forwarding (VRF) instance type. DCU is not supported when the `vrf-table-label` statement is configured. For more information, see the Junos OS VPNs Library for Routing Devices.

```
[edit routing-instances]
VPN-A {
    instance-type vrf;
    interface at-2/1/1.0;
    interface vt-0/3/0.0;
    route-distinguisher 10.255.14.225:100;
    vrf-import import-policy-A;
    vrf-export export-policy-A;
    protocols {
        bgp {
            group to-r4 {
                local-address 10.27.253.1;
                peer-as 400;
                neighbor 10.27.253.2;
            }
        }
    }
}
```

7. Send traffic received from the VPN out the source class output interface.

```
[edit interfaces]
at-2/1/0 {
    unit 0 {
        family inet {
            accounting {
                source-class-usage {
                    output;
                }
            }
        }
    }
}
```

For more information about VPNs, see the Junos OS VPNs Library for Routing Devices. For more information about virtual loopback tunnel interfaces, see the Junos OS Services Interfaces Library for Routing Devices.

SEE ALSO

accounting

destination-classes

family

forward-and-send-to-re

source-classes

targeted-broadcast

unit

Overview

IN THIS SECTION

- Targeted Broadcast Overview | 235
- Targeted Broadcast Implementation | 235
- When to Enable Targeted Broadcast | 236
- When Not to Enable Targeted Broadcast | 236

Targeted broadcast is a process of flooding a target subnet with L3 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.

IP directed broadcast is a technique where a broadcast packet is sent to a specific remote subnet, and then broadcast within that subnet. You can use IP directed broadcast to facilitate remote network management by sending broadcast packets to hosts on a specified subnet without broadcasting to the entire network. IP directed broadcast packets are broadcast on only the target subnet. The rest of the network treats IP directed broadcast packets as unicast packets and forwards them accordingly.

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 VRF instances.

Regular L3 broadcast IP packets originating from a subnet are broadcast within the same subnet. When these IP packets reach a different subnet, the packets are forwarded to the Routing Engine (to be forwarded to other applications). Hence, 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.

L3 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 the packets 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. The forwarding is helpful when the router is flooded with packets to process, or to both an egress interface and the Routing Engine.

Any *firewall filter* that is configured on the Routing Engine lo0 cannot be applied to IP packets that are forwarded to the Routing Engine as a result of a targeted broadcast. The reason is broadcast packets are forwarded as flood next-hop traffic and not as local next-hop traffic. You can apply a firewall filter only to local next-hop routes for traffic directed toward the Routing Engine.

Targeted Broadcast Overview

Targeted broadcast packets have a destination IP address that is a valid broadcast address for the subnet that is the target of the directed broadcast (the target subnet). The intent of a targeted broadcast is to flood the target subnet with the broadcast packets without broadcasting to the entire network. Targeted broadcast packets cannot originate from the target subnet.

When you send a targeted broadcast packet, as it travels to the target subnet, the network forwards it in the same way as it forwards a unicast packet. When the packet reaches a switch that is directly connected to the target subnet, the switch checks to see whether targeted broadcast is enabled on the interface that is directly connected to the target subnet:

- If targeted broadcast is enabled on that interface, the switch broadcasts the packet on that subnet by rewriting the destination IP address as the configured broadcast IP address for the subnet. The switch converts the packet to a link-layer broadcast packet that every host on the network processes.

- If targeted broadcast is disabled on the interface that is directly connected to the target subnet, the switch drops the packet.

Targeted Broadcast Implementation

You configure targeted broadcast on a per-subnet basis by enabling targeted broadcast on the L3 interface of the subnet’s VLAN. When the switch that is connected to that subnet receives a packet that has the subnet's broadcast IP address as the destination address, the switch broadcasts the packet to all hosts on the subnet.

By default, targeted broadcast is disabled.

When to Enable Targeted Broadcast

Targeted broadcast is disabled by default. Enable targeted broadcast when you want to perform remote management or administration services such as backups or WOL tasks on hosts in a subnet that does not have a direct connection to the Internet.

Enabling targeted broadcast on a subnet affects only the hosts within that subnet. Only packets received on the subnet's L3 interface that have the subnet's broadcast IP address as the destination address is flooded on the subnet.

When Not to Enable Targeted Broadcast

Typically, you do not enable targeted broadcast on subnets that have direct connections to the Internet. Disabling targeted broadcast on a subnet's L3 interface affects only that subnet. If you disable targeted broadcast on a subnet and a packet that has the broadcast IP address of that subnet arrives at the switch, the switch drops the broadcast packet.

If a subnet has a direct connection to the Internet, enabling targeted broadcast on it increases the network's susceptibility to DoS attacks.

A malicious attacker can spoof a source IP address to deceive a network into identifying the attacker as legitimate. The attacker can then send targeted broadcasts with ICMP echo (ping) packets. When the hosts on the network with targeted broadcast enabled receive the ICMP echo packets, the hosts send replies to the victim that has the spoofed source IP address. The replies create a flood of ping replies in a DoS attack that can overwhelm the spoofed source address known as a *smurf* attack. Another common DoS attack on exposed networks with targeted broadcast enabled is a *fraggle* attack. The attack is similar to a smurf attack except that the malicious packet is a UDP echo packet instead of an ICMP echo packet.

Configure Targeted Broadcast

IN THIS SECTION

- Configure Targeted Broadcast | **237**

- Display Targeted Broadcast Configuration Options | 238

Configure Targeted Broadcast

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

Either of these configurations is acceptable:

- You can allow the IP broadcast packets destined for a Layer 3 address to be forwarded through the egress interface and to send a copy of the IP broadcast packets to the Routing Engine.
- You can allow the IP broadcast packets to be forwarded through 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 interface.

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

   or

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

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. Forward IP broadcast packets to a Layer 3 address:

   a. through the egress interface and send a copy of the same packets to the Routing Engine.

      ```
      [edit interfaces interface-name unit interface-unit-number family inet targeted-broadcast]
      user@host# forward-and-send-to-re;
      ```

      or

   b. through the egress interface only.

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

Display Targeted Broadcast Configuration Options

IN THIS SECTION

- Forward IP Broadcast Packets on the Egress Interface and to the Routing Engine | **238**
- Forward IP Broadcast Packets on the Egress Interface Only | **239**

The following example topics display targeted broadcast configuration options:

Forward IP Broadcast Packets on the Egress Interface and to the Routing Engine

IN THIS SECTION

- Purpose | **239**
- Action | **239**

Purpose

Display the configuration when targeted broadcast is configured on the egress interface to forward the IP broadcast packets on the egress interface and to send a copy of the same packets to the Routing Engine.

Action

To display the configuration, run the `show` command at the `[edit interfaces` *`interface-name`* `unit` *`interface-unit-number`* `family inet]` where the interface name is ge-2/0/0, the unit value is set to 0, 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;
}
```

To display the configuration for irb, run the `show` command at the `[edit interfaces irb unit` *`interface-unit-number`* `family inet]`.

```
[edit interfaces irb unit interface-unit-number family inet]
user@host#show
targeted-broadcast {
    forward-and-send-to-re;
}
```

Forward IP Broadcast Packets on the Egress Interface Only

IN THIS SECTION

- Purpose | **240**
- Action | **240**

Purpose

Display the configuration when targeted broadcast is configured on the egress interface to forward the IP broadcast packets on the egress interface only.

Action

To display the configuration, run the `show` command at the `[edit interfaces` *interface-name* `unit` *interface-unit-number* `family inet]` where the interface name is ge-2/0/0, the unit value is set to 0, 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;
}
```

To display the configuration, run the `show` command at the `[edit interfaces irb unit` *interface-unit-number* `family inet]`.

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

2
CHAPTER

Other Interfaces

IN THIS CHAPTER

- Discard Interfaces | 242
- IP Demultiplexing Interfaces | 245
- Loopback Interfaces | 263
- Serial Interfaces | 269

Discard Interfaces

IN THIS SECTION

- Discard Interface Overview | 242
- Discard Interface Configuration | 243

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 forwards to the discard interface and what it does with that traffic. A local policy determines which traffic Junos OS forwards to the discard interface. Junos OS 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 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 | 243
- Configure an Output Policy | 244

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
```

Change History Table

Feature support is determined by the platform and release you are using. Use Feature Explorer to determine if a feature is supported on your platform.

Release	Description
20.1	Starting in Junos OS release 20.1, for MX Series routers, the discard interface is also supported for the `inet6` family.

IP Demultiplexing Interfaces

IN THIS SECTION

- Demultiplexing Interface Overview | 246
- Configuring an IP Demultiplexing Interface | 249
- Configuring a VLAN Demultiplexing Interface | 255

Demultiplexing (demux) interfaces are logical interfaces that share a common, underlying interface. You can create logical subscriber interfaces using static or dynamic demultiplexing interfaces. In addition, you can use IP demultiplexing interfaces or VLAN demultiplexing interfaces when creating logical subscriber interfaces.

Demultiplexing Interface Overview

IN THIS SECTION

- IP Demux Interface Overview | **246**
- VLAN Demux Interface Overview | **247**
- Guidelines to Remember When Configuring A Demux Interface | **247**
- MAC Address Validation on Static Demux Interfaces | **248**

Demux interfaces support only Gigabit Ethernet, Fast Ethernet, 10-Gigabit Ethernet, or aggregated Ethernet underlying interfaces.

Use Feature Explorer to confirm platform and release support for demultiplexing (demux) interfaces.

NOTE: You can also configure demux interfaces dynamically. For information about how to configure dynamic VLAN demux interfaces, see *Configuring Dynamic Subscriber Interfaces Using VLAN Demux Interfaces in Dynamic Profiles*.

To configure static demux interfaces, see "Configuring a VLAN Demultiplexing Interface" on page 255 and "Configuring an IP Demultiplexing Interface" on page 249.

IP Demux Interface Overview

IP demux interfaces use the IP source address or IP destination address to demultiplex received packets when the subscriber is not uniquely identified by a Layer 2 circuit.

To determine which IP demux interface to use, the destination or source prefix is matched against the destination or source address of packets that the underlying interface receives. The underlying interface family type must match the demux interface prefix type.

VLAN Demux Interface Overview

VLAN demux interfaces use the VLAN ID to demultiplex received packets when the subscriber is not uniquely identified. A VLAN demux interface uses an underlying logical interface to receive packets.

To determine which VLAN demux interface to use, the VLAN ID is matched against that which the underlying interface receives.

NOTE: VLAN demux subscriber interfaces over aggregated Ethernet physical interfaces are supported only for MX Series routers that have only Trio MPCs installed. If the router has other MPCs in addition to Trio MPCs, theCLI accepts the configuration but errors are reported when the subscriber interfaces are brought up.

Guidelines to Remember When Configuring A Demux Interface

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

- Demux interfaces are supported on M120 or MX Series routers only.
- Only demux0 is supported. If you configure another demux interface, such as demux1, the configuration commit fails.
- You can configure only one `demux0` interface per chassis, but you can define logical demux interfaces on top of it (for example, `demux0.1`, `demux0.2`, and so on).
- If the address in a received packet does not match any demux prefix, the packet is logically received on the underlying interface. For this reason, the underlying interface is often referred to as the *primary* interface.

Points to Remember When Configuring an IP Demux Interface

In addition to the guidelines in "Guidelines to Remember When Configuring A Demux Interface" on page 247, the following guidelines are to be noted when configuring an IP demux interface:

- You must associate demux interfaces with an underlying logical interface.

  NOTE: IP demux interfaces currently support only Gigabit Ethernet, Fast Ethernet, 10-Gigabit Ethernet, and aggregated Ethernet underlying interfaces.

- The demux underlying interface must reside on the same logical system as the demux interfaces that you configure over it.

- IP demux interfaces currently supports the Internet Protocol version 4 (IPv4) suite inet and Internet Protocol version 6 (IPv6) suite inet6 family types.
- You can configure more than one demux prefix for a given demux unit. However, you cannot configure the exact same demux prefix on two different demux units with the same underlying interface.
- You can configure overlapping demux prefixes on two different demux units with the same underlying prefix. However, under this configuration, best match rules apply (in other words, the most specific prefix wins).

Points to Remember When Configuring a VLAN Demux Interface

In addition to the guidelines in "Guidelines to Remember When Configuring A Demux Interface" on page 247, the following guidelines are to be noted when configuring a VLAN demux interface:

- You must associate VLAN demux interfaces with an underlying logical interface.

  NOTE: VLAN demux interfaces currently support only Gigabit Ethernet, Fast Ethernet, 10-Gigabit Ethernet, and aggregated Ethernet underlying interfaces.

- The demux underlying interface must reside on the same logical system as the demux interfaces that you configure over it.
- VLAN demux interfaces currently supports the Internet Protocol version 4 (IPv4) suite inet and Internet Protocol version 6 (IPv6) suite inet6 family types.

MAC Address Validation on Static Demux Interfaces

MAC address validation enables the router to validate that received packets contain a trusted IP source and an Ethernet MAC source address.

MAC address validation is supported on static demux interfaces on MX Series routers only.

There are two types of MAC address validation that you can configure:

Loose

Forwards packets when both the IP source address and the MAC source address match one of the trusted address tuples.

Drops packets when the IP source address matches one of the trusted tuples, but the MAC address does not support the MAC address of the tuple

Continues to forward packets when the source address of the incoming packet does not match any of the trusted IP addresses.

Strict

Forwards packets when both the IP source address and the MAC source address match one of the trusted address tuples.

Drops packets when the MAC address does not match the tuple's MAC source address, or when IP source address of the incoming packet does not match any of the trusted IP addresses.

SEE ALSO

Associating VLAN IDs to VLAN Demux Interfaces

Binding VLAN IDs to Logical Interfaces

Subscriber Interfaces and Demultiplexing Overview

Configuring an IP Demultiplexing Interface

IN THIS SECTION

- Configuring an IP Demux Underlying Interface | 249
- Configuring the IP Demux Interface | 252
- Configuring MAC Address Validation on Static IP Demux Interfaces | 254

Demultiplexing (demux) interfaces are logical interfaces that share a common, underlying interface. You can configure IP demultiplexing interfaces or VLAN demultiplexing interfaces.

To configure an IP demux interface, you must configure the demux prefixes that are used by the underlying interface and then configure the IP demultiplexing interface as explained in the following tasks:

Configuring an IP Demux Underlying Interface

An IP demux interface uses an underlying logical interface to receive packets. To determine which IP demux interface to use, the destination or source prefix is matched against the destination or source address of packets that the underlying interface receives. The underlying interface family type must match the demux interface prefix type.

NOTE: IP demux interfaces currently support only Gigabit Ethernet, Fast Ethernet, 10-Gigabit Ethernet, and aggregated Ethernet underlying interfaces.

To configure a logical interface as an IP demux underlying interface with demux source:

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

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

2. Configure the interface as fe-*x*/ *y*/z and the logical interface with the `unit` statement. Note that IP demux interfaces currently support only Gigabit Ethernet, Fast Ethernet, 10-Gigabit Ethernet, and aggregated Ethernet underlying interfaces. In this procedure, we show a Fast Ethernet interface as an example.

   ```
   [edit interfaces]
   user@host# edit fe-x/y/z unit logical-unit-number
   ```

3. Configure the logical demux source family type on the IP demux underlying interface as inet or inet6, or both.

   ```
   [edit interfaces fe-x/y/z unit logical-unit-number]
   user@host# set demux-source (inet | inet6)
   ```

   or

   ```
   [edit interfaces fe-x/y/z unit logical-unit-number]
   user@host# set demux-source [inet inet6]
   ```

4. (Optional) To improve datapath performance for DHCPv4 subscribers, specify that only subscribers with 32-bit prefixes are allowed to come up on the interface.

   ```
   [edit interfaces fe-x/y/z unit logial-unit-number]
   user@host# set host-prefix-only
   ```

> **NOTE**: This step requires that you specify the `demux-source` as only `inet`. A commit error occurs if you specify only `inet6` or both `inet` and `inet6`.

5. Save the configuration and move to top of the hierarchy level.

```
[edit interfaces fe-x/y/z unit logial-unit-number]
user@host# commit
user@host# top
```

To configure a logical interface as an IP demux underlying interface with demux destination:

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

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

2. Configure the interface as fe-*x*/ *y*/z and the logical interface with the `unit` statement. Note that IP demux interfaces currently support only Gigabit Ethernet, Fast Ethernet, 10-Gigabit Ethernet, and aggregated Ethernet underlying interfaces.

```
[edit interfaces]
user@host# edit fe-x/y/z unit logical-unit-number unit logical-unit-number
```

3. Configure the logical demux destination family type on the IP demux underlying interface as inet or inet6.

```
[edit interfaces fe-x/y/z unit logical-unit-number]
user@host# set demux-destination (inet | inet6)
```

4. Save the configuration and move to top of the hierarchy level.

```
[edit interfaces fe-x/y/z unit logial-unit-number]
user@host# commit
user@host# top
```

Configuring the IP Demux Interface

You can configure one or more logical demux source prefixes or destination prefixes after specifying an underlying interface for the static demux interface to use. This underlying interface must reside on the same logical system as the demux interface.

You configure demux prefixes for use by the underlying interface. The demux prefixes can represent individual hosts or networks. For a given demux interface unit, you can configure either demux source or demux destination prefixes but not both.

You can choose not to configure a demux source or demux destination prefix. This type of configuration results in a transmit-only interface.

To configure the IP demux interface with source prefix:

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

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

2. Configure the interface as a logical demux interface (for example, demux0 interface) and configure the logical interface with the `unit` statement.

   NOTE: You can configure only one demux0 interface per chassis, but you can define logical demux interfaces on top of it (for example, demux0.1, demux0.2, and so on).

   ```
   [edit interfaces]
   user@host# edit demux0 unit logical-unit-number
   ```

3. Configure the underlying interface on which the demux interface is running under the `demux-options` statement.

   ```
   [edit interfaces demux0 unit logical-unit-number]
   user@host# set demux-options underlying-interface interface-name
   ```

4. Configure the protocol family.

   ```
   [edit interfaces demux0 unit logical-unit-number]
   user@host# edit family family
   ```

5. Configure one or more logical demux source prefixes (IP address). The prefixes are matched against the source address of packets that the underlying interface receives. When a match occurs, the packet is processed as if it was received on the demux interface.

```
[edit interfaces demux0 unit logical-unit-number family family]
user@host# set demux-source source-prefix
```

6. Save the configuration and move to top of the hierarchy level.

```
[edit interfaces demux0 unit logical-unit-number family family]
user@host# commit
user@host# top
```

To configure the IP demux interface with destination prefix:

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

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

2. Configure the interface as a logical demux interface (for example, demux0 interface) and configure the logical interface with the `unit` statement.

> **NOTE**: You can configure only one demux0 interface per chassis, but you can define logical demux interfaces on top of it (for example, demux0.1, demux0.2, and so on).

```
[edit interfaces]
user@host# edit demux0 unit logical-unit-number
```

3. Configure the underlying interface on which the demux interface is running under the `demux-options` statement.

```
[edit interfaces demux0 unit logical-unit-number]
user@host# set demux-options underlying-interface interface-name
```

4. Configure the protocol family.

```
[edit interfaces demux0 unit logical-unit-number]
user@host# edit family family
```

5. Configure one or more logical demux destination prefixes. The prefixes are matched against the destination address of packets that the underlying interface receives. When a match occurs, the packet is processed as if it was received on the demux interface.

```
[edit interfaces demux0 unit logical-unit-number family family]
user@host# set demux-destination destination-prefix
```

6. Save the configuration and move to top of the hierarchy level.

```
[edit interfaces demux0 unit logical-unit-number family family]
user@host# commit
user@host# top
```

Configuring MAC Address Validation on Static IP Demux Interfaces

MAC address validation enables the router to validate that received packets contain a trusted IP source and an Ethernet MAC source address.

To configure MAC address validation for an IP demux interface:

1. In configuration mode, go to the `[edit interfaces demux0 unit` *`logical-unit-number`*`]` hierarchy level:

```
[edit]
user@host# edit interfaces demux0 unit logical-unit-number
```

2. Configure the protocol family for the interface.

```
[edit interfaces demux0 unit logical-unit-number]
user@host# edit family family
```

3. Configure the `mac-validate` statement to validate source MAC address with loose or strict options.

```
[edit interfaces demux0 unit logical-unit-number family family]
user@host# set mac-validate (loose | strict)
```

4. Save the configuration and move to top of the hierarchy level.

```
[edit interfaces demux0 unit logical-unit-number  family family]
user@host# commit
user@host# top
```

Configuring a VLAN Demultiplexing Interface

IN THIS SECTION

- Configuring a VLAN Demux Underlying Interface | 255
- Configuring the VLAN Demux Interface | 257
- Configuring MAC Address Validation on Static VLAN Demux Interfaces | 260
- Verifying a Demux Interface Configuration | 261

Demultiplexing (demux) interfaces are logical interfaces that share a common, underlying interface. You can configure IP demultiplexing interfaces or VLAN demultiplexing interfaces.

To configure a VLAN demux interface, you must configure the demux prefixes that are used by the underlying interface and then configure the VLAN demultiplexing interface as explained by the following tasks:

Configuring a VLAN Demux Underlying Interface

A VLAN demux interface uses an underlying logical interface to receive packets. To determine which VLAN demux interface to use, the VLAN ID is matched against that which the underlying interface receives.

NOTE: VLAN demux interfaces currently support only Gigabit Ethernet, Fast Ethernet, 10-Gigabit Ethernet, and aggregated Ethernet underlying interfaces.

VLAN demux subscriber interfaces over aggregated Ethernet physical interfaces are supported only for MX Series routers that have only Trio MPCs installed. If the router has other MPCs in addition to Trio MPCs, the CLI accepts the configuration but errors are reported when the subscriber interfaces are brought up

To configure a logical interface as a VLAN demux underlying interface with demux source:

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

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

2. Configure the interface as fe-*x*/ *y*/z and the logical interface with the `unit` option.

   ```
   [edit interfaces]
   user@host# edit fe-x/y/z unit logical-unit-number unit logical-unit-number
   ```

3. Configure the VLAN ID. The VLAN ID is used to determine which VLAN demux interface to use, that is the VLAN ID is matched against that which the underlying interface receives.

   ```
   [edit interfaces fe-x/y/z unit logical-unit-number]
   user@host# set vlan-id number
   ```

4. Configure the logical demux source family type on the VLAN demux underlying interface as inet or inet6.

   ```
   [edit interfaces fe-x/y/z unit logical-unit-number]
   user@host# set demux-source (inet | inet6)
   ```

5. Save the configuration and move to top of the hierarchy level.

   ```
   [edit interfaces fe-x/y/z unit logial-unit-number]
   user@host# commit
   user@host# top
   ```

To configure a logical interface as a VLAN demux underlying interface with demux destination:

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

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

2. Configure the interface as fe-*x*/ *y*/z and the logical interface with the `unit` option.

```
[edit interfaces]
user@host# edit fe-x/y/z unit logical-unit-number unit logical-unit-number
```

3. Configure the VLAN ID. The VLAN ID is used to determine which VLAN demux interface to use, that is the VLAN ID is matched against that which the underlying interface receives.

```
[edit interfaces fe-x/y/z unit logical-unit-number]
user@host# set vlan-id number
```

4. Configure the logical demux destination family type on the VLAN demux underlying interface as inet or inet6.

```
[edit interfaces fe-x/y/z unit logical-unit-number]
user@host# set demux-destination (inet | inet6)
```

5. Save the configuration and move to top of the hierarchy level.

```
[edit interfaces fe-x/y/z unit logial-unit-number]
user@host# commit
user@host# top
```

Configuring the VLAN Demux Interface

You can configure one or more logical demux source prefixes or destination prefixes after specifying an underlying interface for the static demux interface to use. This underlying interface must reside on the same logical system as the demux interface.

You configure demux prefixes for use by the underlying interface. The demux prefixes can represent individual hosts or networks. For a given demux interface unit, you can configure either demux source prefix or demux destination prefixes but not both.

You can choose not to configure a demux source prefix or a demux destination prefix. This type of configuration results in a transmit-only interface

To configure VLAN demux interface with demux source prefix:

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

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

2. Configure the interface as a logical demux interface (for example, demux0 interface) and configure the logical interface with the `unit` statement.

   > **NOTE**: You can configure only one demux0 interface per chassis, but you can define logical demux interfaces on top of it (for example, demux0.1, demux0.2, and so on).

   ```
   [edit interfaces]
   user@host# edit demux0 unit logical-unit-number
   ```

3. Configure the underlying interface on which the demux interface is running under the `demux-options` statement.

   ```
   [edit interfaces demux0 unit logical-unit-number]
   user@host# set demux-options underlying-interface interface-name
   ```

4. Configure the protocol family for the interface.

   ```
   [edit interfaces demux0 unit logical-unit-number]
   user@host# edit family family
   ```

5. Configure one or more logical demux source prefixes. The prefixes are matched against the source address of packets that the underlying interface receives. When a match occurs, the packet is processed as if it was received on the demux interface.

   ```
   [edit interfaces demux0 unit logical-unit-number family family]
   user@host# set demux-source source-prefix
   ```

6. Save the configuration and move to top of the hierarchy level.

```
[edit interfaces demux0 unit logical-unit-number]
user@host# commit
user@host# top
```

To configure VLAN demux interface with demux destination prefix:

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

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

2. Configure the interface as a logical demux interface (for example, demux0 interface) and configure the logical interface with the `unit` statement.

> **NOTE**: You can configure only one demux0 interface per chassis, but you can define logical demux interfaces on top of it (for example, demux0.1, demux0.2, and so on).

```
[edit interfaces]
user@host# edit demux0 unit logical-unit-number
```

3. Configure the underlying interface on which the demux interface is running under the `demux-options` statement.

```
[edit interfaces demux0 unit logical-unit-number]
user@host# set demux-options underlying-interface interface-name
```

4. Configure the protocol family for the interface.

```
[edit interfaces demux0 unit logical-unit-number]
user@host# edit family family
```

5. Configure one or more logical demux destination prefixes. The prefixes are matched against the destination address of packets that the underlying interface receives. When a match occurs, the packet is processed as if it was received on the demux interface.

```
[edit interfaces demux0 unit logical-unit-number family family]
user@host# set demux-destination destination-prefix
```

6. Save the configuration and move to top of the hierarchy level.

```
[edit interfaces demux0 unit logical-unit-number]
user@host# commit
user@host# top
```

Configuring MAC Address Validation on Static VLAN Demux Interfaces

MAC address validation enables the router to validate that received packets contain a trusted IP source and an Ethernet MAC source address.

To configure MAC address validation for a VLAN demux interface:

1. In configuration mode, go to the `[edit interfaces demux0 unit logical-unit-number]` hierarchy level:

```
[edit]
user@host# edit interfaces demux0 unit logical-unit-number
```

2. Configure the protocol family for the interface.

```
[edit interfaces demux0 unit logical-unit-number]
user@host# edit family family
```

3. Configure the `mac-validate` statement to validate source MAC address with loose or strict options.

```
[edit interfaces demux0 unit logical-unit-number family family]
user@host# set mac-validate (loose | strict)
```

4. Save the configuration and move to top of the hierarchy level.

```
[edit interfaces demux0 unit logical-unit-number  family family]
user@host# commit
user@host# top
```

Verifying a Demux Interface Configuration

IN THIS SECTION

- Purpose | 261
- Action | 261

Purpose

Check the configuration of a demux interface and its underlying interface when the following are configured:

- Two VLANs are configured, where each VLAN consists of two IP demux interfaces.
- One VLAN demultiplexes based on the source address
- The other VLAN demultiplexes based on the destination address.

Action

From configuration mode on the MX Series router, run the `show interfaces fe-0/0/0` and `show interfaces demux0` configuration mode commands.

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

vlan-tagging;
unit 100 {
    vlan-id 100;
    demux-source inet; # Enable demux of inet prefixes
    family inet {
        address 10.1.1.1/24;
        filter {
            input vlan1-primary-in-filter;
            output vlan1-primary-out-filter;
        }
        mac-validate loose;
    }
}
unit 200 {
    vlan-id 200;
    demux-destination inet; # Enable demux of inet using destination addresses
    family inet {
        address 20.1.1.1/24;
    }
}
unit 300 {
    vlan-id 300;
    demux-source inet; # Enable demux of inet using source addresses
    family inet {
        address 20.1.2.1/24;
    }
}
```

```
user@host> show interfaces demux0

unit 101 {
    description vlan1-sub1;
    demux-options {
        underlying-interface fe-0/0/0.100;
    }
    family inet {
        demux-source 10.1.1.0/24;
        filter {
            input vlan1-sub1-in-filter;
            output vlan1-sub1-out-filter;
        }
    mac-validate loose;
    }
}
unit 102 {
    description vlan1-sub2;
    demux-options {
        underlying-interface fe-0/0/0.100;
    }
    family inet {
        demux-source {
            10.1.0.0/16;
            10.2.1.0/24;
        }
        filter {
            input vlan1-sub2-in-filter;
            output vlan1-sub2-out-filter;
        }
        mac-validate loose;
    }
}
unit 202 {
    description vlan2-sub2;
    demux-options {
        underlying-interface fe-0/0/0.200;
    }
    family inet {
        demux-destination 100.1.2.0/24;
    }
}
unit 302 {
    description vlan2-sub2;
    demux-options {
        underlying-interface fe-0/0/0.300;
    }
    family inet {
        demux-source 100.1.2.0/24;
    }
}
```

Loopback Interfaces

IN THIS SECTION

- Loopback Interface Overview | **264**
- Loopback Interface Configuration | **265**

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.

Junos OS Evolved supports two different filters to control the flow of local packets: one for network control traffic (loopback traffic) and one for management traffic. For additional information, see Top Differences Between Junos OS Evolved and Junos OS.

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.

- Junos OS 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 | 265
- Example: Configure Two Addresses on the Loopback Interface with Host Routes | 267
- Example: Configure Two Addresses on the Loopback Interface with Subnetwork Routes | 267
- Example: Configure an IPv4 and an IPv6 Address on the Loopback Interface with Subnetwork Routes | 268

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 more information, see the Junos OS VPNs Library for Routing Devices.

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 28: 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;
        }
    }
}
```

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

In some instances, you may need to advertise a subnetwork route as internal rather than a Type 5 route for an redistributed static route using OSPF. In this scenario, you may want to configure subnetwork routes on the loopback interface, as shown in the following example:

```
[edit]
user@host# edit interfaces lo0 unit 0 family inet
[edit interfaces lo0 unit 0 family inet]
user@host# set address 10.2.0.1/16
[edit interfaces lo0 unit 0 family inet]
user@host# set address 192.16.0.1/24
[edit interfaces lo0 unit 0 family inet]
user@host# top
[edit]
user@host# show
interfaces {
    lo0 {
        unit 0 {
            family inet {
                10.2.0.1/16;
                192.16.0.1/24;
            }
        }
    }
}
```

Example: Configure an IPv4 and an IPv6 Address on the Loopback Interface with Subnetwork Routes

In the following example, the user configures an IPv4 and an IPv6 address on the loopback interface with subnetwork routes:

```
[edit]
user@host# edit interfaces lo0 unit 0 family inet
[edit interfaces lo0 unit 0 family inet]
user@host# set address 192.16.0.1/24
[edit interfaces lo0 unit 0 family inet]
user@host# up
[edit interfaces lo0 unit 0 family]
user@host# edit interfaces lo0 unit 0 family inet6
[edit interfaces lo0 unit 0 family inet6]
user@host# set address 2001:db8::200:f8ff:fe75:50df/64
[edit interfaces lo0 unit 0 family inet6]
user@host# top
[edit]
user@host# show
interfaces {
    lo0 {
        unit 0 {
            family inet {
                192.16.0.1/24;
            }
            family inet6 {
                2001:db8::200:f8ff:fe75:50df/64;
            }
        }
    }
}
```

Serial Interfaces

IN THIS SECTION

- Serial Interfaces Overview | 269
- Configure the Serial Line Protocol | 275
- Configure the Serial Clocking Mode | 280
- Configure the Serial Signal Handling | 283
- Configure the Serial DTR Circuit | 286
- Configure Serial Signal Polarities | 287
- Configure Serial Loopback Capability | 288
- Configure Serial Line Encoding | 290
- Configure Serial Interfaces on SRX devices | 291

This topic discusses about the serial interfaces, and how to configure serial line protocol, serial clocking mode, serial signal handling, serial DTR circuit, serial signal polarities, serial loopback capability, and serial line encoding.

Serial Interfaces Overview

IN THIS SECTION

- Serial Transmissions | 271
- 8-Port Synchronous Serial GPIM on SRX devices | 272
- Benefits of Serial Interfaces | 275

Devices that communicate over a serial interface are divided into two classes: data terminal equipment (DTE) and data circuit-terminating equipment (DCE). Juniper Networks Serial Physical Interface Cards (PICs) have two ports per PIC and support full-duplex data transmission. These PICs support DTE mode only. On the Serial PIC. Table 29 on page 270 specifies the key details of the serial interfaces.

Table 29: Serial Interface Details

Interface Details	Description
Interface name	Serial interface
Supported on	For information about platforms support, see hardware compatibility tool (HCT).
Standards to configure serial interfaces type	• EIA-530—An Electronics Industries Alliance (EIA) standard. • V.35—An ITU-T standard. • X.21—An ITU-T standard. • RS-232 —A Recommended Standard (RS) known as EIA-232. • RS-422/449 —A Recommended Standard (RS). The RS-449 standard (known as EIA-449) is compatible with RS-422 signal levels.
Features supported	• Serial transmissions • Signal polarity • Serial clocking modes • Serial Line protocol
Logical properties	There are no serial interface-specific logical properties. For information about general logical properties that you can configure, see Configuring Logical Interface Properties. This support on serial interfaces is the same as the existing LFI and MLPPP support on T1 and E1 interfaces.

Serial Transmissions

In basic serial communications, nine signals are critical to the transmission. Each signal is associated with a pin in either the 9-pin or 25-pin connector. Table 30 on page 271 lists and defines serial signals and their sources.

Table 30: Serial Transmission Signals

Signal Name	Definition	Signal Source
TD	Transmitted data	DTE
RD	Received data	DCE
RTS	Request to send	DTE
CTS	Clear to send	DCE
DSR	Data set ready	DCE
Signal Ground	Grounding signal	–
CD	Carrier detect	–
DTR	Data terminal ready	DTE
RI	Ring indicator	–

Serial line protocol guidelines:

- The DCE transmits a DSR signal to the DTE, which responds with a DTR signal. This establishes the link and traffic can pass.
- When the DTE device is ready to receive data:
  - It sets its RTS signal to a marked state all 1s to indicate to the DCE that it can transmit data. If the DTE is not able to receive data—because of buffer conditions, for example—it sets the RTS signal to all 0s.

  - It sets its CTS signal to a marked state to indicate to the DTE that it can transmit data. If the DCE is not able to receive data, it sets the CTS signal to all 0s.

- When you send the information, it transmits data across the transmitted data (TD) lines and receives data across received data (RD) lines:

  - TD line—Line through which the data transmits from a DTE device to a DCE device

  - RD line—Line through which the data transmits from a DCE device to a DTE device

- The wire name does not indicate the direction of data flow.

When a serial port is opens, the DTE device sets its DTR signal to a marked state. Similarly, the DCE sets its DSR signal to a marked state. However, because of the negotiation that takes place with the RTS and CTS signals, the DTR and DSR signals are hardly utilized.

The carrier detect and ring indicator signals detect connections with remote modems and these signals are hardly used.

8-Port Synchronous Serial GPIM on SRX devices

A Gigabit-Backplane *Physical Interface Module* (GPIM) is a network interface card (NIC) that you can install in the front slots of the SRX550 Services Gateway to provide physical connections to a LAN or a WAN. The 8-port synchronous serial GPIM provides the physical connection to serial network media types, receiving incoming packets and transmitting outgoing packets of the network. Besides forwarding packets for processing, the GPIM performs framing and line-speed signaling. This GPIM provides 8 ports that operate in sync mode and supports a line rate of 64 Mbps or 8 Mbps per port.

For information on configuration of 8-Port Serial GPIM, see 8-Port Serial GPIM Basic Configuration.

Features Supported on 8-Port Synchronous Serial GPIM

Table 31 on page 272 lists the features supported on the 8-port synchronous serial GPIM.

Table 31: Supported Features

Features	Description
Operation modes (autoselection based on cable, no configuration required)	• DTE (data terminal equipment) • DCE (data communication equipment)

Table 31: Supported Features *(Continued)*

Features	Description
Clocking	- Tx clock modes - DCE clock (only valid in DTE mode) - Baud clock (internally generated) - Loop clock (external) - Rx clock modes - Baud clock (internally generated) - Loop clock (external)
Clock rates (baud rates)	1.2 KHz to 8.0 MHz **NOTE**: RS-232 serial interfaces might cause an error with a clock rate greater than 200 KHz.
MTU	9192 bytes, default value is 1504 bytes
HDLC features	- Idle flag/fill (0x7e or all ones), default idle flag is (0x7e) - Counters—giants, runts, FCS error, terminate error, align error
Line encoding	NRZ and NRZI
Invert data	Enabled
Line protocol	EIA530/EIA530A, X.21, RS-449, RS-232, V.35
Data cables	Separate cable for each line protocol (both DTE/DCE mode)
Error counters (conformance to ANSI specification)	Enabled

Table 31: Supported Features *(Continued)*

Features	Description
Alarms and defects	• Rx clock absent • Tx clock absent • DCD absent • RTS/CTS absent • DSR/DTR absent
Data signal	Rx clock
Control signals	• To DTE: CTS, DCD, DSR • From DTE: DTR, RTS
Serial autoresync	• Configurable resync duration • Configurable resync interval
Diagnostic features	• Loopback modes—local, remote, and dce-local loopback • Ability to ignore control signals
Layer 2 features	Encapsulation • PPP • Cisco HDLC • Frame Relay • MLPPP • MLFR

Table 31: Supported Features *(Continued)*

Features	Description
SNMP features	SNMP information receivable at each port • IF-MIB - rfc2863a.mib • jnx-chassis.mib
Anticounterfeit check	Enabled

Benefits of Serial Interfaces

- Serial interface are a simple, cost-effective way to connect transmitting and receiving devices or ICs. A serial interface requires fewer conducting wires (often only one) than other interfaces, which eases implementation.
- Serial interfaces support long-distance communication.

Configure the Serial Line Protocol

IN THIS SECTION

- Configure the Serial Line Protocol | 275
- Serial Interface Default Settings | 276

Configure the Serial Line Protocol

By default, serial interfaces use the EIA-530 line protocol. You can configure each port on the PIC independently to use one of the following line protocols:

- EIA-530
- V.35
- X.21

To configure the serial line protocol:

Include the `line-protocol` statement, specifying the `eia530`, `v.35`, or `x.21` option:

```
line-protocol protocol;
```

You can include these statements at the following hierarchy levels:

- [edit interfaces se-*pim*/0/*port* serial-options]
- [edit interfaces se-*fpc*/*pic*/*port* serial-options]

For more information about serial interfaces, see the following sections:

Serial Interface Default Settings

IN THIS SECTION

- Serial Interface Default Settings | **276**
- Invalid Serial Interface Statements | **278**

Serial Interface Default Settings

EIA-530 Interface Default Settings

If you do not include the `line-protocol` statement or if you explicitly configure the default EIA-530 line protocol, the default settings are as follows:

```
dce-options | dte-options {
    cts normal;
    dcd normal;
    dsr normal;
    dtr normal;
    rts normal;
    tm normal;
}
clock-rate 16.384mhz;
clocking-mode loop;
cts-polarity positive;
```

```
dcd-polarity positive;
dsr-polarity positive;
dtr-circuit balanced;
dtr-polarity positive;
encoding nrz;
rts-polarity positive;
tm-polarity positive;
```

NOTE: On M Series routers, you can set the DCE clocking mode for EIA-530 interfaces and commit. An error message is not displayed and the CLI is not blocked.

You can include the *line-protocol* statement at the following hierarchy levels:

- `[edit interfaces se-`*`pim`*`/0/`*`port`*` serial-options]`
- `[edit interfaces se-`*`fpc`*`/`*`pic`*`/`*`port`*` serial-options]`

V.35 Interface Default Settings

If you include the `line-protocol v.35` statement, the default settings are as follows:

```
dce-options | dte-options {
    cts normal;
    dcd normal;
    dsr normal;
    dtr normal;
    rts normal;
}
clock-rate 16.384mhz;
clocking-mode loop;
cts-polarity positive;
dcd-polarity positive;
dsr-polarity positive;
dtr-circuit balanced;
dtr-polarity positive;
encoding nrz;
rts-polarity positive;
```

You can include the *line-protocol* statement at the following hierarchy levels:

- `[edit interfaces se-`*`pim`*`/0/`*`port`*` serial-options]`

- `[edit interfaces se-fpc/pic/port serial-options]`

X.21 Interface Default Settings

If you include the `line-protocol x.21` statement, the default settings are as follows:

```
dce-options | dte-options {
    control-signal normal;
    indication normal;
}
clock-rate 16.384mhz;
clocking-mode loop;
control-polarity positive;
encoding nrz;
indication-polarity positive;
```

You can include the *line-protocol* statement at the following hierarchy levels:

- `[edit interfaces se-pim/0/port serial-options]`
- `[edit interfaces se-fpc/pic/port serial-options]`

Invalid Serial Interface Statements

The following sections show the invalid configuration statements for each type of serial interface. If you include the following statements in the configuration, an error message indicates the location of the error and the configuration is not activated.

Invalid EIA-530 Interface Statements

If you do not include the `line-protocol` statement or if you explicitly configure the default EIA-530 line protocol, the following statements are invalid:

```
dce-options | dte-options {
    control-signal (assert | de-assert | normal);
    indication (ignore | normal | require);
}
control-polarity (negative | positive);
indication-polarity (negative | positive);
```

You can include the *line-protocol* statement at the following hierarchy levels:

- [edit interfaces se-*pim*/0/*port* serial-options]
- [edit interfaces se-*fpc*/*pic*/*port* serial-options]

Invalid V.35 interface Statements

If you include the line-protocol v.35 statement, the following statements are invalid:

```
dce-options | dte-options {
    control-signal (assert | de-assert | normal);
    indication (ignore | normal | require);
    tm (ignore | normal | require);
}
control-polarity (negative | positive);
indication-polarity (negative | positive);
loopback (dce-local | dce-remote);
tm-polarity (negative | positive);
```

You can include the *line-protocol* statement at the following hierarchy levels:

- [edit interfaces se-*pim*/0/*port* serial-options]
- [edit interfaces se-*fpc*/*pic*/*port* serial-options]

Invalid X.21 Interface Statements

If you include the line-protocol x.21 statement, the following statements are invalid:

```
dce-options | dte-options {
    cts (ignore | normal | require);
    dcd (ignore | normal | require);
    dsr (ignore | normal | require);
    dtr (assert | de-assert | normal);
    rts (assert | de-assert | normal);
    tm (ignore | normal | require);
}
clocking-mode (dce | internal);
cts-polarity (negative | positive);
dce-polarity (negative | positive);
dsr-polarity (negative | positive);
dtr-circuit (balanced | unbalanced);
dtr-polarity (negative | positive);
```

```
loopback (dce-local | dce-remote);
rts-polarity (negative | positive);
tm-polarity (negative | positive);
```

You can include the *line-protocol* statement at the following hierarchy levels:

- `[edit interfaces se-`*`pim`*`/0/`*`port`*` serial-options]`
- `[edit interfaces se-`*`fpc`*`/`*`pic`*`/`*`port`*` serial-options]`

Configure the Serial Clocking Mode

IN THIS SECTION

- Configure the Serial Clocking Mode | **280**
- Invert the Serial Interface Transmit Clock | **281**
- Configure the DTE Clock Rate | **282**

Configure the Serial Clocking Mode

By default, serial interfaces use loop clocking mode. For EIA-530 and V.35 interfaces, you can configure each port on the PIC independently to use loop, DCE, or internal clocking mode. For X.21 interfaces, only loop clocking mode is supported.

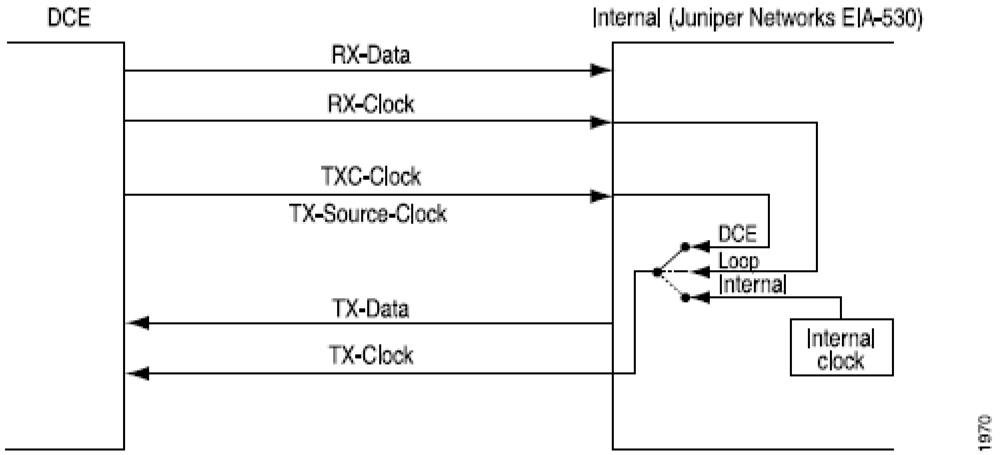
The three clocking modes work as follows:

- Loop clocking mode—Uses the DCE's RX clock to clock data from the DCE to the DTE.
- DCE clocking mode—Uses the TXC clock, which is generated by the DCE specifically to be used by the DTE as the DTE's transmit clock.
- Internal clocking mode—Also known as line timing, uses an internally generated clock. You can configure the speed of this clock by including the `clock-rate` statement at the `[edit interfaces se-`*`pim`*`/0/`*`port`*` serial-options]` or `[edit interfaces se-`*`fpc`*`/`*`pic`*`/`*`port`*` dte-options]` hierarchy levels. For more information about the DTE clock rate, see "Configure the DTE Clock Rate" on page 282.

Note that DCE clocking mode and loop clocking mode use external clocks generated by the DCE.

Figure 1 shows the clock sources of loop, DCE, and internal clocking modes.

Figure 14: Serial Interface Clocking Mode



To configure the clocking mode of a serial interface, include the `clocking-mode` statement:

```
clocking-mode (dce | internal | loop);
```

You can include this statement at the following hierarchy levels:

- `[edit interfaces se-pim/0/port serial-options]`
- `[edit interfaces se-fpc/pic/port serial-options]`

Invert the Serial Interface Transmit Clock

When an externally timed clocking mode (DCE or loop) is used, long cables might introduce a phase shift of the DTE-transmitted clock and data. At high speeds, this phase shift might cause errors. Inverting the transmit clock corrects the phase shift, thereby reducing error rates.

By default, the transmit clock is not inverted. To invert the transmit clock, include the `transmit-clock invert` statement:

```
transmit-clock invert;
```

You can include this statement at the following hierarchy levels:

- `[edit interfaces se-pim/0/port serial-options]`
- `[edit interfaces se-fpc/pic/port serial-options]`

Configure the DTE Clock Rate

By default, the serial interface has a clock rate of 16.384 MHz. For EIA-530 and V.35 interfaces with internal clocking mode configured, you can configure the clock rate.

To configure the clock rate, include the `clock-rate` statement:

```
clock-rate rate;
```

You can include this statement at the following hierarchy levels:

- `[edit interfaces se-pim/0/port serial-options]`
- `[edit interfaces se-fpc/pic/port serial-options]`

You can configure the following interface speeds:

- 2.048 MHz
- 2.341 MHz
- 2.731 MHz
- 3.277 MHz
- 4.096 MHz
- 5.461 MHz
- 8.192 MHz
- 16.384 MHz

Although the serial interface is intended for use at the default rate of 16.384 MHz, you might need to use a slower rate if any of the following conditions prevail:

- The interconnecting cable is too long for effective operation.
- The interconnecting cable is exposed to an extraneous noise source that might cause an unwanted voltage in excess of +1 volt measured differentially between the signal conductor and circuit common at the load end of the cable, with a 50-ohm resistor substituted for the generator.
- You need to minimize interference with other signals.
- You need to invert signals.

For detailed information about the relationship between signaling rate and interface cable distance, see the following standards:

- EIA-422-A, *Electrical Characteristics of Balanced Voltage Digital Interface Circuits*
- EIA-423-A, *Electrical Characteristics of Unbalanced Voltage Digital Interface Circuits*

Configure the Serial Signal Handling

By default, normal signal handling is enabled for all signals. For each signal, the `normal` option applies to the normal signal handling for that signal, as defined by the following standards:

- TIA/EIA Standard 530
- ITU-T Recommendation V.35
- ITU-T Recommendation X.21

Table 32 on page 283 shows the serial interface modes that support each signal type.

Table 32: Signal Handling by Serial Interface Type

Signal	Serial Interfaces
From-DCE signals	
Clear to send (CTS)	EIA-530 and V.35
Data carrier detect (DCD)	EIA-530 and V.35
Data set ready (DSR)	EIA-530 and V.35
Indication	X.21 only
Test mode (TM)	EIA-530 only
To-DCE signals	
Control signal	X.21 only
Data transfer ready (DTR)	EIA-530 and V.35

Table 32: Signal Handling by Serial Interface Type *(Continued)*

Signal	Serial Interfaces
Request to send (RTS)	EIA-530 and V.35

You configure serial interface signal characteristics by including the `dce-options` or `dte-options` statement:

```
dce-options |dte-options {
    control-signal (assert | de-assert | normal);
    cts (ignore | normal | require);
    dcd (ignore | normal | require);
    dsr (ignore | normal | require);
    dtr signal-handling-option;
    ignore-all;
    indication (ignore | normal | require);
    rts (assert | de-assert | normal);
    tm (ignore | normal | require);
}
```

You can include these statements at the following hierarchy levels:

- `[edit interfaces se-`*`pim`*`/0/`*`port`*` serial-options]`
- `[edit interfaces se-`*`fpc`*`/`*`pic`*`/`*`port`*` serial-options]`

For EIA-530 and V.35 interfaces, configure to-DCE signals by including the `dtr` and `rts` statements, specifying the `assert`, `de-assert`, or `normal` option:

```
dtr (assert | de-assert | normal);
rts (assert | de-assert | normal);
```

For X.21 interfaces, configure to-DCE signals by including the `control-signal` statement, specifying the `assert`, `de-assert`, or `normal` option:

```
control-signal (assert | de-assert | normal);
```

Assertion is when the positive side of a given signal is at potential high-level output voltage (Voh), while the negative side of the same signal is at potential low-level output voltage (Vol). *Deassertion* is when the positive side of a given signal is at potential Vol, while the negative side of the same signal is at potential Voh.

For the DTR signal, you can configure normal signal handling using the signal for automatic resynchronization by including the `dtr` statement, and specifying the `auto-synchronize` option:

```
dtr {
    auto-synchronize {
        duration milliseconds;
        interval seconds;
    }
}
```

The pulse duration of resynchronization can be from 1 through 1000 milliseconds. The offset interval for resynchronization can be from 1 through 31 seconds.

For EIA-530 and V.35 interfaces, configure from-DCE signals by including the `cts`, `dcd`, and `dsr` statements, specifying the `ignore`, `normal`, or `require` option:

```
cts (ignore | normal | require);
dcd (ignore | normal | require);
dsr (ignore | normal | require);
```

For X.21 interfaces, configure from-DCE signals by including the `indication` statement, specifying the `ignore`, `normal`, or `require` option:

```
indication (ignore | normal | require);
```

For EIA-530 interfaces only, you can configure from-DCE test-mode (TM) signaling by including the `tm` statement, specifying the `ignore`, `normal`, or `require` option:

```
tm (ignore | normal | require);
```

To specify that the from-DCE signal must be asserted, include the `require` option in the configuration. To specify that the from-DCE signal must be ignored, include the `ignore` option in the configuration.

NOTE: For V.35 and X.21 interfaces, you cannot include the `tm` statement in the configuration.

For X.21 interfaces, you cannot include the `cts`, `dcd`, `dsr`, `dtr`, and `rts` statements in the configuration.

For EIA-530 and V.35 interfaces, you cannot include the `control-signal` and `indication` statements in the configuration.

For a complete list of serial options statements that are not supported by each serial interface mode, see "Invalid Serial Interface Statements" on page 269.

To return to the default normal signal handling, delete the `require`, `ignore`, `assert`, `de-assert`, or `auto-synchronize` statement from the configuration, as shown in the following example:

```
[edit]
user@host# delete interfaces se-fpc/pic/port dte-options control-leads cts require
```

To explicitly configure normal signal handling, include the `control-signal` statement with the `normal` option:

```
control-signal normal;
```

You can configure the serial interface to ignore all control leads by including the `ignore-all` statement:

```
ignore-all;
```

You can include the `ignore-all` statement in the configuration only if you do not explicitly enable other signal handling options at the `[edit interfaces se-pim/0/port serial-options dce-options]` or `[edit interfaces se-fpc/pic/port serial-options dte-options]` hierarchy levels.

You can include the `control-signal`, `cts`, `dcd`, `dsr`, `dtr`, `indication`, `rts`, and `tm` statements at the following hierarchy levels:

- `[edit interfaces se-pim/0/port serial-options dte-options]`
- `[edit interfaces se-fpc/pic/port serial-options dte-options]`

Configure the Serial DTR Circuit

A balanced circuit has two currents that are equal in magnitude and opposite in phase. An unbalanced circuit has one current and a ground; if a pair of terminals is unbalanced, one side is connected to electrical ground and the other carries the signal. By default, the DTR circuit is balanced.

For EIA-530 and V.35 interfaces, configure the DTR circuit by including the `dtr-circuit` statement:

```
dtr-circuit (balanced | unbalanced);
```

You can include this statement at the following hierarchy levels:

- [edit interfaces se-*pim*/0/*port* serial-options]
- [edit interfaces se-*fpc*/*pic*/*port* serial-options]

Configure Serial Signal Polarities

Serial interfaces use a differential protocol signaling technique. Of the two serial signals associated with a circuit, the one referred to as the A signal is denoted with a plus sign, and the one referred to as the B signal is denoted with a minus sign; for example, DTR+ and DTR–. If DTR is low, then DTR+ is negative with respect to DTR–. If DTR is high, then DTR+ is positive with respect to DTR–.

By default, all signal polarities are positive. You can reverse this polarity on a Juniper Networks serial interface. You might need to do this if signals are miswired as a result of reversed polarities.

For EIA-530 and V.35 interfaces, configure signal polarities by including the `cts-polarity`, `dcd-polarity`, `dsr-polarity`, `dtr-polarity`, `rts-polarity`, and `tm-polarity` statements:

```
cts-polarity (negative | positive);
dcd-polarity (negative | positive);
dsr-polarity (negative | positive);
dtr-polarity (negative | positive);
rts-polarity (negative | positive);
tm-polarity (negative | positive);
```

You can include these statements at the following hierarchy levels:

- [edit interfaces se-*pim*/0/*port* serial-options]
- [edit interfaces se-*fpc*/*pic*/*port* serial-options]

For X.21 interfaces, configure signal polarities by including the `control-polarity` and `indication-polarity` statements:

```
control-polarity (negative | positive);
indication-polarity (negative | positive);
```

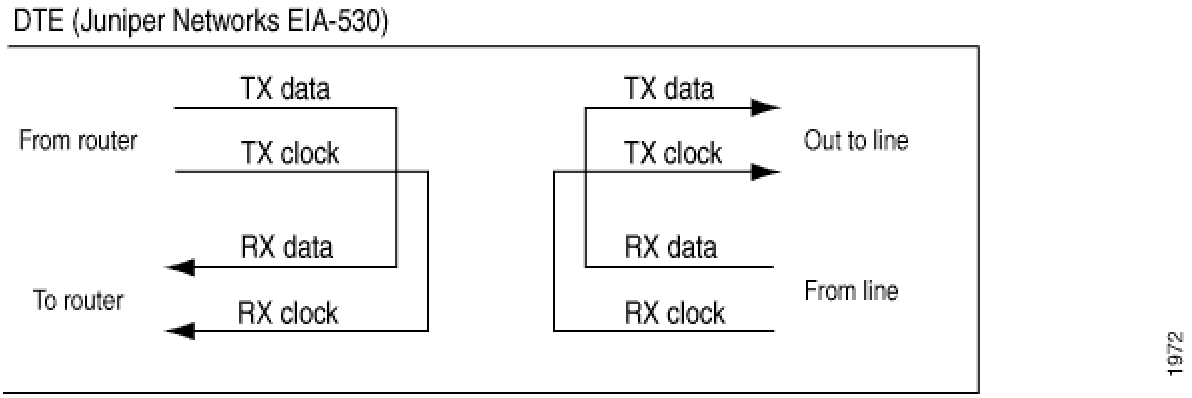
You can include these statements at the following hierarchy levels:

- `[edit interfaces se-`*`pim`*`/0/`*`port`*` serial-options]`
- `[edit interfaces se-`*`fpc`*`/`*`pic`*`/`*`port`*` serial-options]`

Configure Serial Loopback Capability

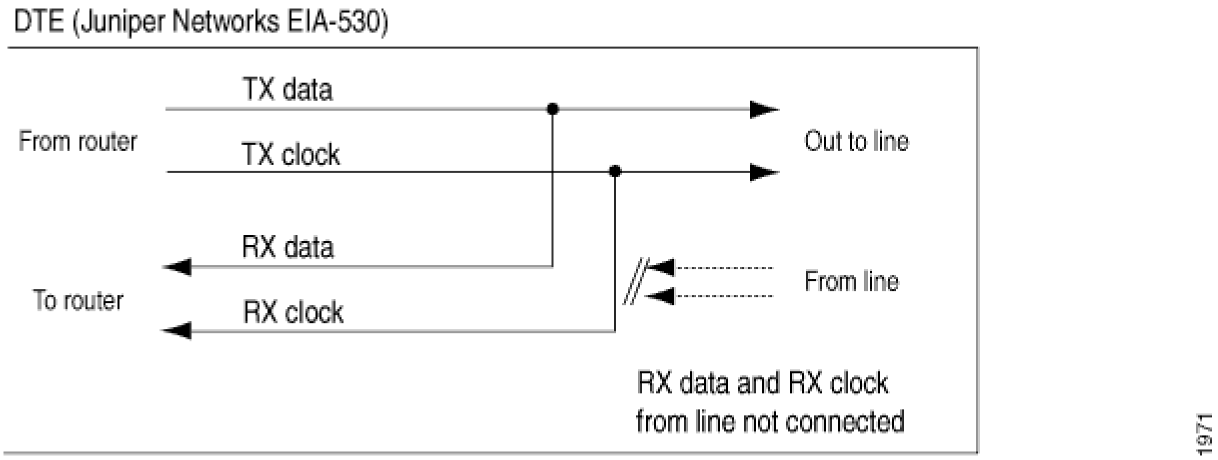
From the router, remote line interface unit (LIU) loopback loops the TX (transmit) data and TX clock back to the router as RX (receive) data and RX clock. From the line, LIU loopback loops the RX data and RX clock back out the line as TX data and TX clock, as shown in Figure 15 on page 288.

Figure 15: Serial Interface LIU Loopback



DCE local and DCE remote control the EIA-530 interface-specific signals for enabling local and remote loopback on the link partner DCE. Local loopback is shown in Figure 16 on page 289.

Figure 16: Serial Interface Local Loopback



For EIA-530 interfaces, you can configure DCE local, DCE remote, local, and remote (LIU) loopback capability.

For V.35, you can configure remote LIU and local loopback capability. DCE local and DCE remote loopbacks are not supported on V.35 and X.21 interfaces. Local and remote loopbacks are not supported on X.21 interfaces.

To configure the loopback capability on a serial interface, include the `loopback` statement, specifying the `dce-local`, `dce-remote`, `local`, or `remote` option:

```
loopback mode;
```

You can include this statement at the following hierarchy levels:

- `[edit interfaces se-pim/0/port serial-options]`
- `[edit interfaces se-fpc/pic/port serial-options]`

To disable the loopback capability, remove the `loopback` statement from the configuration:

```
[edit]
user@host# delete interfaces se-fpc/pic/port serial-options loopback
```

You can determine whether there is an internal or external problem by checking the error counters in the output of the `show interface se-`*fpc/pic/port* `extensive` command:

```
user@host> show interfaces se-fpc/pic/port extensive
```

To Configure Serial Loopback Capability:

1. To determine the source of a problem, loop the packets on the local router, the local DCE, the remote DCE, and the remote line interface unit (LIU).
2. To do this, include the `no-keepalives` and `encapsulation cisco-hdlc` statements at the `[edit interfaces se-`*fpc/pic/port*`]` hierarchy level, and the `loopback local` option at the `[edit interfaces se-`*pim*`/0/`*port* `serial-options]` or `[edit interfaces se-`*fpc/pic/port* `serial-options]` hierarchy level. With this configuration, the link stays up, so you can loop ping packets to a remote router. The `loopback local` statement causes the interface to loop within the PIC just before the data reaches the transceiver.

   ```
   [edit interfaces]
   se-1/0/0 {
       no-keepalives;
       encapsulation cisco-hdlc;
       serial-options {
           loopback local;
       }
       unit 0 {
           family inet {
               address 10.100.100.1/24;
           }
       }
   }
   ```

Configure Serial Line Encoding

By default, serial interfaces use non-return to zero (NRZ) line encoding. You can configure non-return to zero inverted (NRZI) line encoding if necessary.

To have the interface use NRZI line encoding, include the `encoding` statement, specifying the `nrzi` option:

```
encoding nrzi;
```

To explicitly configure the default NRZ line encoding, include the `encoding` statement, specifying the `nrz` option:

```
encoding nrz;
```

You can include this statement at the following hierarchy levels:

- `[edit interfaces se-`*`pim`*`/0/`*`port`*` serial-options]`
- `[edit interfaces se-`*`fpc`*`/`*`pic`*`/`*`port`*` serial-options]`

When setting the line encoding parameter, you must set the same value for paired ports. Ports 0 and 1 must share the same value.

Configure Serial Interfaces on SRX devices

IN THIS SECTION

- Basic Serial Interface Configuration | **292**
- Delete the Serial Interface | **293**
- Example: Configure serial interface on 8-Port Synchronous Serial GPIM | **293**
- Verification | **299**

In this example you learn how to complete the initial configuration on a serial interface, how to delete a serial interface and how to configure serial interface 8-Port Synchronous Serial GPIM.

For information on installation of a serial PIM in the SRX Series Firewall, see *SRX Series Firewalls for the Branch Physical Interface Modules Hardware Guide*.

In this example:

1. Create a new interface on a serial interface, `se-1/0/0`.
2. Set the encapsulation type to ppp and create the basic configuration for `se-1/0/0`.
3. Set the logical interface to 0 and logical unit number can range from 0 through 16,384.
4. Enter additional values for properties you need to configure on the logical interface, such as logical encapsulation or protocol family.

5. Set IPv4 address 10.10.10.10/24 on `se-1/0/0`.

When you delete the `se-1/0/0` interface, the interface is disabled and removed from the software configuration. Network interfaces remain physically present, and their identifiers continue to appear on J-Web pages.

Basic Serial Interface Configuration

In this example, you create a serial interface called se-1/0/0 and set the encapsulation type to ppp. To quickly configure this example, use CLI quick configuration at the `[edit]` hierarchy level, and commit from configuration mode.

```
set interfaces se-1/0/0 encapsulation ppp unit 0 family inet address 10.10.10.10/24
```

To configure the serial interface, `se-1/0/0`:

1. Create the interface.

   ```
   [edit]
   user@host# edit interfaces se-1/0/0
   ```

2. Set the encapsulation type for `se-1/0/0`.

   ```
   [edit interfaces se-1/0/0]
   user@host# set encapsulation ppp
   ```

3. Add logical interfaces.

   ```
   [edit interfaces se-1/0/0]
   user@host# edit unit 0
   ```

4. Specify an IPv4 address for the interface.

   ```
   [edit interfaces se-1/0/0 unit 0]
   user@host# set family inet address 10.10.10.10/24
   ```

After completing the configuration successfully, view the parameters by using the `show interfaces se-1/0/0` command.

Delete the Serial Interface

In this example, you delete a serial interface `se-1/0/0`. No configuration beyond device initialization is required before configuring an interface.

To delete the serial interface, `se-1/0/0`:

1. Specify the interface you want to delete.

```
[edit]
user@host# delete interfaces se-1/0/0
```

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

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

After completing the configuration successfully, to verify the configuration use the `show interfaces` command.

Example: Configure serial interface on 8-Port Synchronous Serial GPIM

In this example, you can perform a basic back-to-back device configuration with an 8-port synchronous serial GPIM. The devices are shown as both data communication equipment (DCE) and data terminal equipment (DTE). In certain deployment scenarios, the DTE can be a serial modem or an encryptor or decryptor.

In this scenario, you can configure serial interface using two interfaces. You can configure all ports with different encapsulations, such as Cisco High-Level Data Link Control (HDLC), Frame Relay, and Point-to-Point Protocol (PPP). When Frame Relay is set, then the data link connection identifier (in this example, 111) must also be set. All the eight ports on Device 1 (SRX650) are configured in DTE mode and their respective eight ports on Device 2 (SRX650) are configured in DCE mode.

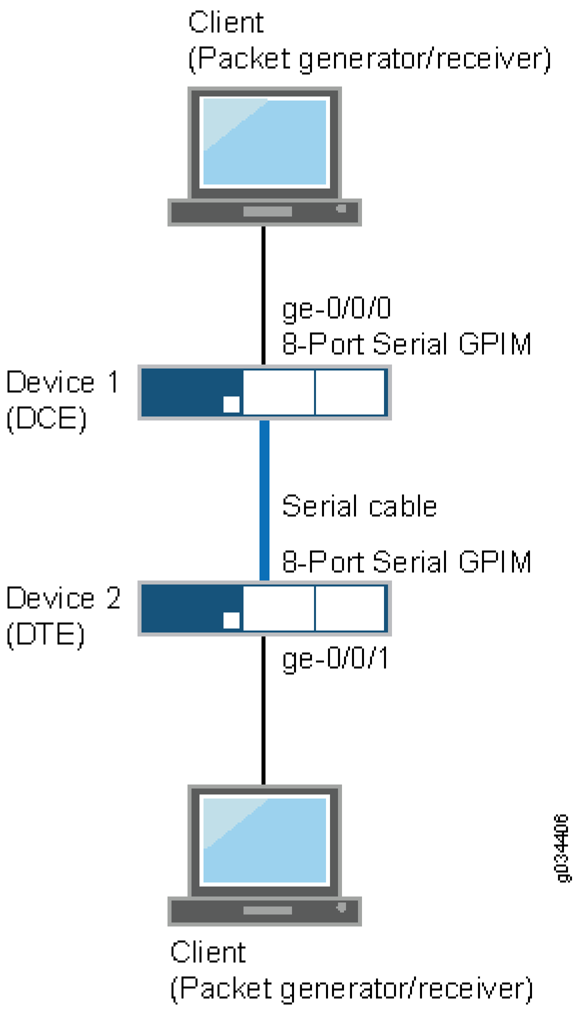
In this example, for device 1:

- Set the encapsulation type to `ppp` and the logical interface to `0`. The logical unit number can range from 0 through 16,384.
- Enter additional values for properties you need to configure on the logical interface, such as logical encapsulation or protocol family.
- Set the IPv4 address to 10.10.10.1/24 on the serial port.

For Device 2, you follow a procedure similar to Device 1, but you set the clocking mode to dce.

Figure 17 on page 294 shows the topology used in this example.

Figure 17: Basic Back-to-Back Device Configuration



To quickly configure this example, CLI at the [edit] hierarchy level:

Device 1

```
set interfaces se-7/0/0 mtu 9192
set interfaces se-7/0/0 encapsulation ppp
set interfaces se-7/0/0 serial-options clocking-mode internal
set interfaces se-7/0/0 unit 0 family inet address 10.10.10.1/24
set interfaces se-7/0/1 mtu 9192
set interfaces se-7/0/1 encapsulation cisco-hdlc
set interfaces se-7/0/1 serial-options clocking-mode internal
set interfaces se-7/0/1 unit 0 family inet address 11.11.11.1/24
set interfaces se-7/0/2 dce
set interfaces se-7/0/2 mtu 9192
set interfaces se-7/0/2 encapsulation frame-relay
set interfaces se-7/0/2 serial-options clocking-mode internal
set interfaces se-7/0/2 unit 0 dlci 111
set interfaces se-7/0/2 unit 0 family inet address 12.12.12.1/24
set interfaces se-7/0/3 mtu 9192
set interfaces se-7/0/3 encapsulation ppp
set interfaces se-7/0/3 serial-options clocking-mode internal
set interfaces se-7/0/3 unit 0 family inet address 13.13.13.1/24
set interfaces se-7/0/4 mtu 9192
set interfaces se-7/0/4 encapsulation cisco-hdlc
set interfaces se-7/0/4 serial-options clocking-mode internal
set interfaces se-7/0/4 unit 0 family inet address 14.14.14.1/24
set interfaces se-7/0/5 dce
set interfaces se-7/0/5 mtu 9192
set interfaces se-7/0/5 encapsulation frame-relay
set interfaces se-7/0/5 serial-options clocking-mode internal
set interfaces se-7/0/5 unit 0 dlci 112
set interfaces se-7/0/5 unit 0 family inet address 15.15.15.1/24
set interfaces se-7/0/6 mtu 9192
set interfaces se-7/0/6 encapsulation cisco-hdlc
set interfaces se-7/0/6 serial-options clocking-mode internal
set interfaces se-7/0/6 unit 0 family inet address 16.16.16.1/24
set interfaces se-7/0/7 mtu 9192
set interfaces se-7/0/7 encapsulation ppp
set interfaces se-7/0/7 serial-options clocking-mode internal
set interfaces se-7/0/7 unit 0 family inet address 17.17.17.1/24
set routing-options static route 21.21.21.0/24 next-hop 10.10.10.2
set routing-options static route 23.23.23.0/24 next-hop 11.11.11.2
set routing-options static route 25.25.25.0/24 next-hop 12.12.12.2
set routing-options static route 27.27.27.0/24 next-hop 13.13.13.2
set routing-options static route 29.29.29.0/24 next-hop 14.14.14.2
set routing-options static route 31.31.31.0/24 next-hop 15.15.15.2
set routing-options static route 33.33.33.0/24 next-hop 16.16.16.2
set routing-options static route 35.35.35.0/24 next-hop 17.17.17.2
```

Device 2

```
set interfaces se-3/0/0 mtu 9192
set interfaces se-3/0/0 encapsulation ppp
set interfaces se-3/0/0 serial-options clocking-mode dce
set interfaces se-3/0/0 unit 0 family inet address 10.10.10.2/24
set interfaces se-3/0/1 mtu 9192
set interfaces se-3/0/1 encapsulation cisco-hdlc
set interfaces se-3/0/1 serial-options clocking-mode dce
set interfaces se-3/0/1 unit 0 family inet address 11.11.11.2/24
set interfaces se-3/0/2 dce
set interfaces se-3/0/2 mtu 9192
set interfaces se-3/0/2 encapsulation frame-relay
set interfaces se-3/0/2 serial-options clocking-mode dce
set interfaces se-3/0/2 unit 0 dlci 111
set interfaces se-3/0/2 unit 0 family inet address 12.12.12.2/24
set interfaces se-3/0/3 mtu 9192
set interfaces se-3/0/3 encapsulation ppp
set interfaces se-3/0/3 serial-options clocking-mode dce
set interfaces se-3/0/3 unit 0 family inet address 13.13.13.2/24
set interfaces se-3/0/4 mtu 9192
set interfaces se-3/0/4 encapsulation cisco-hdlc
set interfaces se-3/0/4 serial-options clocking-mode dce
set interfaces se-3/0/4 unit 0 family inet address 14.14.14.2/24
set interfaces se-3/0/5 dce
set interfaces se-3/0/5 mtu 9192
set interfaces se-3/0/5 encapsulation frame-relay
set interfaces se-3/0/5 serial-options clocking-mode dce
set interfaces se-3/0/5 unit 0 dlci 112
set interfaces se-3/0/5 unit 0 family inet address 15.15.15.2/24
set interfaces se-3/0/6 mtu 9192
set interfaces se-3/0/6 encapsulation cisco-hdlc
set interfaces se-3/0/6 serial-options clocking-mode dce
set interfaces se-3/0/6 unit 0 family inet address 16.16.16.2/24
set interfaces se-3/0/7 mtu 9192
set interfaces se-3/0/7 encapsulation ppp
set interfaces se-3/0/7 serial-options clocking-mode dce
set interfaces se-3/0/7 unit 0 family inet address 17.17.17.2/24
set routing-options static route 20.20.20.0/24 next-hop 10.10.10.1
set routing-options static route 22.22.22.0/24 next-hop 11.11.11.1
set routing-options static route 24.24.24.0/24 next-hop 12.12.12.1
set routing-options static route 26.26.26.0/24 next-hop 13.13.13.1
set routing-options static route 28.28.28.0/24 next-hop 14.14.14.1
set routing-options static route 30.30.30.0/24 next-hop 15.15.15.1
set routing-options static route 32.32.32.0/24 next-hop 16.16.16.1
set routing-options static route 34.34.34.0/24 next-hop 17.17.17.1
```

To configure the interfaces on Device 1:

1. Specify the maximum transmission unit (MTU) value for the interface.

   ```
   [edit interfaces]
   user@host# set se-7/0/0 mtu 9192
   ```

2. Set the encapsulation type.

   ```
   [edit interfaces]
   user@host# set se-7/0/0 encapsulation ppp
   ```

3. Set the serial options, such as the clocking mode.

   ```
   [edit interfaces]
   user@host# set se-7/0/0 serial-options clocking-mode internal
   ```

4. Set the IPv4 address on the serial port.

   ```
   [edit interfaces]
   user@host# set se-7/0/0 unit 0 family inet address 10.10.10.1/24
   ```

5. Specify the static route information.

   ```
   [edit routing-options]
   user@host# set static route 21.21.21.0/24 next-hop 10.10.10.2
   ```

   Repeat the same configuration for the other seven ports on Device 1.

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

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

To configure the interfaces on Device 2:

1. Specify the MTU value for the interface.

```
[edit interfaces]
user@host# set se-3/0/0 mtu 9192
```

2. Set the encapsulation type.

```
[edit interfaces]
user@host# set se-3/0/0 encapsulation ppp
```

3. Set the serial options, such as the clocking mode.

```
[edit interfaces]
user@host# set se-3/0/0 serial-options clocking-mode dce
```

4. Set the IPv4 address on the serial port.

```
[edit interfaces]
user@host# set se-3/0/0 unit 0 family inet address 10.10.10.2/24
```

5. Specify the static route information.

```
[edit routing-options]
user@host# set static route 20.20.20.0/24 next-hop 10.10.10.1
```

   Repeat the same configuration for the other seven ports on Device 2.

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

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

Verification

IN THIS SECTION

- Purpose | 299
- Action | 299

Purpose

Display information about the parameters configured on the serial interfaces.

Action

- You can use the ping tool on each peer address in the network to verify that all interfaces on the device are operational. To verify the link state of all interfaces:

  For each interface on the device:

  1. In the J-Web interface, select `Troubleshoot > Ping Host`.
  2. In the Remote Host box, type the address of the interface for which you want to verify the link state.
  3. Click `Start`. The output appears on a separate page.

  ```
  PING 10.10.10.10 : 56 data bytes
  64 bytes from 10.10.10.10: icmp_seq=0 ttl=255 time=0.382 ms
  64 bytes from 10.10.10.10: icmp_seq=1 ttl=255 time=0.266 ms
  ```

  If the interface is operational, it generates an ICMP response. If this response is received, the round-trip time, in milliseconds, is listed in the time field.

- To verify that the interface properties are correct, use the `show interfaces detail` command to display a summary of interface information. Verify the following information:

  - The physical interface is Enabled. If the interface is shown as Disabled, do one of the following:

    - In the CLI configuration editor, delete the `disable` statement at the [edit interfaces se-1/0/0] level of the configuration hierarchy.

        - In the J-Web configuration editor, clear the `Disable` check box on the Interfaces > se-1/0/0 page.

    - The physical link is Up. A link state of Down indicates a problem with the interface module, interface port, or physical connection (link-layer errors).

    - The Last Flapped time is an expected value. It indicates the last time the physical interface became unavailable and then available again. Unexpected flapping indicates likely link-layer errors.

    - The traffic statistics reflect expected input and output rates. Verify that the number of inbound and outbound bytes and packets matches expected throughput for the physical interface. To clear the statistics and see only new changes, use the `clear interfaces statistics se-1/0/0` command.

- To verify and that the interface link status is up, use the enter the `show interface terse se-7/0/*` command:

```
user@srx650-1> show interface terse se-7/0/*
```

```


Interface               Admin Link Proto    Local                 Remote
se-7/0/0                up    up
se-7/0/0.0              up    up   inet     10.10.10.1/24
se-7/0/1                up    up
se-7/0/1.0              up    up   inet     11.11.11.1/24
se-7/0/2                up    up
se-7/0/2.0              up    up   inet     12.12.12.1/24
se-7/0/3                up    up
se-7/0/3.0              up    up   inet     13.13.13.1/24
se-7/0/4                up    up
se-7/0/4.0              up    up   inet     14.14.14.1/24
se-7/0/5                up    up
se-7/0/5.0              up    up   inet     15.15.15.1/24
se-7/0/6                up    up
se-7/0/6.0              up    up   inet     16.16.16.1/24
se-7/0/7                up    up
se-7/0/7.0              up    up   inet     17.17.17.1/24
```

The output displays a list of all interfaces configured. If the Link column displays `up` for all interfaces, the configuration is correct. This verifies that the GPIM is up and end-to-end ping is working.

- To verify the interface statistics for DCE, use the `show interface se-7/0/0 extensive | no-more` command:

```
user@srx650-1>show interface se-7/0/0 extensive | no-more
```

```
Physical interface: se-7/0/0, Enabled, Physical link is Up
  Interface index: 161, SNMP ifIndex: 592, Generation: 164
  Type: Serial, Link-level type: PPP, MTU: 1504, Maximum speed: 8mbps
  Device flags   : Present Running
  Interface flags: Point-To-Point Internal: 0x0
  Link flags     : Keepalives
  Hold-times     : Up 0 ms, Down 0 ms
  Keepalive settings: Interval 10 seconds, Up-count 1, Down-count 3
  Keepalive statistics:
    Input : 123 (last seen 00:00:02 ago)
    Output: 123 (last sent 00:00:01 ago)
  LCP state: Opened
  NCP state: inet: Opened, inet6: Not-configured, iso: Not-configured, mpls:
  Not-configured
  CHAP state: Closed
  PAP state: Closed
  CoS queues     : 8 supported, 8 maximum usable queues
  Last flapped   : 2011-06-27 22:57:24 PDT (00:20:59 ago)
  Statistics last cleared: Never
  Traffic statistics:
   Input  bytes  :                23792                  160 bps
   Output bytes  :                22992                  536 bps
   Input  packets:                  404                    0 pps
   Output packets:                  409                    0 pps
  Input errors:
    Errors: 3, Drops: 0, Framing errors: 3, Runts: 0, Giants: 0,
    Policed discards: 0, Resource errors: 0
  Output errors:
    Carrier transitions: 1, Errors: 0, Drops: 0, MTU errors: 0,
    Resource errors: 0
  Egress queues: 8 supported, 4 in use
  Queue counters:       Queued packets  Transmitted packets      Dropped packets
    0 best-effort                    0                    0                    0
    1 expedited-fo                   0                    0                    0
    2 assured-forw                   0                    0                    0
    3 network-cont                 409                  409                    0
  Queue number:         Mapped forwarding classes
```

```
    0                   best-effort
    1                   expedited-forwarding
    2                   assured-forwarding
         3                   network-control
  Serial media information:
    Line protocol: eia530
    Resync history:
      Sync loss count: 0
    Data signal:
      Rx Clock: OK
    Control signals:
             Local mode: DCE
      To DTE: CTS: up, DCD: up, DSR: up
      From DTE: DTR: up, RTS: up
    DCE loopback override: Off
    Clocking mode: internal
    Loopback: none
    Tx clock: non-invert
    Line encoding: nrz
  Packet Forwarding Engine configuration:
    Destination slot: 7
  CoS information:
    Direction : Output
    CoS transmit queue               Bandwidth               Buffer Priority   Limit
                              %            bps     %           usec
    0 best-effort            95        7600000    95               0      low    none
    3 network-control         5         400000     5               0      low    none

  continue................................................................................
..........................................................................................
```

The output displays a list of all DCE verification parameters and the mode configured. If the local mode displays DCE, the configuration is correct.

- To verify the interface statistics for DTE, use the `show interface se-3/0/0 extensive | no-more` command:

```
user@srx650-2>show interfaces se-3/0/0 extensive | no-more
```

```
Physical interface: se-3/0/0, Enabled, Physical link is Up
  Interface index: 168, SNMP ifIndex: 594, Generation: 171
  Type: Serial, Link-level type: PPP, MTU: 1504, Maximum speed: 8mbps
  Device flags   : Present Running
  Interface flags: Point-To-Point Internal: 0x0
  Link flags     : Keepalives
  Hold-times     : Up 0 ms, Down 0 ms
  Keepalive settings: Interval 10 seconds, Up-count 1, Down-count 3
  Keepalive statistics:
    Input : 242 (last seen 00:00:09 ago)
    Output: 242 (last sent 00:00:10 ago)
  LCP state: Opened
  NCP state: inet: Opened, inet6: Not-configured, iso: Not-configured, mpls:
  Not-configured
  CHAP state: Closed
  PAP state: Closed
  CoS queues     : 8 supported, 8 maximum usable queues
  Last flapped   : 2011-06-27 22:52:06 PDT (00:40:41 ago)
  Statistics last cleared: Never
  Traffic statistics:
   Input  bytes  :                44582                    0 bps
   Output bytes  :                42872                    0 bps
   Input  packets:                  776                    0 pps
   Output packets:                  779                    0 pps
Input errors:
    Errors: 6, Drops: 0, Framing errors: 6, Runts: 0, Giants: 0,
    Policed discards: 0, Resource errors: 0
  Output errors:
    Carrier transitions: 1, Errors: 0, Drops: 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                    2                    2                    0
    1 expedited-fo                   0                    0                    0
    2 assured-forw                   0                    0                    0
    3 network-cont                 777                  777                    0
  Queue number:         Mapped forwarding classes
    0                   best-effort
    1                   expedited-forwarding
    2                   assured-forwarding
    3                   network-control
  Serial media information:
    Line protocol: eia530
    Resync history:
      Sync loss count: 0
    Data signal:
      Rx Clock: OK
    Control signals:
      Local mode: DTE
      To DCE: DTR: up, RTS: up
      From DCE: CTS: up, DCD: up, DSR: up
    Clocking mode: loop-timed
    Loopback: none
    Tx clock: non-invert
    Line encoding: nrz
  Packet Forwarding Engine configuration:
    Destination slot: 3
  CoS information:
    Direction : Output
    CoS transmit queue               Bandwidth               Buffer Priority   Limit
                              %            bps     %           usec
    0 best-effort            95        7600000    95              0      low    none
    3 network-control         5         400000     5              0      low    none
continue ...........................................................................
.....................................................................................
```

The output displays a list of all DTE verification parameters and the mode configured. If the local mode displays DTE, the configuration is correct.

RELATED DOCUMENTATION

Using the CLI Editor in Configuration Mode

3

CHAPTER

Monitor and Troubleshooting Interfaces

IN THIS CHAPTER

- Monitor Interfaces | 306
- Troubleshooting Interfaces | 310

Monitor Interfaces

IN THIS SECTION

- Trace Interface Operations Overview | 306
- Trace Operations of an Individual Router Interface | 306
- Trace Operations of the Interface Process | 307

This topic discusses about tracing operations of individual router interface, interface process, and pppd process.

Trace Interface Operations Overview

You can trace the operations of individual router interfaces and those of the interface process (dcd). For a general discussion of tracing and of the precedence of multiple tracing operations, see the Junos OS Administration Library for Routing Devices.

For information about the operations of Virtual Router Resolution Protocol (VRRP)-enabled interfaces, see the Junos OS High Availability User Guide.

SEE ALSO

Trace Operations of an Individual Router Interface | 306

Tracing Operations of the Interface Process

Trace Operations of an Individual Router Interface

To trace the operations of individual router interfaces, perform the following steps:

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

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

2. Configure the traceoptions option.

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

3. Configure the tracing flag.

```
[edit interfaces interface-name traceoptions]
user@host# set flag flag-option
```

You can specify the following interface tracing flags:

- all—Trace all interface operations.
- event—Trace all interface events.
- ipc—Trace all interface interprocess communication (IPC) messages.
- media—Trace all interface media changes.

The interfaces traceoptions statement does not support a trace file. The logging is done by the kernel, so the tracing information is placed in the system syslog files.

For more information about trace operations, see *Tracing Operations of the Interface Process*.

SEE ALSO

traceoptions

Trace Operations of the Interface Process

SUMMARY

Learn about how to trace interface process operations using traceoptions.

To trace the operations of the router or switch interface process, dcd, perform the following steps:

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

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

2. Configure the `traceoptions` statement.

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

3. Configure the `no-remote-trace` option to disable remote tracing.

```
[edit interfaces traceoptions]
user@host# set no-remote-trace
```

4. Configure the `file` *`filename`* option.

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

5. Configure the `files` *`number`* option, `match` *`regular-expression`* option, `size` *`size`* option, and `world-readable | no-world-readable` option.

```
[edit interfaces traceoptions file]
user@host# set files number
user@host# set match regular-expression
user@host# set size size
user@host# set word-readable | no-world-readable
```

6. Configure the tracing flag.

```
[edit interfaces traceoptions]
user@host# set flag flag-option
```

7. Configure the `disable` option in `flag` *`flag-option`* statement to disable the tracing operation. You can use this option to disable a single operation when you have defined a broad group of tracing operations, such as `all`.

```
[edit interfaces traceoptions]
user@host# set flag flag-option disable
```

You can specify the following flags in the `interfaces traceoptions` statement:

- `all`—Enable all configuration logging.
- `change-events`—Log changes that produce configuration events.
- `gres-events`—Log the events related to GRES.
- `resource-usage`—Log the resource usage for different states.
- `config-states`—Log the configuration state machine changes.
- `kernel`—Log configuration IPC messages to kernel.
- `kernel-detail`—Log details of configuration messages to kernel.
- `select-events`—Log the events on select state machine.

By default, interface process operations are placed in the file named dcd and three 1-MB files of tracing information are maintained.

For general information about tracing, see the tracing and logging information in the Junos OS Administration Library for Routing Devices.

SEE ALSO

Tracing Interface Operations Overview

Tracing Operations of an Individual Router Interface

traceoptions

Troubleshooting Interfaces

IN THIS SECTION

- Troubleshooting: em0 Management Interface Link is Down | 310
- Troubleshooting: fxp0 Management Interface Link is Down | 313
- Troubleshooting: Faulty Ethernet Physical Interface on an M Series, an MX Series, or a T Series Router | 315
- Time Domain Reflectometry on ACX Series Routers Overview | 327
- Diagnose a Faulty Twisted-Pair Cable on ACX Series Routers | 330

This topic discusses various troubleshooting scenarios.

Troubleshooting: em0 Management Interface Link is Down

IN THIS SECTION

- Problem | 310
- Diagnosis | 311
- Resolution | 312

Problem

Description

`Ethernet Link Down` alarm is raised when you run the `show chassis alarm` operational mode command on a T640 router, a T1600 router, T4000 router, or a TX Matrix Plus router.

Diagnosis

Perform the following tests to check if the em0 management interface is down on the 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
   2011-10-19 11:13:02 MYT Major Host 1 em0 : Ethernet Link Down
   ```

   Is the alarm `Ethernet Link Down` displayed against the em0 interface of the primary Routing Engine (Host 0)?

   - Yes: Contact JTAC for further assistance.
   - No: Continue to the next diagnostic test.

1. Run the `show interfaces em0` and the `show interfaces em0 terse` operational mode commands.

   show interfaces em0

   ```
   user@host> show interfaces em0
   Physical interface: em0, Enabled, Physical link is Up
   Interface index: 1, SNMP ifIndex: 1
   Type: Ethernet, Link-level type: Ethernet, MTU: 1514, Speed: 100mbps
   Device flags : Present Running
   Interface flags: SNMP-Traps
   ...
   ```

   show interfaces em0 terse

   ```
   user@host> show interfaces em0 terse
   Interface Admin Link Proto Local Remote
   em0 up up
   em0.0 up up inet 10.100.100.1/30
   ```

   Is the em0 interface on the primary Routing Engine `up`?

   - Yes: Continue to resolution.

- No: Contact JTAC for further assistance

Resolution

To Resolve This Issue

From the aforementioned diagnosis, we ascertain that the chassis alarm has been raised for the em0 management interface in the backup Routing Engine (Host 1) and not for the primary Routing Engine (Host 0).

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

- Disable the em0 interface in the backup Routing Engine:

  1. In configuration mode, go to the `[edit groups re1]` hierarchy level.

     ```
     user@host1# edit groups re1
     ```

  2. Disable the em0 interface.

     ```
     [edit groups re1]
     user@host1# set interfaces em0 disable
     ```

- Ignore the alarm:

  1. In configuration mode, go to the `[edit chassis]` hierarchy level.

     ```
     user@host1# edit chassis
     ```

  2. Ignore the `Ethernet link down` alarm on the management interface by setting the `management-ethernet link-down` alarm option to `ignore`.

     ```
     [edit chassis]
     user@host1# set alarm management-ethernet link-down ignore
     ```

SEE ALSO

Supported Routing Engines by Router

show chassis alarms

Troubleshooting: fxp0 Management Interface Link is Down

IN THIS SECTION

- Problem | 313
- Diagnosis | 313
- Resolution | 314

Problem

Description

`Ethernet Link Down` alarm is raised when you run the `show chassis alarm` operational mode command on an M Series router, an MX Series router, a T320 router, a T640 router, a T1600 router, or on a TX Matrix router.

Diagnosis

Perform the following tests to check if the fxp0 interface is down on the 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
   2011-10-19 11:13:02 MYT Major Host 1 fxp0 : Ethernet Link Down
   ```

   Is the alarm `Ethernet Link Down` displayed against the fxp0 interface of the primary Routing Engine (Host 0)?

   - Yes: Contact JTAC for further assistance.

- No: Continue to the next diagnostic test.

1. Run the `show interfaces fxp0` and the `show interfaces fxp0 terse` operational mode commands.

   show interfaces fxp0

   ```
   user@host> show interfaces fxp0
   Physical interface: fxp0, Enabled, Physical link is Up
   Interface index: 1, SNMP ifIndex: 1
   Type: Ethernet, Link-level type: Ethernet, MTU: 1514, Speed: 100mbps
   Device flags : Present Running
   Interface flags: SNMP-Traps
   ...
   ```

   show interfaces fxp0 terse

   ```
   user@host> show interfaces fxp0 terse
   Interface Admin Link Proto Local Remote
   fxp0 up up
   fxp0.0 up up inet 10.100.100.1/30
   ```

   Is the fxp0 interface on the primary Routing Engine `up`?

   - Yes: Continue to resolution.

   - No: Contact JTAC for further assistance

Resolution

To Resolve This Issue

From the diagnosis, we ascertain that the chassis alarm has been raised for the fxp0 management interface in the backup Routing Engine (Host 1) and not for the primary Routing Engine (Host 0).

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

- Disable the fxp0 interface in the backup Routing Engine:

  1. In configuration mode, go to the `[edit groups re1]` hierarchy level.

     ```
     user@host1# edit groups re1
     ```

2. Disable the fxp0 interface.

```
[edit groups re1]
user@host1# set interfaces fxp0 disable
```

- Ignore the alarm:

1. In configuration mode, go to the `[edit chassis]` hierarchy level.

```
user@host1# edit chassis
```

2. Ignore the `Ethernet link down` alarm on the management interface by setting the `management-ethernet link-down` alarm option to `ignore`.

```
[edit chassis]
user@host1# set alarm management-ethernet link-down ignore
```

SEE ALSO

Supported Routing Engines by Router

show chassis alarms

Troubleshooting: Faulty Ethernet Physical Interface on an M Series, an MX Series, or a T Series Router

IN THIS SECTION

- Check the Cable Connection | **316**
- Check the Physical Link Status of the Interface | **317**
- Check the Interface Statistics in Detail | **319**
- Perform the Loopback Diagnostic Test | **322**
- Check for Other Possibilities | **324**

- Enable a Physical Interface | 326

You can follow the basic troubleshooting checklist as explained in the following topics from one through five to troubleshoot an Ethernet physical interface on an M Series, MX Series, or a T Series router.

Check the Cable Connection

IN THIS SECTION

- Problem | 316
- Diagnosis | 316
- Resolution | 316

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 317.
   - 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. Swap the Ethernet cable for a known good cable if the existing cable is damaged.

3. Connect a single-mode fiber cable to a single-mode interface only and a multimode fiber cable to a multimode interface only. To check fiber optic cable integrity, see 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 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 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 | 317
- Solution | 318
- Diagnosis | 318

Problem

Description

Unable to transmit and receive packets on the Ethernet interface even though the cable connection is correct.

Solution

To display the physical link status of the interface, run the `show interface` *`interface-name`* `media` operational mode command. For example, on the ge-5/0/1 interface.

```
user@host> show interfaces ge-5/0/1 media
Physical interface: ge-5/0/1, Enabled, Physical link is Up
  Interface index: 317, SNMP ifIndex: 1602
  Link-level type: Ethernet, MTU: 1514, Speed: 1000mbps, BPDU Error: None, MAC-REWRITE Error:
None, Loopback: Disabled,
  Source filtering: Disabled, Flow control: Enabled, Auto-negotiation: Enabled, Remote fault:
Online, Speed-negotiation: Disabled,
  Auto-MDIX: Enabled
  Device flags   : Present Running
  Interface flags: SNMP-Traps Internal: 0x4000
  Link flags     : None
  CoS queues     : 8 supported, 8 maximum usable queues
  Current address: 2c:6b:f5:4c:26:73, Hardware address: 2c:6b:f5:4c:26:73
  Last flapped   : 2012-11-30 01:25:37 UTC (03:46:55 ago)
  Input rate     : 880 bps (1 pps)
  Output rate    : 312 bps (0 pps)
  Active alarms  : None
  Active defects : None
  MAC statistics:
    Input bytes: 901296, Input packets: 9799, Output bytes: 976587, Output packets: 10451
  Filter statistics:
    Filtered packets: 68, Padded packets: 0, Output packet errors: 0
  Autonegotiation information:
    Negotiation status: Complete
    Link partner:
        Link mode: Full-duplex, Flow control: Symmetric/Asymmetric, Remote fault: OK
    Local resolution:
        Flow control: Symmetric, Remote fault: Link OK
  Interface transmit statistics: Disabled
```

For information about `show interfaces` *`interface-name`* `media`, see *show interfaces* .

Diagnosis

1. Are there any connectivity problems such as input errors and packet loss even though the `Enabled` field displays `Physical link is Up` status and the `Active alarms` and `Active defect` field displays `None`?

- Yes: Go to "Check the Interface Statistics in Detail" on page 319.

- 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 | 319
- Solution | 319
- Diagnosis | 322

Problem

Description

The physical interface is not working even though the `Enabled` field displays `Physical link is Up` status and the `Active alarms and Active defect` field displays `None`.

Solution

To display the interface statistics in detail, run the `show interface` *`interface-name`* `extensive` operational command. For example, on ge-5/0/1 interface.

```
user@host> show interfaces ge-5/0/1 extensive
Physical interface: ge-5/0/1, Enabled, Physical link is Up
  Interface index: 317, SNMP ifIndex: 1602, Generation: 322
  Link-level type: Ethernet, MTU: 1514, Speed: 1000mbps, BPDU Error: None, MAC-REWRITE Error:
None, Loopback: Disabled,
  Source filtering: Disabled, Flow control: Enabled, Auto-negotiation: Enabled, Remote fault:
Online, Speed-negotiation: Disabled,
  Auto-MDIX: Enabled
  Device flags   : Present Running
  Interface flags: SNMP-Traps Internal: 0x4000
  Link flags     : None
  CoS queues     : 8 supported, 8 maximum usable queues
  Hold-times     : Up 0 ms, Down 0 ms
  Current address: 2c:6b:f5:4c:26:73, Hardware address: 2c:6b:f5:4c:26:73
  Last flapped   : 2012-11-30 01:25:37 UTC (04:38:32 ago)
  Statistics last cleared: Never
  Traffic statistics:
   Input  bytes  :               806283                    0 bps
   Output bytes  :              1153215                  424 bps
   Input  packets:                10818                    0 pps
   Output packets:                11536                    0 pps
   IPv6 transit statistics:
    Input  bytes  :                   0
    Output bytes  :                   0
    Input  packets:                   0
    Output packets:                   0
  Label-switched interface (LSI) traffic statistics:
   Input  bytes  :                    0                    0 bps
   Input  packets:                    0                    0 pps
  Dropped traffic statistics due to STP State:
   Input  bytes  :                    0
   Output bytes  :                    0
   Input  packets:                    0
   Output packets:                    0
  Input errors:
    Errors: 0, Drops: 0, Framing errors: 0, Runts: 0, Policed discards: 233060, L3 incompletes:
0, L2 channel errors: 0,
    L2 mismatch timeouts: 0, FIFO errors: 0, Resource errors: 0
  Output errors:
    Carrier transitions: 11, Errors: 0, Drops: 0, Collisions: 0, Aged packets: 0, FIFO errors:
0, HS link CRC errors: 0,
    MTU errors: 0, Resource errors: 0
  Egress queues: 8 supported, 4 in use
  Queue counters:       Queued packets  Transmitted packets      Dropped packets
    0 best-effort                 3216                 3216                    0
    1 expedited-fo                   0                    0                    0
    2 assured-forw                   0                    0                    0
    3 network-cont                8320                 8320                    0
  Queue number:         Mapped forwarding classes
    0                   best-effort
    1                   expedited-forwarding
    2                   assured-forwarding
    3                   network-control
  Active alarms  : None
  Active defects : None
  MAC statistics:                      Receive         Transmit
    Total octets                       1007655          1082219
    Total packets                        10886            11536
    Unicast packets                       4350             4184
    Broadcast packets                       32               77
    Multicast packets                     6504             7275
    CRC/Align errors                         0                0
    FIFO errors                              0                0
    MAC control frames                       0                0
    MAC pause frames                         0                0
    Oversized frames                         0
    Jabber frames                            0
    Fragment frames                          0
    VLAN tagged frames                       0
    Code violations                          0
  Filter statistics:
    Input packet count                   10886
    Input packet rejects                    68
    Input DA rejects                        68
    Input SA rejects                         0
    Output packet count                                   11536
    Output packet pad count                                   0
    Output packet error count                                 0
    CAM destination filters: 0, CAM source filters: 0
  Autonegotiation information:
    Negotiation status: Complete
    Link partner:
        Link mode: Full-duplex, Flow control: Symmetric/Asymmetric, Remote fault: OK
    Local resolution:
        Flow control: Symmetric, Remote fault: Link OK
  Packet Forwarding Engine configuration:
    Destination slot: 5
  CoS information:
    Direction : Output
    CoS transmit queue               Bandwidth               Buffer Priority   Limit
                              %            bps     %           usec
    0 best-effort            95      950000000    95              0      low    none
```

```
    3 network-control         5       50000000     5              0       low     none
  Interface transmit statistics: Disabled
```

For information about `show interfaces` *`interface-name`* `detail`, see *show interfaces* .

Diagnosis

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

   For information about the errors, see*show interfaces* .

   - Yes: Resolve the errors as needed. Resolving these errors is beyond the scope of this topic.
   - No: Continue with "Perform the Loopback Diagnostic Test" on page 322.

Perform the Loopback Diagnostic Test

IN THIS SECTION

- Problem | 322
- Solution | 322
- Diagnosis | 324

Problem

Description

The interface cable is connected correctly and there are no alarms or errors associated with the Ethernet physical interface; yet the interface is not working.

Solution

To check whether the Ethernet port or PIC is faulty, you must perform the internal loopback test and hardware loopback test.

To perform an internal loopback diagnostic test on an Ethernet interface, for example on ge-5/0/1 interface:

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

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

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

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

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

```
user@host> show interfaces ge-5/0/1 media
Physical interface: ge-5/0/1, Enabled, Physical link is Up
  Interface index: 317, SNMP ifIndex: 1602
  Link-level type: Ethernet, MTU: 1514, Speed: 1000mbps, BPDU Error: None, MAC-REWRITE Error: 
None, Loopback: Enabled,
  Source filtering: Disabled, Flow control: Enabled, Auto-negotiation: Enabled, Remote fault: 
Online, Speed-negotiation: Disabled,
  Auto-MDIX: Enabled
  Device flags   : Present Running
  Interface flags: SNMP-Traps Internal: 0x4000
  Link flags     : None
  CoS queues     : 8 supported, 8 maximum usable queues
  Current address: 2c:6b:f5:4c:26:73, Hardware address: 2c:6b:f5:4c:26:73
  Last flapped   : 2012-11-30 01:25:37 UTC (03:46:55 ago)
  Input rate     : 880 bps (1 pps)
  Output rate    : 312 bps (0 pps)
  Active alarms  : None
  Active defects : None
  MAC statistics:
    Input bytes: 901296, Input packets: 9799, Output bytes: 976587, Output packets: 10451
  Filter statistics:
    Filtered packets: 68, Padded packets: 0, Output packet errors: 0
  Autonegotiation information:
    Negotiation status: Complete
    Link partner:
        Link mode: Full-duplex, Flow control: Symmetric/Asymmetric, Remote fault: OK
    Local resolution:
        Flow control: Symmetric, Remote fault: Link OK
  Interface transmit statistics: Disabled
```

NOTE: Delete the `loopback` statement after completing your diagnosis.

Execute one of the following steps for a hardware loopback diagnostic test as needed:

- For an Ethernet PIC with a fiber optic interface—Physically loop the T_X and R_X port and check the status of the physical link with the `show interfaces` *`interface-name`* `media` operational mode command.

- For an Ethernet PIC with an RJ-45 Ethernet interface—Build a loopback plug by crossing pin 1 (T_X +) to pin 3 (R_X +) together and pin 2 (T_X -) and pin 6 (R_X -) together and check the status of the physical link with the `show interfaces` *`interface-name`* `media` operational mode command.

NOTE: For information about loopback testing, see *Performing Loopback Testing for Fast Ethernet and Gigabit Ethernet Interfaces*.

Diagnosis

1. Does the `Enabled` field display `Physical link is Up` status and the `Active alarms and Active defect` field display `None` when you perform the loopback test?

   - Yes: Go to the "Check for Other Possibilities" on page 324 section.

   - No: Continue to the next diagnostic test.

1. 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 324 section.

   - No: Contact JTAC for further assistance.

Check for Other Possibilities

IN THIS SECTION

Problem | 325

Solution | 325
Diagnosis | 325

Problem

Description

Loopback diagnostic test is successful but unable to transmit and receive packets on the Ethernet interface.

Solution

Use the following commands as needed to troubleshoot an Ethernet interface, for example, a ge-5/0/1 interface:

- Run the `show interfaces` *`interface-name`* `terse` operational command to check if the physical interface and logical interfaces are administratively disabled. For example, on ge-5/0/1 interface.

```
user@host> show interfaces ge-5/0/1 terse
     Interface               Admin Link Proto    Local                 Remote
     ge-5/0/1                up    up
     ge-5/0/1.0              up    up   inet     20.1.1.2/24
```

Diagnosis

1. Does the physical interface and its corresponding logical interfaces display `down` in the output of the `show interfaces` *`interface-name`* `terse` operational mode command?

   - Yes: Enable the interfaces as shown in "Enable a Physical Interface" on page 326.

   - 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 correctly set for the interface?

   NOTE: Check if the associated Flexible PIC Concentrator (FPC), Modular Port Concentrator (MPC), or Dense Port Concentrator (DPC) and its Modular Interface Card

(MIC) or PIC with its 10-gigabit small form-factor pluggable transceiver (XFP) or SFP supports speed and auto-negotiation settings.

- Yes: Check *Monitoring Fast Ethernet and Gigabit Ethernet Interfaces* for more troubleshooting tips.
- No: Contact JTAC for further assistance.

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 a ge-5/0/1 interface:

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

```
disable;
```

3. Enable the interface and commit.

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

SEE ALSO

show interfaces

Time Domain Reflectometry on ACX Series Routers Overview

Time Domain Reflectometry (TDR) is a technology used for diagnosing copper cable states. This technique is used to determine if cabling is at fault when you cannot establish a link. TDR detects the defects by sending a signal through a cable, and reflecting it from the end of the cable. Open circuits, short circuits, sharp bends, and other defects in the cable, reflects the signal back, at different amplitudes, depending on the severity of the defect.

Several factors that result in degraded or low-quality cable plants can cause packet loss, suboptimal connection speed, reduced network efficiency, and complete connection failures. These types of problems can occur because of poor cable construction, identification of pair twists, loose connectors, poor contacts between the points, and stretched or broken pairs of cables. Broadcom transceivers enable you to analyze the condition of the cable plant or topology and identify any problems that have occurred. This functionality is effectively used in the following scenarios:

- Troubleshooting during initial network equipment installation.
- Discovery of failures when network problems occur.
- Maintenance of optimally functioning cable plants.
- Fault determination during the testing of network equipment in production cable networks.

TDR supports the following capabilities for examination of cable faults on ACX Series routers:

- Cable status pair (open or short)—When the router operates in Gigabit Ethernet mode, all four pairs (8 wires) are used. Only Pair-A and Pair-B are required to operate in 10/100BASE-T Ethernet mode. If either of these required pairs is open or short-circuited, the transceiver reports the following faults:
    - Any open wire
    - Wires of a particular pair that are shorted
- Distance to fault per pair—Distance at which an open or a short-circuit is detected in meters. This measurement is also termed as cable length. The transceiver reports the following faults:
    - Cable length when the cable status is normal
    - Distance to fault when the cable status is not normal
- Pair Swap—Swapping of twisted-pairs in straight-through and cross-over cable plants are detected.
- Polarity Swap—Each cable pair carries a differential signal from one end to the other end of the cable. Each wire within the pair is assigned a polarity. The wires in a pair are normally connected in a one-to-one form. This connection enables the transmitter at one end to be connected to the receiver at the other end with same polarity. Sometimes, the wiring within the pair is also swapped. This type of connection is called polarity swap. Broadcom transceivers can detect such swapping and

automatically adjust the connection to enable the links to operate normally. However, the transceiver reports polarity swaps that it detects in the cable plant.

On 4-port Gigabit Ethernet and 8-port Gigabit Ethernet MICs with copper SFP transceivers (using BCM54880) and 4-port Gigabit Ethernet, 6-port Gigabit Ethernet, and 8-port Gigabit Ethernet MICs with copper and optical SFP transceivers (using BCM54640E PHY), only 10BASE-T pair polarity is supported. 100BASE-T and 1000BASE-T polarities are not supported.

When the Gigabit Ethernet link cannot be established (for example, if only two pairs are present that are fully functional), TDR in the physical layer (PHY) brings down the link to a 100 MB link, which is called a downshift in the link. The physical layer might require 10-20 seconds for the link to come up if a downgrade in wire speed occurs because it attempts to connect at 1000 MB five times before it falls back to 100BASE-TX.

TDR diagnostics is supported only on copper interfaces and not on fiber interfaces.

Keep the following points in mind when you configure TDR:

- If you connect a port undergoing a TDR test to a Gigabit Ethernet interface that is enabled to automatically detect MDI (Media Dependent Interface) and MDIX (Media Dependent Interface with Crossover) port connections, the TDR result might be invalid.
- If you connect a port undergoing a TDR test to a 100BASE-T copper interface, the unused pairs are reported as faulty because the remote end does not terminate these pairs.
- You must not modify the port configuration while the TDR test is running.
- Because of cable characteristics, you need to run the TDR test multiple times to get accurate results.
- Do not change the port status (such as removing the cable at the near or far end) because such a change can result in inaccurate statistics in the results.
- While measuring the cable length or distance to fault (per pair), sometimes, a few cable length inconsistencies might be observed during a TDR test. Broadcom transceivers have the following cable length limitations:
  - For a properly-terminated good cable, the accuracy of the cable length reported is plus or minus 10 meters.
  - If a pair is open or short-circuited, the far-end termination does not affect the computed result for that pair.
  - The accuracy of the measured cable length, when open and short-circuit conditions are detected, is plus or minus 5 meters.
  - The accuracy of a good pair, when one or more pairs are open or short-circuited, is plus or minus 10 meters.

- Polarity swap detection is supported only in 10BASE-T mode.
- The TDR test does not impact the traffic if the interface operates at 10-Gigabit Ethernet per second of bandwidth, which is the default configuration. However, if the speed of the interface is configured to be other than 10-Gigabit Ethernet, running the TDR test affects the traffic.

  TDR diagnostics might bring the link down and initialize the physical layer (PHY) with default configuration to perform its operation.

  When the TDR validation test is completed, the PHY layer resumes operation in the same manner as before the cable diagnostics test was performed. However, link flaps might be momentarily observed. We recommend that you run the TDR test at a speed of 1 gigabit per second, which is the default configuration, to obtain more accurate results.

TDR is supported on the following interfaces on ACX Series routers:

- On ACX1000 routers, 4 RJ45 (Cu) ports or 8-port Gigabit Ethernet MICs with small form-factor pluggable (SFP) transceivers and RJ45 connectors.

  On ACX1100 routers, 4-port or 8-port Gigabit Ethernet MICs with SFP transceivers and RJ45 connectors.
- On ACX2000 routers, 8-port Gigabit Ethernet MICs with SFP transceivers and RJ45 connectors.
- On ACX2100 and ACX2200 routers, 4-port Gigabit Ethernet MICs with SFP transceivers and RJ45 connectors.
- On ACX4000 routers, 4-port, 6-port, or 8-port Gigabit Ethernet MICs with SFP transceivers and RJ45 connectors.

You must select the media type as copper for the 1-Gigabit Ethernet interfaces. To specify the media type, include the `media-type` statement with the `copper` option at the `[edit interfaces *interface-name*]` hierarchy level. Media type selection is applicable to ports only in slot 2. When media-type is not set, the port accepts either type of connection. The media type is fiber if a transceiver is installed in the SFP connection. If no transceiver is installed, the media type is copper. The COMBO ports (combination ports) on ACX routers support both the copper and fiber-optic media types. On such ports or interfaces, you must configure the media type as copper to run the TDR test.

You can run the TDR test from operational mode and view the success or failure results of the test. To start a test on a specific interface, issue the `request diagnostics tdr start interface` *`interface-name`* command. To stop the TDR test currently in progress on the specified interface, issue the `request diagnostics tdr abort interface` *`interface-name`* command. To display the test results for all copper interfaces, enter the `show diagnostics tdr` command. To display the test results for a particular interface, enter the `show diagnostics tdr interface interface-name` command.

SEE ALSO

Diagnose a Faulty Twisted-Pair Cable on ACX Series Routers | 330

Diagnose a Faulty Twisted-Pair Cable on ACX Series Routers

IN THIS SECTION

- Problem | 330
- Solution | 330

Problem

Description

A 10/100BASE-T Ethernet interface has connectivity problems that you suspect might be caused by a faulty cable.

Solution

Use the time domain reflectometry (TDR) test to determine whether a twisted-pair Ethernet cable is faulty.

The TDR test:

- Detects and reports faults for each twisted pair in an Ethernet cable. Faults detected include open circuits, short circuits, and impedance mismatches.
- Reports the distance to fault to within 1 meter.
- Detects and reports pair swaps, pair polarity reversals, and excessive pair skew.

The TDR test is supported on the following ACX routers and interfaces:

- On ACX1000 routers, 4 RJ45 (Cu) ports or 8-port Gigabit Ethernet MICs with small form-factor pluggable (SFP) transceivers and RJ45 connectors.
- On ACX1100 routers, 4-port or 8-port Gigabit Ethernet MICs with SFP transceivers and RJ45 connectors.

- On ACX2000 routers, 8-port Gigabit Ethernet MICs with SFP transceivers and RJ45 connectors.
- On ACX2100 and ACX2200 routers, 4-port Gigabit Ethernet MICs with SFP transceivers and RJ45 connectors.
- On ACX4000 routers, 4-port, 6-port, or 8-port Gigabit Ethernet MICs with SFP transceivers and RJ45 connectors.

> **NOTE**: We recommend running the TDR test on an interface when there is no traffic on the interface.
>
> TDR diagnostics are applicable for copper ports only and not for optical fiber ports.

To diagnose a cable problem by running the TDR test:

1. Run the `request diagnostics tdr` command.

```
user@host> request diagnostics tdr start interface ge-0/0/10

Interface TDR detail:
Test status                     : Test successfully executed  ge-0/0/10
```

2. View the results of the TDR test with the `show diagnostics tdr` command.

```
user@host> show diagnostics tdr interface ge-0/0/10

Interface TDR detail:
Interface name                  : ge-0/0/10
Test status                     : Passed
Link status                     : Down
MDI pair                        : 1-2
  Cable status                  : Normal
  Distance fault                : 0 Meters
  Polartiy swap                 : N/A
  Skew time                     : N/A
MDI pair                        : 3-6
  Cable status                  : Normal
  Distance fault                : 0 Meters
  Polartiy swap                 : N/A
  Skew time                     : N/A
MDI pair                        : 4-5
  Cable status                  : Open
  Distance fault                : 1 Meters
  Polartiy swap                 : N/A
  Skew time                     : N/A
MDI pair                        : 7-8
  Cable status                  : Normal
  Distance fault                : 0 Meters
  Polartiy swap                 : N/A
  Skew time                     : N/A
Channel pair                    : 1
  Pair swap                     : N/A
Channel pair                    : 2
  Pair swap                     : N/A
Downshift                       : N/A
```

3. Examine the `Cable status` field for the four MDI pairs to determine if the cable has a fault. In the preceding example, the twisted pair on pins 4 and 5 is broken or cut at approximately one meter from the `ge-0/0/10` port connection.

> **NOTE**: The `Test Status` field indicates the status of the TDR test, not the cable. The value `Passed` means the test completed—it does not mean that the cable has no faults.

The following is additional information about the TDR test:

- The TDR test can take some seconds to complete. If the test is still running when you execute the `show diagnostics tdr` command, the `Test status` field displays `Started`. For example:

```
user@host> show diagnostics tdr interface ge-0/0/22

Interface TDR detail:
Interface name                  : ge-0/0/22
Test status                     : Started
```

- You can terminate a running TDR test before it completes by using the `request diagnostics tdr abort interface` *`interface-name`* command. The test terminates with no results, and the results from any previous test are cleared.

- You can display summary information about the last TDR test results for all interfaces on the router that support the TDR test by not specifying an interface name with the `show diagnostics tdr` command. For example:

```
user@host> show diagnostics tdr
Interface   Test status    Link status  Cable status  Max distance fault
 ge-0/0/0        Passed     UP           OK             0
 ge-0/0/1   Not Started    N/A          N/A           N/A
 ge-0/0/2        Passed     UP           OK             0
 ge-0/0/3   Not Started    N/A          N/A           N/A
 ge-0/0/4        Passed     UP           OK             0
 ge-0/0/5        Passed     UP           OK             0
 ge-0/0/6        Passed     UP           OK             0
 ge-0/0/7   Not Started    N/A          N/A           N/A
 ge-0/0/8        Passed    Down          OK             0
 ge-0/0/9   Not Started    N/A          N/A           N/A
ge-0/0/10        Passed    Down         Fault           1
ge-0/0/11        Passed     UP           OK             0
ge-0/0/12   Not Started    N/A          N/A           N/A
ge-0/0/13   Not Started    N/A          N/A           N/A
ge-0/0/14   Not Started    N/A          N/A           N/A
ge-0/0/15   Not Started    N/A          N/A           N/A
ge-0/0/16   Not Started    N/A          N/A           N/A
ge-0/0/17   Not Started    N/A          N/A           N/A
ge-0/0/18   Not Started    N/A          N/A           N/A
ge-0/0/19        Passed    Down          OK             0
ge-0/0/20   Not Started    N/A          N/A           N/A
ge-0/0/21   Not Started    N/A          N/A           N/A
ge-0/0/22        Passed     UP           OK             0
ge-0/0/23   Not Started    N/A          N/A           N/A
```

SEE ALSO

Time Domain Reflectometry on ACX Series Routers Overview | 327

request diagnostics tdr

show diagnostics tdr

4

CHAPTER

Configuration Statements and Operational Commands

IN THIS CHAPTER

- Common Output Fields Description | 335
- Improvements to Interface Transmit Statistics Reporting | 345
- Junos CLI Reference Overview | 346

Common Output Fields Description

IN THIS SECTION

- Damping Field | 335
- Destination Class Field | 336
- Enabled Field | 336
- Filters Field | 337
- Flags Fields | 337
- Label-Switched Interface Traffic Statistics Field | 342
- Policer Field | 343
- Protocol Field | 343
- RPF Failures Field | 344
- Source Class Field | 344

This chapter explains the content of the output fields, which appear in the output of most **show interfaces** commands.

Damping Field

For the physical interface, the Damping field shows the setting of the following damping parameters:

- `half-life`—Decay half-life. The number of seconds after which the accumulated interface penalty counter is reduced by half if the interface remains stable.
- `max-suppress`—Maximum hold-down time. The maximum number of seconds that an interface can be suppressed irrespective of how unstable the interface has been.
- `reuse`—Reuse threshold. When the accumulated interface penalty counter falls below this number, the interface is no longer suppressed.
- `suppress`—Cutoff (suppression) threshold. When the accumulated interface penalty counter exceeds this number, the interface is suppressed.

- `state`—Interface damping state. If damping is enabled on an interface, it is suppressed during interface flaps that match the configured damping parameters.

Destination Class Field

For the logical interface, the `Destination class` field provides the names of destination class usage (DCU) counters per family and per class for a particular interface. The counters display packets and bytes arriving from designated user-selected prefixes. For example:

```

                                           Packets                          Bytes
Destination class                  (packet-per-second)               (bits-per-second)

             gold                            1928095                     161959980
                               (               889)                (   597762)
            bronze                               0                          0
                               (                 0)                (        0)
            silver                               0                          0
                               (                 0)                (        0)
```

Enabled Field

For the physical interface, the `Enabled` field provides information about the state of the interface, displaying one or more of the following values:

- `Administratively down, Physical link is Down`—The interface is turned off, and the physical link is inoperable and cannot pass packets even when it is enabled.To change the interface state to `Enabled`, use the following command:

  ```
  user@host# set interfaces interface enable
  ```

  Manually verify the connections to bring the physical link up.

- `Administratively down, Physical link is Up`—The interface is turned off, but the physical link is operational and can pass packets when it is enabled.To change the interface state to `Enabled`, use the following command:

  ```
  user@host# set interfaces interface enable
  ```

- `Enabled, Physical link is Down`—The interface is turned on, but the physical link is inoperable and cannot pass packets. Manually verify the connections to bring the physical link up.

- `Enabled, Physical link is Up`—The interface is turned on, and the physical link is operational and can pass packets.

Filters Field

For the logical interface, the `Filters` field provides the name of the firewall filters to be evaluated when packets are received or transmitted on the interface. The format is `Filters: Input:` *`filter-name`* and `Filters: Output:` *`filter-name`*. For example:

```
Filters: Input: sample-all
Filters: Output: cp-ftp
```

Flags Fields

The following sections provide information about flags that are specific to interfaces:

Addresses, Flags Field

The `Addresses, Flags` field provides information about the addresses configured for the protocol family on the logical interface and displays one or more of the following values:

- `Dest-route-down`—The routing process detected that the link was not operational and changed the interface routes to nonforwarding status

- `Is-Default`—The default address of the router used as the source address by SNMP, ping, traceroute, and other network utilities.

- `Is-Preferred`—The default local address for packets originating from the local router and sent to destinations on the subnet.
- `Is-Primary`—The default local address for broadcast and multicast packets originated locally and sent out the interface.
- `Preferred`—This address is a candidate to become the preferred address.
- `Primary`—This address is a candidate to become the primary address.
- `Trunk`—Interface is a trunk.
- `Trunk, Inter-Switch-Link`—Interface is a trunk, and InterSwitch Link protocol (ISL) is configured on the trunk port of the primary VLAN in order to connect the routers composing the PVLAN to each other.

Device Flags Field

The `Device flags` field provides information about the physical device and displays one or more of the following values:

- `ASIC Error`—Device is down because of ASIC wedging and due to which PFE is disabled.
- `Down`—Device has been administratively disabled.
- `Hear-Own-Xmit`—Device receives its own transmissions.
- `Link-Layer-Down`—The link-layer protocol has failed to connect with the remote endpoint.
- `Loopback`—Device is in physical loopback.
- `Loop-Detected`—The link layer has received frames that it sent, thereby detecting a physical loopback.
- `No-Carrier`—On media that support carrier recognition, no carrier is currently detected.
- `No-Multicast`—Device does not support multicast traffic.
- `Present`—Device is physically present and recognized.
- `Promiscuous`—Device is in promiscuous mode and recognizes frames addressed to all physical addresses on the media.
- `Quench`—Transmission on the device is quenched because the output buffer is overflowing
- `Recv-All-Multicasts`—Device is in multicast promiscuous mode and therefore provides no multicast filtering.
- `Running`—Device is active and enabled.

Family Flags Field

The `Family flags` field provides information about the protocol family on the logical interface and displays one or more of the following values:

- `DCU`—Destination class usage is enabled.
- `Dest-route-down`—The software detected that the link is down and has stopped forwarding the link's interface routes.
- `Down`—Protocol is inactive.
- `Is-Primary`—Interface is the primary one for the protocol.
- `Mac-Validate-Loose`—Interface is enabled with loose MAC address validation.
- `Mac-Validate-Strict`—Interface is enabled with strict MAC address validation.
- `Maximum labels`—Maximum number of MPLS labels configured for the MPLS protocol family on the logical interface.
- `MTU-Protocol-Adjusted`—The effective MTU is not the configured value in the software.
- `No-Redirects`—Protocol redirects are disabled.
- `Primary`—Interface can be considered for selection as the primary family address.
- `Protocol-Down`—Protocol failed to negotiate correctly.
- `SCU-in`—Interface is configured for source class usage input.
- `SCU-out`—Interface is configured for source class usage output.
- `Sendbcast-pkt-to-re`—Interface is configured to forward IPv4 broadcast packets to the Routing Engine.
- `targeted-broadcast`—Interface is configured to forward IPv4 broadcast packets to the LAN interface and the Routing Engine.
- `Unnumbered`—Protocol family is configured for unnumbered Ethernet. An unnumbered Ethernet interface borrows an IPv4 address from another interface, which is referred to as the donor interface.
- `Up`–Protocol is configured and operational.
- `uRPF`—Unicast Reverse Path Forwarding is enabled.

Interface Flags Field

The `Interface flags` field provides information about the physical interface and displays one or more of the following values:

- `Admin-Test`—Interface is in test mode and some sanity checking, such as loop detection, is disabled.
- `Disabled`—Interface is administratively disabled.
- `Down`—A hardware failure has occurred.
- `Hardware-Down`—Interface is nonfunctional or incorrectly connected.
- `Link-Layer-Down`—Interface keepalives have indicated that the link is incomplete.
- `No-Multicast`—Interface does not support multicast traffic.
- `No-receive No-transmit`—Passive monitor mode is configured on the interface.
- `OAM-On-SVLAN`—(Routers with MPC/MIC interfaces only) Interface is configured to propagate the Ethernet OAM state of a static, single-tagged service VLAN (S-VLAN) on a Gigabit Ethernet, 10-Gigabit Ethernet, or aggregated Ethernet interface to a dynamic or static double-tagged customer VLAN (C-VLAN) that has the same S-VLAN (outer) tag as the S-VLAN.
- `Point-To-Point`—Interface is point-to-point.
- `Pop all MPLS labels from packets of depth`—MPLS labels are removed as packets arrive on an interface that has the `pop-all-labels` statement configured. The depth value can be one of the following:
  - `1`—Takes effect for incoming packets with one label only.
  - `2`—Takes effect for incoming packets with two labels only.
  - `[ 1 2 ]`—Takes effect for incoming packets with either one or two labels.
- `Promiscuous`—Interface is in promiscuous mode and recognizes frames addressed to all physical addresses.
- `Recv-All-Multicasts`—Interface is in multicast promiscuous mode and provides no multicast filtering.
- `SNMP-Traps`—SNMP trap notifications are enabled.
- `Up`—Interface is enabled and operational.

Link Flags Field

The `Link flags` field provides information about the physical link and displays one or more of the following values:

- ACFC—Address control field compression is configured. The Point-to-Point Protocol (PPP) session negotiates the ACFC option.
- Give-Up—Link protocol does not continue connection attempts after repeated failures.
- Loose-LCP—PPP does not use the Link Control Protocol (LCP) to indicate whether the link protocol is operational.
- Loose-LMI—Frame Relay does not use the Local Management Interface (LMI) to indicate whether the link protocol is operational.
- Loose-NCP—PPP does not use the Network Control Protocol (NCP) to indicate whether the device is operational.
- No-Keepalives—Link protocol keepalives are disabled.
- PFC—Protocol field compression is configured. The PPP session negotiates the PFC option.

Logical Interface Flags Field

The Logical interface flags field provides information about the logical interface and displays one or more of the following values:

- ACFC Encapsulation—Address control field Compression (ACFC) encapsulation is enabled (negotiated successfully with a peer).
- Device-down—Device has been administratively disabled.
- Disabled—Interface is administratively disabled.
- Down—A hardware failure has occurred.
- Clear-DF-Bit—GRE tunnel or IPsec tunnel is configured to clear the Don't Fragment (DF) bit.
- Hardware-Down—Interface protocol initialization failed to complete successfully.
- PFC—Protocol field compression is enabled for the PPP session.
- Point-To-Point—Interface is point-to-point.
- SNMP-Traps—SNMP trap notifications are enabled.
- Up—Interface is enabled and operational.

Label-Switched Interface Traffic Statistics Field

When you use the `vrf-table-label` statement to configure a VRF routing table, a label-switched interface (LSI) logical interface label is created and mapped to the VRF routing table.

Any routes present in a VRF routing table and configured with the `vrf-table-label` statement are advertised with the LSI logical interface label allocated for the VRF routing table. When packets for this VPN arrive on a core-facing interface, they are treated as if the enclosed IP packet arrived on the LSI interface and are then forwarded and filtered based on the correct table. For more information on the `vrf-table-label` statement, including a list of supported interfaces, see the *Junos VPNs Configuration Guide*.

If you configure the `family mpls` statement at the `[edit interfaces` *`interface-name`* `unit` *`logical-unit-number`*`]` hierarchy level and you also configure the `vrf-table-label` statement at the `[edit routing-instances` *`routing-instance-name`*`]` hierarchy level, the output for the `show interface` *`interface-name`* `extensive` command includes the following output fields about the LSI traffic statistics:

- `Input bytes`—Number of bytes entering the LSI and the current throughput rate in bits per second (bps).
- `Input packets`—Number of packets entering the LSI and the current throughput rate in packets per second (pps).

> **NOTE**: If LSI interfaces are used with VPLS when `no-tunnel-services` is configured or L3VPN when `vrf-table-label` configuration is applied inside the routing-instance, the `Input packets` field associated with the core-facing interfaces may not display the correct value. Only the Input counter is affected because the LSI is used to receive traffic from the remote PEs. Traffic that arrives on an LSI interface might not be counted at both the Traffic Statistics and the Label-switched interface (LSI) traffic statistics levels.
>
> This note applies to the following platforms:
>
> - Routers with DPC or ADPC only

The following example shows the LSI traffic statistics that you might see as part of the output of the `show interface` *`interface-name`* `extensive` command:

```
Label-switched interface (LSI) traffic statistics:
   Input  bytes:                         0                 0 bps
   Input  packets:                     0                 0 pps
```

Policer Field

For the logical interface, the `Policer` field provides the policers that are to be evaluated when packets are received or transmitted on the interface. The format is `Policer: Input:` *type-fpc/picport*`-in-policer, Output:` *type-fpc/pic/port*`-out-policer`. For example:

```
Policer: Input: at-1/2/0-in-policer, Output: at-2/4/0-out-policer
```

Protocol Field

For the logical interface, the `Protocol` field indicates the protocol family or families that are configured on the interface, displaying one or more of the following values:

- `aenet`—Aggregated Ethernet. Displayed on Fast Ethernet interfaces that are part of an aggregated Ethernet bundle.
- `ccc`—Circuit cross-connect (CCC). Configured on the logical interface of CCC physical interfaces.
- `inet`—IP version 4 (IPv4). Configured on the logical interface for IPv4 protocol traffic, including Open Shortest Path First (OSPF), Border Gateway Protocol (BGP), Internet Control Message Protocol (ICMP), and Internet Protocol Control Protocol (IPCP).
- `inet6`—IP version 6 (IPv6). Configured on the logical interface for IPv6 protocol traffic, including Routing Information Protocol for IPv6 (RIPng), Intermediate System-to-Intermediate System (IS-IS), and BGP.
- `iso`—International Organization for Standardization (ISO). Configured on the logical interface for IS-IS traffic.
- `mlfr-uni-nni`—Multilink Frame Relay (MLFR) FRF.16 user-to-network network-to-network (UNI NNI). Configured on the logical interface for link services bundling.
- `mlfr-end-to-end`—Multilink Frame Relay end-to-end. Configured on the logical interface for multilink bundling.
- `mlppp`—Multilink Point-to-Point Protocol (MLPPP). Configured on the logical interface for multilink bundling.
- `mpls`—Multiprotocol Label Switching (MPLS). Configured on the logical interface for participation in an MPLS path.

- pppoe— Point-to-Point Protocol over Ethernet (PPPoE). Configured on Ethernet interfaces enabled to support multiple protocol families.
- tcc—Translational cross-connect (TCC). Configured on the logical interface of TCC physical interfaces.
- tnp—Trivial Network Protocol (TNP). Used to communicate between the Routing Engine and the router's packet forwarding components. The Junos OS automatically configures this protocol family on the router's internal interfaces only.
- vpls—Virtual private LAN service (VPLS). Configured on the logical interface on which you configure VPLS.

RPF Failures Field

For the logical interface, the RPF Failures field provides information about the amount of incoming traffic (in packets and bytes) that failed a unicast reverse path forwarding (RPF) check on a particular interface. The format is RPF Failures: Packets: xx,Bytes: yy. For example:

```
RPF Failures: Packets: 0, Bytes:0
```

Source Class Field

For the logical interface, the Source class field provides the names of source class usage (SCU) counters per family and per class for a particular interface. The counters display packets and bytes arriving from designated user-selected prefixes. For example:

```
                                                       Packets
Bytes
Source class                    (packet-per-second)                  (bits-per-second)


                                 gold            1928095                         161959980
                                            (               889)                 (          5977
62)
                              bronze                        0
    0
                                            (                   0)                  (
         0)
                                  silver                         0
        0
                                             (                    0)                 (
         0)
```

Improvements to Interface Transmit Statistics Reporting

The offered load on an interface can be defined as the amount of data the interface is capable of transmitting during a given time period. The actual traffic that goes out of the interface is the transmitted load. However, when outgoing interfaces are oversubscribed, there could be traffic drops in the schedulers attached to the outgoing interfaces. Hence, the offered load is not always the same as the actual transmitted load because the offered load calculation does not take into account possible packet drop or traffic loss.

On routers, the logical interface-level statistics show the offered load, which is often different from the actual transmitted load. To address this limitation, Junos OS introduces a configuration option, `interface-transmit-statistics`, at the `[edit interface` *`interface-name`*`]` hierarchy level, enables you to configure Junos OS to accurately capture and report the transmitted load on interfaces.

Aggregated Ethernet interfaces do not support reporting of the transmitted load statistics. You cannot configure aggregated Ethernet interfaces to capture and report the actual transmitted load statistics.

When the `interface-transmit-statistics` statement is included at the `[edit interface` *`interface-name`*`]` hierarchy level, the following operational mode commands report the actual transmitted load:

- `show interface` *`interface-name`* `<detail | extensive>`
- `monitor interface` *`interface-name`*
- `show snmp mib get` *`objectID.ifIndex`*

The `show interface` *`interface-name`* command also shows whether the `interface-transmit-statistics` configuration is enabled or disabled on the interface.

RELATED DOCUMENTATION

interface-transmit-statistics

show interfaces

Junos CLI Reference Overview

We've consolidated all Junos CLI commands and configuration statements in one place. Read this guide to learn about the syntax and options that make up the statements and commands. Also understand the contexts in which you'll use these CLI elements in your network configurations and operations.

- Junos CLI Reference

Click the links to access Junos OS and Junos OS Evolved configuration statement and command summary topics.

- Configuration Statements
- Operational Commands