JUNIPER
NETWORKS | Engineering
Simplicity

Junos® OS for EX Series Ethernet Switches

Class of Service User Guide (EX Series Switches Except EX4600 Switches)

Published
2025-12-12

JUNOS

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

YEAR 2000 NOTICE

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

END USER LICENSE AGREEMENT

Table of Contents

3   **CoS on Overlay Networks**

4

Configuration Statements and Operational Commands

About This Guide

Use this guide to understand and configure class of service (CoS) features in Junos OS to define service levels that provide different delay, jitter, and packet loss characteristics to particular applications served by specific traffic flows. Applying CoS features to each device in your network ensures quality of service (QoS) for traffic throughout your entire network. This guide applies to all EX Series switches except the EX4600 lines of switches.

1

PART

Basic CoS Configuration

CoS Overview

IN THIS CHAPTER

Junos OS CoS for EX Series Switches Overview

IN THIS SECTION

When a network experiences congestion and delay, some packets must be dropped. Junos operating system (Junos OS) *class of service* (CoS) divides traffic into classes to which you can apply different levels of throughput and packet loss when congestion occurs. This allows packet loss to happen according to rules that you configure.

For interfaces that carry IPv4, IPv6, and MPLS traffic, you can configure Junos OS CoS features to provide multiple classes of service for different applications. CoS also allows you to rewrite the Differentiated Services code point (DSCP), IP precedence, 802.1p, or EXP CoS bits of packets egressing out of an interface, thus allowing you to tailor packets for the remote peers' network requirements. See *Understanding Using CoS with MPLS Networks on EX Series Switches* for more information about CoS for MPLS networks.

CoS provides multiple classes of service for different applications. You can configure multiple forwarding classes for transmitting packets, define which packets are placed into each output queue, and schedule the transmission service level for each queue.

In designing CoS applications, you must give careful consideration to your service needs and thoroughly plan and design your CoS configuration to ensure consistency and interoperability across all platforms in a CoS domain.

Because Juniper Networks EX Series Ethernet Switches implement CoS in hardware rather than in software, you can experiment with and deploy CoS features without affecting packet-forwarding and switching performance.

> (i) **NOTE**: CoS policies can be enabled or disabled on each interface of an EX Series switch. Also, each physical and *logical interface* on the switch can have custom CoS rules associated with it. When CoS is used in an MPLS network, there are some additional restrictions. See *Understanding Using CoS with MPLS Networks on EX Series Switches*.

How Junos OS CoS Works

Junos OS CoS works by examining traffic entering at the edge of your network. The switches classify traffic into defined service groups to provide the special treatment of traffic across the network. For example, voice traffic can be sent across certain links, and data traffic can use other links. In addition, the data traffic streams can be serviced differently along the network path. As the traffic leaves the network at the far edge, you can rewrite the traffic to meet the policies of the targeted peer.

To support CoS, you must configure each switch in the network. Generally, each switch examines the packets that enter it to determine their CoS settings. These settings then dictate which packets are transmitted first to the next downstream switch. Switches at the edges of the network might be required to alter the CoS settings of the packets that enter the network to classify the packets into the appropriate service groups.

Figure 1 on page 4 represents the network scenario of an enterprise. Switch A is receiving traffic from various network nodes such as desktop computers, servers, surveillance cameras, and VoIP telephones. As each packet enters, Switch A examines the packet's CoS settings and classifies the traffic into one of the groupings defined by the enterprise. This definition allows Switch A to prioritize resources for servicing the traffic streams it receives. Switch A might alter the CoS settings of the packets to better match the enterprise's traffic groups.

When Switch B receives the packets, it examines the CoS settings, determines the appropriate traffic groups, and processes the packets according to those settings. It then transmits the packets to Switch C, which performs the same actions. Switch D also examines the packets and determines the appropriate groups. Because Switch D sits at the far end of the network, it can rewrite the CoS bits of the packets before transmitting them.

Figure 1: Packet Flow Across the Network
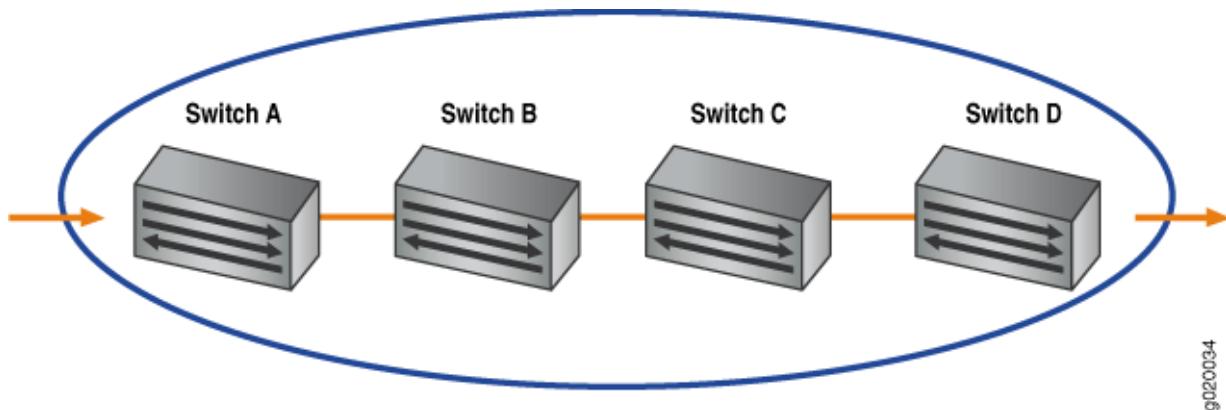
Default CoS Behavior on EX Series Switches

If you do not configure any CoS settings on the switch, the software still ensures that user traffic and protocol packets are forwarded with minimum delay when the network is experiencing congestion. Some CoS settings, such as classifiers, are automatically applied to each logical interface that you configure. Other settings, such as *rewrite rules*, are applied only if you explicitly associate them with an interface.

RELATED DOCUMENTATION

Understanding Junos OS CoS Components for EX Series Switches | **8**

Understanding Junos OS EZQoS for CoS Configurations on EX Series Switches | **48**

Example: Configuring CoS on EX Series Switches | **11**

Example: Combining CoS with MPLS on EX Series Switches

Configuring CoS on EX Series Switches

The topics in this guide describe how to configure the Junos OS class-of-service (CoS) components. Junos CoS provides a flexible set of tools that enable you to fine tune control over the traffic on your network.

- Define classifiers that classify incoming traffic into forwarding classes to place traffic in groups for transmission.

- Map forwarding classes to output queues to define the type of traffic on each output queue.

- Configure schedulers for each output queue to control the service level (priority, bandwidth characteristics) of each type of traffic.

- Provide different service levels for the same forwarding classes on different interfaces.

- Provide congestion management with tail drop profiles, queue shaping, and congestion notification.

- Configure CoS on MPLS networks.

- Configure various CoS components individually or in combination to define CoS services.

> **NOTE**: When you change the CoS configuration or when you deactivate and then reactivate the CoS configuration, the system experiences packet drops because the system momentarily blocks traffic to change the mapping of incoming traffic to input queues.

Table 1 on page 6 lists the primary CoS configuration tasks, includes platform limitations, and provides links to those tasks.

Table 1: CoS Configuration Tasks

CoS Configuration Task	Links

Basic CoS Configuration:
 	
• Configure CoS using EZQoS with templates for key traffic classes, or a browser and the J-Web interface.
 	
• Configure code-point aliases to assign a name to a pattern of code-point bits that you can use instead of the bit pattern when you configure CoS components such as classifiers and rewrite rules.
 	
• Configure classifiers and multidestination classifiers.
 	
   • Configure rewrite rules to alter code-point bit values in outgoing packets on the outbound interfaces of a switch so that the CoS treatment matches the policies of a targeted peer.
 	
   • Set the forwarding class and loss priority of a packet based on the incoming CoS value and assign packets to output queues based on the associated forwarding class.
 	
• Configure forwarding classes.
 	
• Configure priority-based flow control to apply link-level flow control on a specific traffic class so that different types of traffic can efficiently use the same network interface card (NIC).
 	
• Configure CoS schedulers to define the properties of output queues on EX Series switches. These properties include the amount of interface bandwidth assigned to the queue, the size of the memory buffer allocated for storing packets, the priority of the queue, and the drop profiles associated with the queue.
 	
• Assign the following CoS components to physical or logical interfaces:
 	
   • Classifiers (logical interfaces only)
	• "Configuring Junos OS EZQoS for CoS (CLI Procedure)" on page 49
	
	• "Configuring CoS (J-Web Procedure)" on page 50
	
	• "Defining CoS Code-Point Aliases (CLI Procedure)" on page 62
	
	• "Defining CoS Classifiers (CLI Procedure)" on page 70
	
	• "Example: Configuring Multidestination (Multicast, Broadcast, DLF) Classifiers" on page 75
	
	• "Defining CoS Rewrite Rules (CLI Procedure)" on page 91
	
	• "Defining CoS Forwarding Classes (CLI Procedure)" on page 111
	
	• "Configuring Priority-Based Flow Control for an EX Series Switch (CLI Procedure)" on page 120
	
	• "Defining CoS Schedulers and Scheduler Maps (CLI Procedure)" on page 129
	
	• "Assigning CoS Components to Interfaces (CLI Procedure)" on page 52

Table 1: CoS Configuration Tasks *(Continued)*

CoS Configuration Task	Links
• Forwarding classes (logical interfaces only) • Scheduler maps • Rewrite rules	
Configure congestion management mechanisms for a switch to drop arriving packets based on certain parameters when a queue is full. Based on the EX Series switch that you are using, packets are dropped depending on the priority of a packet or on both priority and drop probability of a packet. • Configure a weighted tail drop profile, a simple and effective traffic congestion avoidance mechanism. When you apply this mechanism to manage congestion, packets are dropped when the output queue is full. • Configure a weighted random early detection (WRED) drop profile. When the configured capacity (fill level) is reached, packets marked with a packet loss priority (PLP) of high are discarded. • Configure port shaping and queue shaping to enable you to limit traffic on an interface or queue, respectively, so that you can control the amount of traffic passing through the interface or the queue. • Configure explicit congestion notification (ECN) to enable end-to-end congestion notification between two endpoints on TCP/IP based networks. Apply WRED drop profiles to forwarding classes to control how the switch marks ECN-capable packets.	• "Configuring CoS Congestion Management (CLI Procedure)" on page 148 • "Configuring Shaping for CoS (CLI Procedure)" on page 159 • "Example: Configuring Static and Dynamic ECN" on page 179
Configure CoS on MPLS networks to ensure better performance for low-latency applications such as VoIP and other business-critical functions.	• "Example: Combining CoS with MPLS on EX Series Switches" on page 192

Understanding Junos OS CoS Components for EX Series Switches

This topic describes the Juniper Networks Junos operating system (Junos OS) class-of-service (CoS) components for Juniper Networks EX Series Ethernet Switches:

Code-Point Aliases

A code-point alias assigns a name to a pattern of code-point bits. You can use this name instead of the bit pattern when you configure other CoS components such as classifiers, drop-profile maps, and *rewrite rules*.

Policers

Policers limit traffic of a certain class to a specified bandwidth and *burst size*. Packets exceeding the policer limits can be discarded. You define policers with filters that can be associated with input interfaces.

For more information about policers, see *Understanding the Use of Policers in Firewall Filters*.

> **NOTE**: You can configure policers to discard packets that exceed the rate limits. If you want to configure CoS parameters such as **loss-priority** and **forwarding-class**, you must use firewall filters.

Classifiers

Packet classification associates incoming packets with a particular CoS servicing level. In Juniper Networks Junos operating system (Junos OS), *classifiers* associate packets with a forwarding class and loss priority and assign packets to output queues based on the associated forwarding class. Junos OS supports two general types of classifiers:

- Behavior aggregate or CoS value traffic classifiers—Examines the CoS value in the packet header. The value in this single field determines the CoS settings applied to the packet. BA classifiers allow you to set the forwarding class and loss priority of a packet based on the Differentiated Services code point (DSCP) value, IP precedence value, and IEEE 802.1p value.

- Multifield traffic classifiers—Examines multiple fields in the packet such as source and destination addresses and source and destination port numbers of the packet. With multifield classifiers, you set the forwarding class and loss priority of a packet based on *firewall filter* rules.

Forwarding Classes

Forwarding classes group the packets for transmission. Based on forwarding classes, you assign packets to output queues. Forwarding classes affect the forwarding, scheduling, and marking policies applied to packets as they transit a switch. By default, four categories of forwarding classes are defined: best effort, assured forwarding, expedited forwarding, and network control. For EX Series switches, 16 forwarding classes are supported, providing granular classification capability.

Tail Drop Profiles

Drop profile is a mechanism that defines parameters that allow packets to be dropped from the network. Drop profiles define the meanings of the loss priorities. When you configure drop profiles you are essentially setting the value for queue fullness. The queue fullness represents a percentage of the queue used to store packets in relation to the total amount that has been allocated for that specific queue.

Loss priorities set the priority of dropping a packet. Loss priority affects the scheduling of a packet without affecting the packet's relative ordering. You can use the loss priority setting to identify packets that have experienced congestion. Typically you mark packets exceeding some service level with a high loss priority.

Schedulers

Each switch interface has multiple queues assigned to store packets. The switch determines which queue to service based on a particular method of scheduling. This process often involves determining which type of packet should be transmitted before another. You can define the priority, bandwidth, delay buffer size, and tail drop profiles to be applied to a particular queue for packet transmission.

A scheduler map associates a specified forwarding class with a scheduler configuration. You can associate up to four user-defined scheduler maps with the interfaces.

Rewrite Rules

A *rewrite rule* sets the appropriate CoS bits in the outgoing packet, thus allowing the next downstream device to classify the packet into the appropriate service group. Rewriting, or marking, outbound packets is useful when the switch is at the border of a network and must alter the CoS values to meet the policies of the targeted peer.

> **NOTE**: Egress firewall filters can also assign forwarding class and loss priority so that the packets are rewritten based on forwarding class and loss priority.

RELATED DOCUMENTATION

Understanding CoS Two-Color Marking

Networks police traffic by limiting the input or output transmission rate of a class of traffic on the basis of user-defined criteria. Policing traffic allows you to control the maximum rate of traffic sent or received on an interface and to partition a network into multiple priority levels or classes of service.

Policers require you to apply limits to the traffic flow and set a consequence for packets that exceed these limits—usually a higher loss priority, so that packets exceeding the policer limits are discarded first.

Juniper Networks EX Series Ethernet Switches support a single-rate two-color marking type of policer, which is a simplified version of Single-Rate-Three-Color marking, defined in RFC 2697, *A Single Rate Three Color Marker*. This type of policer meters traffic based on the configured committed information rate (CIR) and committed burst size (CBS).

The single-rate two-color marker meters traffic and marks incoming packets depending on whether they are smaller than the committed burst size (CBS)—marked green—or exceed it— marked red.

The single-rate two-color marking policer operates in color-blind mode. In this mode, the policer's actions are not affected by any previous marking or metering of the examined packets. In other words, the policer is "blind? to any previous coloring a packet might have had.

RELATED DOCUMENTATION

Understanding the Use of Policers in Firewall Filters

Configuring Policers to Control Traffic Rates (CLI Procedure)

Example: Configuring CoS on EX Series Switches

IN THIS SECTION

Configure class of service (CoS) on your switch to manage traffic so that when the network experiences congestion and delay, critical applications are protected. Using CoS, you can divide traffic on your switch into classes and provide various levels of throughput and packet loss. This is especially important for traffic that is sensitive to jitter and delay, such as voice traffic.

This example shows how to configure CoS on a single EX Series switch in the network.

Requirements

This example uses the following hardware and software components:

- EX Series switches

- Any supported Junos OS release

Overview and Topology

This example uses the topology shown in .

Figure 2: Topology for Configuring CoS
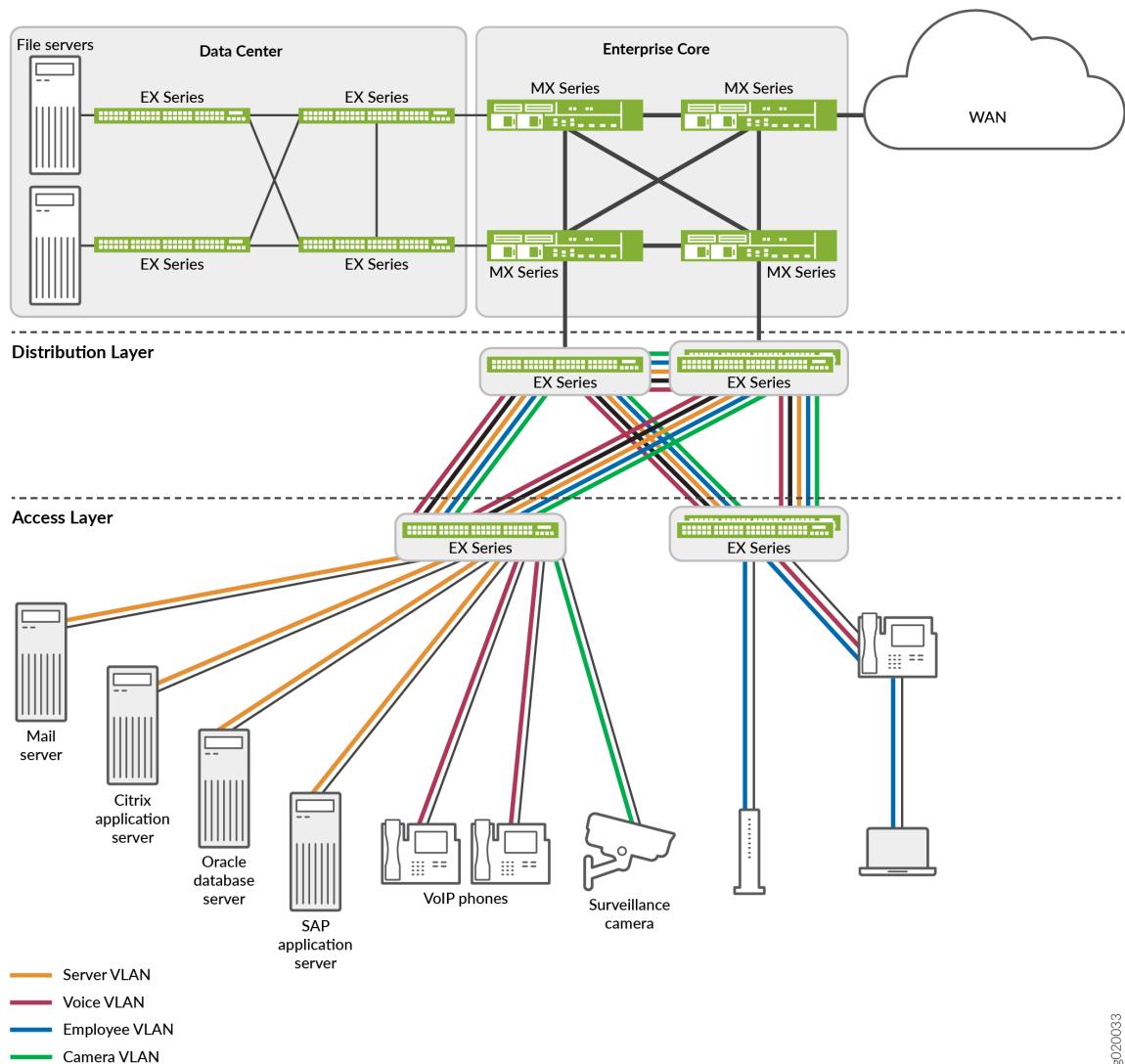
g020033

The topology for this configuration example consists of EX switches at the access layer.

The EX Series access switches are configured to support VLAN membership. On the access layer switch, interfaces ge-0/0/0 and ge-0/0/1 are assigned to the voice VLAN (voice-vlan) for two VoIP IP phones. Switch interface ge-0/0/2 is assigned to the camera VLAN (camera-vlan) for the surveillance camera. Switch interfaces ge-0/0/3, ge-0/0/4, ge-0/0/5, and ge-0/0/6 are assigned to the server VLAN (server-vlan) for the servers hosting various applications such as those provided by Citrix, Microsoft, Oracle, and SAP. The trunk ports, ge-0/0/20 and ge-0/0/21, are assigned to the server, voice, employee, and camera VLANs and used as uplink ports to connect the distribution layer switches.

shows the VLAN configuration components.

Table 2: Configuration Components: VLANs

VLAN Name	VLAN ID	VLAN Subnet and Available IP Addresses	VLAN Description
voice-vlan	10	192.168.1.0/28 192.168.1.1 through 192.168.1.14 192.168.1.15 is the subnet's broadcast address.	Voice VLAN used for employee VoIP communication.
camera-vlan	20	192.168.1.16/28 192.168.1.17 through 192.168.1.30 192.168.1.31 is the subnet's broadcast address.	VLAN for the surveillance cameras.
server-vlan	30	192.168.1.32/28 192.168.1.33 through 192.168.1.46 192.168.1.47 is the subnet's broadcast address.	VLAN for the servers hosting enterprise applications.

PoE-capable ports on EX Series switches support Power over Ethernet (PoE) to provide both network connectivity and power for VoIP telephones connecting to the ports. shows the

switch interfaces that are assigned to the VLANs and the IP addresses for devices connected to the switch ports on a 48-port switch, all ports of which are PoE-capable.

Table 3: Configuration Components: Switch Interfaces Assigned to VLANs and Devices on a 48-Port All-PoE Switch

Interfaces	VLAN Membership	IP Addresses	Port Devices
ge-0/0/0, ge-0/0/1	voice-vlan	192.168.1.1/28 through 192.168.1.2/28	Two VoIP telephones.
ge-0/0/2	camera-vlan	192.168.1.17/28	Surveillance camera.
ge-0/0/3, ge-0/0/4, ge-0/0/5, ge-0/0/6	server-vlan	192.168.1.33/28 through 192.168.1.36/28	Four servers hosting applications such as those provided by Citrix, Microsoft, Oracle, and SAP.

> (i) **NOTE**: This example shows how to configure CoS on a standalone EX Series switch. This example does not consider across-the-network applications of CoS in which you might implement different configurations on ingress and egress switches to provide differentiated treatment to different classes across a set of nodes in a network.

> (i) **NOTE**: Although you will sometimes see schedulers configured for strict-high priority with a transmit-rate configured, that configuration is misleading because strict-high priority schedulers get unlimited bandwidth and the transmit-rate parameter has no effect on them. With this configuration, lower priority queues can suffer starvation if there is congestion. It is better that schedulers with strict-high priority have shaping-rate parameters configured, which is the correct way to limit their bandwidth.

Configuration

IN THIS SECTION

Procedure

CLI Quick Configuration

To quickly configure CoS, copy the following commands and paste them into the switch terminal window:

```
[edit]
set class-of-service forwarding-classes class app queue-num 5
set class-of-service forwarding-classes class mail queue-num 1
set class-of-service forwarding-classes class db queue-num 2
set class-of-service forwarding-classes class erp queue-num 3
set class-of-service forwarding-classes class video queue-num 4
set class-of-service forwarding-classes class best-effort queue-num 0
set class-of-service forwarding-classes class voice queue-num 6
set class-of-service forwarding-classes class network-control queue-num 7
set firewall family ethernet-switching filter voip_class term voip from source-address
192.168.1.1/28
set firewall family ethernet-switching filter voip_class term voip from source-address
192.168.1.2/28
set firewall family ethernet-switching filter voip_class term voip from protocol udp
set firewall family ethernet-switching filter voip_class term voip from source-port 2698
set firewall family ethernet-switching filter voip_class term voip then forwarding-class voice
loss-priority low
set firewall family ethernet-switching filter voip_class term network_control from precedence
[net-control internet-control]
set firewall family ethernet-switching filter voip_class term network_control then forwarding-
class network-control loss-priority low
set firewall family ethernet-switching filter voip_class term best_effort_traffic then
forwarding-class best-effort loss-priority low
set interfaces ge-0/0/0 description phone1-voip-ingress-port
set interfaces ge-0/0/0 unit 0 family ethernet-switching filter input voip_class
set class-of-service interfaces ge-0/0/0 shaping-rate 100m
set interfaces ge-0/0/1 description phone2-voip-ingress-port
set interfaces ge-0/0/1 unit 0 family ethernet-switching filter input voip_class
set firewall family ethernet-switching filter video_class term video from source-address
192.168.1.17/28
set firewall family ethernet-switching filter video_class term video from protocol udp
set firewall family ethernet-switching filter video_class term video from source-port 2979
set firewall family ethernet-switching filter video_class term video then forwarding-class video
loss-priority low
```

```
set firewall family ethernet-switching filter video_class term network_control from precedence
[net-control internet-control]
set firewall family ethernet-switching filter video_class term network_control then forwarding-
class network-control loss-priority low
set firewall family ethernet-switching filter video_class term best_effort_traffic then
forwarding-class best-effort loss-priority low
set interfaces ge-0/0/2 description video-ingress-port
set interfaces ge-0/0/2 unit 0 family ethernet-switching filter input video_class
set firewall family ethernet-switching filter app_class term app from source-address
192.168.1.33/28
set firewall family ethernet-switching filter app_class term app from protocol tcp
set firewall family ethernet-switching filter app_class term app from source-port [1494 2512
2513 2598 2897]
set firewall family ethernet-switching filter app_class term app then forwarding-class app loss-
priority low
set firewall family ethernet-switching filter app_class term mail from source-address
192.168.1.34/28
set firewall family ethernet-switching filter app_class term mail from protocol tcp
set firewall family ethernet-switching filter app_class term mail from source-port [25 143 389
691 993 3268 3269]
set firewall family ethernet-switching filter app_class term mail then forwarding-class mail
loss-priority low
set firewall family ethernet-switching filter app_class term db from source-address
192.168.1.35/28
set firewall family ethernet-switching filter app_class term db from protocol tcp
set firewall family ethernet-switching filter app_class term db from source-port [1521 1525 1527
1571 1810 2481]
set firewall family ethernet-switching filter app_class term db then forwarding-class db loss-
priority low
set firewall family ethernet-switching filter app_class term erp from source-address
192.168.1.36/28
set firewall family ethernet-switching filter app_class term erp from protocol tcp
set firewall family ethernet-switching filter app_class term erp from source-port [3200 3300
3301 3600]
set firewall family ethernet-switching filter app_class term erp then forwarding-class erp loss-
priority low
set firewall family ethernet-switching filter app_class term network_control from precedence
[net-control internet-control]
set firewall family ethernet-switching filter app_class term network_control then forwarding-
class network-control loss-priority low
set firewall family ethernet-switching filter app_class term best_effort_traffic then forwarding-
class best-effort loss-priority low
set interfaces ge-0/0/3 unit 0 family ethernet-switching filter input app_class
```

```
set interfaces ge-0/0/4 unit 0 family ethernet-switching filter input app_class
set interfaces ge-0/0/5 unit 0 family ethernet-switching filter input app_class
set interfaces ge-0/0/6 unit 0 family ethernet-switching filter input app_class
set class-of-service schedulers voice-sched shaping-rate percent 10
set class-of-service schedulers voice-sched buffer-size percent 10
set class-of-service schedulers voice-sched priority strict-high
set class-of-service schedulers video-sched priority low
set class-of-service schedulers video-sched transmit-rate percent 15
set class-of-service schedulers app-sched buffer-size percent 10
set class-of-service schedulers app-sched priority low
set class-of-service schedulers app-sched transmit-rate percent 10
set class-of-service schedulers mail-sched buffer-size percent 5
set class-of-service schedulers mail-sched priority low
set class-of-service schedulers mail-sched transmit-rate percent 5
set class-of-service schedulers db-sched buffer-size percent 10
set class-of-service schedulers db-sched priority low
set class-of-service schedulers db-sched transmit-rate percent 10
set class-of-service schedulers erp-sched buffer-size percent 10
set class-of-service schedulers erp-sched priority low
set class-of-service schedulers erp-sched transmit-rate percent 10
set class-of-service schedulers nc-sched shaping-rate percent 5
set class-of-service schedulers nc-sched buffer-size percent 5
set class-of-service schedulers nc-sched priority strict-high
set class-of-service schedulers be-sched buffer-size percent 35
set class-of-service schedulers be-sched priority low
set class-of-service schedulers be-sched transmit-rate percent 35
set class-of-service scheduler-maps ethernet-cos-map forwarding-class voice scheduler voice-sched
set class-of-service scheduler-maps ethernet-cos-map forwarding-class video scheduler video-sched
set class-of-service scheduler-maps ethernet-cos-map forwarding-class app scheduler app-sched
set class-of-service scheduler-maps ethernet-cos-map forwarding-class mail scheduler mail-sched
set class-of-service scheduler-maps ethernet-cos-map forwarding-class db scheduler db-sched
set class-of-service scheduler-maps ethernet-cos-map forwarding-class erp scheduler erp-sched
set class-of-service scheduler-maps ethernet-cos-map forwarding-class network-control scheduler
nc-sched
set class-of-service scheduler-maps ethernet-cos-map forwarding-class best-effort scheduler be-
sched
set class-of-service interfaces ge-0/0/20 scheduler-map ethernet-cos-map
set class-of-service interfaces ge-0/0/21 scheduler-map ethernet-cos-map
set class-of-service schedulers voice-sched-queue-shap shaping-rate 30m
set class-of-service scheduler-maps sched-map-be forwarding-class best-effort scheduler voice-
sched-queue-shap
set class-of-service interfaces ge-0/0/2 scheduler-map sched-map-be
```

Step-by-Step Procedure

To configure and apply CoS:

1. Configure one-to-one mappings between eight forwarding classes and eight queues:

```
[edit class-of-service]
user@switch# set forwarding-classes class app queue-num 5
user@switch# set forwarding-classes class mail queue-num 1
user@switch# set forwarding-classes class db queue-num 2
user@switch# set forwarding-classes class erp queue-num 3
user@switch# set forwarding-classes class video queue-num 4
user@switch# set forwarding-classes class best-effort queue-num 0
user@switch# set forwarding-classes class voice queue-num 6
user@switch# set forwarding-classes class network-control queue-num 7
```

2. Define the firewall filter `voip_class` to classify the VoIP traffic:

```
[edit firewall]
user@switch# set family ethernet-switching filter voip_class
```

3. Define the term `voip`:

```
[edit firewall]
user@switch# set family ethernet-switching filter voip_class term voip from source-address
192.168.1.1/28
user@switch# set family ethernet-switching filter voip_class term voip from source-address
192.168.1.2/28
user@switch# set family ethernet-switching filter voip_class term voip protocol udp
user@switch# set family ethernet-switching filter voip_class term voip source-port 2698
user@switch# set family ethernet-switching filter voip_class term voip then forwarding-
class voice loss-priority low
```

4. Define the term `network_control` (for the `voip_class` filter):

```
[edit firewall]
user@switch# set family ethernet-switching filter voip_class term network_control from
precedence [net-control internet-control]
user@switch# set family ethernet-switching filter voip_class term network_control then
forwarding-class network-control loss-priority low
```

5. Define the term `best_effort_traffic` with no match conditions (for the `voip_class` filter):

```
[edit firewall]
user@switch# set family ethernet-switching filter voip_class term best_effort_traffic then
forwarding-class best-effort loss-priority low
```

6. Apply the firewall filter `voip_class` as an input filter to the interfaces for the VoIP phones:

```
[edit interfaces]
user@switch# set ge-0/0/0 description phone1-voip-ingress-port
user@switch# set ge-0/0/0 unit 0 family ethernet-switching filter input voip_class
user@switch# set ge-0/0/1 description phone2-voip-ingress-port
user@switch# set ge-0/0/1 unit 0 family ethernet-switching filter input voip_class
```

7. Apply port shaping on the interface ge-0/0/0:

```
[edit]
user@switch# set class-of-service interfaces ge-0/0/0 shaping-rate 100m
```

8. Define the firewall filter `video_class` to classify the video traffic:

```
[edit firewall]
user@switch# set family ethernet-switching filter video_class
```

9. Define the term `video`:

```
[edit firewall]
user@switch# set family ethernet-switching filter video_class term video from source-
address 192.168.1.17/28
user@switch# set family ethernet-switching filter video_class term video protocol udp
user@switch# set family ethernet-switching filter video_class term video source-port 2979
user@switch# set family ethernet-switching filter video_class term video then forwarding-
class video loss-priority low
```

10. Define the term `network_control` (for the `video_class` filter):

```
[edit firewall]
user@switch# set family ethernet-switching filter video_class term network_control from
precedence [net-control internet-control]
user@switch# set family ethernet-switching filter video_class term network_control then
forwarding-class network-control loss-priority low
```

11. Define the term `best_effort_traffic` with no match conditions (for the `video_class` filter):

```
[edit firewall]
user@switch# set family ethernet-switching filter video_class term best_effort_traffic then
forwarding-class best-effort loss-priority low
```

12. Apply the firewall filter `video_class` as an input filter to the interface for the surveillance camera:

```
[edit interfaces]
user@switch# set ge-0/0/2 description video-ingress-port
user@switch# set ge-0/0/2 unit 0 family ethernet-switching filter input video_class
```

13. Define the firewall filter app_class to classify the application server traffic:

```
[edit firewall]
user@switch# set family ethernet-switching filter app_class
```

14. Define the term app (for the app_class filter):

```
[edit firewall]
user@switch# set family ethernet-switching filter app_class term app from source-address
192.168.1.33/28
user@switch# set family ethernet-switching filter app_class term app protocol tcp
user@switch# set family ethernet-switching filter app_class term app source-port [1494 2512
2513 2598 2897]
user@switch# set family ethernet-switching filter app_class term app then forwarding-class
app loss-priority low
```

15. Define the term mail (for the app_class filter):

```
[edit firewall]
user@switch# set family ethernet-switching filter app_class term mail from source-address
192.168.1.34/28
user@switch# set family ethernet-switching filter app_class term mail protocol tcp
user@switch# set family ethernet-switching filter app_class term mail source-port [25 143
389 691 993 3268 3269]
user@switch# set family ethernet-switching filter app_class term mail then forwarding-class
mail loss-priority low
```

16. Define the term db (for the app_class filter):

```
[edit firewall]
user@switch# set family ethernet-switching filter app_class term db from source-address
192.168.1.35/28
user@switch# set family ethernet-switching filter app_class term db protocol tcp
user@switch# set family ethernet-switching filter app_class term db source-port [1521 1525
1527 1571 1810 2481]
```

```
user@switch# set family ethernet-switching filter app_class term db then forwarding-class
db loss-priority low
```

17. Define the term erp (for the app_class filter):

```
[edit firewall]
user@switch# set family ethernet-switching filter app_class term erp from source-address
192.168.1.36/28
user@switch# set family ethernet-switching filter app_class term erp protocol tcp
user@switch# set family ethernet-switching filter app_class term erp source-port [3200 3300
3301 3600]
user@switch# set family ethernet-switching filter app_class term erp then forwarding-class
erp loss-priority low
```

18. Define the term network_control (for the app_class filter):

```
[edit firewall]
user@switch# set family ethernet-switching filter app_class term network_control from
precedence [net-control internet-control]
user@switch# set family ethernet-switching filter app_class term network_control then
forwarding-class network-control loss-priority low
```

19. Define the term best_effort_traffic (for the app_class filter):

```
[edit firewall]
user@switch# set family ethernet-switching filter app_class term best_effort_traffic then
forwarding-class best-effort loss-priority low
```

20. Apply the firewall filter app_class as an input filter to the interfaces for the servers hosting
applications:

```
[edit interfaces]
user@switch# set ge-0/0/3 unit 0 family ethernet-switching filter input app_class
user@switch# set ge-0/0/4 unit 0 family ethernet-switching filter input app_class
```

```
user@switch# set ge-0/0/5 unit 0 family ethernet-switching filter input app_class
user@switch# set ge-0/0/6 unit 0 family ethernet-switching filter input app_class
```

21. Configure schedulers:

```
[edit class-of-service]
user@switch# set schedulers voice-sched shaping-rate percent 10
user@switch# set schedulers voice-sched buffer-size percent 10
user@switch# set schedulers voice-sched priority strict-high
user@switch# set schedulers video-sched priority low
user@switch# set schedulers video-sched transmit-rate percent 15
user@switch# set schedulers app-sched buffer-size percent 10
user@switch# set schedulers app-sched priority low
user@switch# set schedulers app-sched transmit-rate percent 10
user@switch# set schedulers mail-sched buffer-size percent 5
user@switch# set schedulers mail-sched priority low
user@switch# set schedulers mail-sched transmit-rate percent 5
user@switch# set schedulers db-sched buffer-size percent 10
user@switch# set schedulers db-sched priority low
user@switch# set schedulers db-sched transmit-rate percent 10
user@switch# set schedulers erp-sched buffer-size percent 10
user@switch# set schedulers erp-sched priority low
user@switch# set schedulers erp-sched transmit-rate percent 10
user@switch# set schedulers nc-sched shaping-rate percent 5
user@switch# set schedulers nc-sched buffer-size percent 5
user@switch# set schedulers nc-sched priority strict-high
user@switch# set schedulers nc-sched transmit-rate percent 5
user@switch# set schedulers be-sched buffer-size percent 35
user@switch# set schedulers be-sched priority low
user@switch# set schedulers be-sched transmit-rate percent 35
```

22. Assign the forwarding classes to schedulers with the scheduler map `ethernet-cos-map`:

```
[edit class-of-service]
user@switch# set scheduler-maps ethernet-cos-map forwarding-class voice scheduler voice-
sched
user@switch# set scheduler-maps ethernet-cos-map forwarding-class video scheduler video-
sched
user@switch# set scheduler-maps ethernet-cos-map forwarding-class app scheduler app-sched
```

```
user@switch# set scheduler-maps ethernet-cos-map forwarding-class mail scheduler mail-sched
user@switch# set scheduler-maps ethernet-cos-map forwarding-class db scheduler db-sched
user@switch# set scheduler-maps ethernet-cos-map forwarding-class erp scheduler erp-sched
user@switch# set scheduler-maps ethernet-cos-map forwarding-class network-control scheduler
nc-sched
user@switch# set scheduler-maps ethernet-cos-map forwarding-class best-effort scheduler be-
sched
```

23. Associate the scheduler map with the outgoing interfaces:

```
[edit class-of-service interfaces]
user@switch# set ge-0/0/20 scheduler-map ethernet-cos-map
user@switch# set ge-0/0/21 scheduler-map ethernet-cos-map
```

24. Apply queue shaping for the best-effort queue:

```
[edit]
user@switch# set class-of-service schedulers voice-sched-queue-shap shaping-rate 30m
user@switch# set class-of-service scheduler-maps sched-map-be forwarding-class best-effort
scheduler voice-sched-queue-shap
user@switch# set class-of-service interfaces ge-0/0/2 scheduler-map sched-map-be
```

Results

Display the results of the configuration:

```
user@switch> show firewall
```

```
firewall family ethernet-switching {
    filter voip_class {
        term voip {
            from {
                source-address {
                    192.168.1.1/28;
                    192.168.1.2/28;
                }
```

```
                protocol udp;
                source-port 2698;
            }
            then {
                forwarding-class voice;
                loss-priority low;
            }
        }
        term network control {
            from {
                precedence [net-control internet-control];
            }
            then {
                forwarding-class network-control;
                loss-priority low;
            }
        }
        term best_effort_traffic {
            then {
                forwarding-class best-effort;
                loss-priority low;
            }
        }
    }
    filter video_class {
        term video {
            from {
                source-address {
                    192.168.1.17/28;
                }
                protocol udp;
                source-port 2979;
            }
            then {
                forwarding-class video;
                loss-priority low;
            }
        }
        term network control {
            from {
                precedence [net-control internet-control];
            }
            then {
```

```
                forwarding-class network-control;
                loss-priority low;
            }
        }
        term best_effort_traffic {
            then {
                forwarding-class best-effort;
                loss-priority low;
            }
        }
    }
    filter app_class {
        term app {
            from {
                source-address {
                    192.168.1.33/28;
                }
                protocol tcp;
                source-port [1491 2512 2513 2598 2897];
            }
            then {
                forwarding-class app;
                loss-priority low;
            }
        }
        term mail {
            from {
                source-address {
                    192.168.1.34/28;
                }
                protocol tcp;
                source-port [25 143 389 691 993 3268 3269];
            }
            then {
                forwarding-class mail;
                loss-priority low;
            }
        }
        term db  {
            from {
                source-address {
                    192.168.1.35/28;
                }
```

```
                protocol tcp;
                source-port [1521 1525 1527 1571 1810 2481];
        }
        then {
            forwarding-class db;
            loss-priority low;
        }
    }
    term erp  {
        from {
            source-address {
                192.168.1.36/28;
            }
            protocol tcp;
            source-port [3200 3300 3301 3600];
        }
        then {
            forwarding-class erp;
            loss-priority low;
        }
    }
    term network control {
        from {
            precedence [net-control internet-control];
        }
        then {
            forwarding-class  network-control;
            loss-priority low;
        }
    }
    term best_effort_traffic {
        then {
            forwarding-class best-effort;
            loss-priority low;
        }
    }
```

```
        }
}
```

```
user@switch# show class-of-service
```

```
forwarding-classes {
    class app queue-num 5;
    class mail queue-num 1;
    class db queue-num 2;
    class erp queue-num 3;
    class video queue-num 4;
    class best-effort queue-num 0;
    class voice queue-num 6;
    class network-control queue-num 7;
}
interfaces {
    ge-0/0/0 {
        shaping-rate 100m;
    }
    ge-0/0/2 {
        scheduler-map sched-map-be;
    }
    ge-0/0/20 {
        scheduler-map ethernet-cos-map;
    }
    ge-0/0/21 {
        scheduler-map ethernet-cos-map;
    }
}
schedulers {
    voice-sched-queue-shap {
        shaping-rate 30m;
    }
    voice-sched {
        shaping-rate percent 10;
        buffer-size percent 10;
        priority strict-high;
    }
    video-sched  {
        buffer-size percent 15;
```

```
            priority low;
            transmit-rate percent 15;
        }
        app-sched {
            buffer-size percent 10;
            priority low;
            transmit-rate percent 10;
        }
        mail-sched {
            buffer-size percent 5;
            priority low;
            transmit-rate percent 5;
        }
        db-sched {
            buffer-size percent 10;
            priority low;
            transmit-rate percent 10;
        }
        erp-sched {
            buffer-size percent 10;
            priority low;
            transmit-rate percent 10;
        }
        nc-sched {
            shaping-rate percent 5;
            buffer-size percent 5;
            priority strict-high;
        }
        be-sched {
            buffer-size percent 35;
            priority low;
            transmit-rate percent 35;
        }
    }
    scheduler-maps {
        ethernet-cos-map {
            forwarding-class voice scheduler voice-sched;
            forwarding-class video scheduler video-sched;
            forwarding-class app scheduler app-sched;
            forwarding-class mail scheduler mail-sched;
            forwarding-class db scheduler db-sched;
            forwarding-class erp scheduler erp-sched;
            forwarding-class network-control scheduler nc-sched;
```

```
            forwarding-class best-effort scheduler be-sched;
    }
    sched-map-be {
        forwarding-class best-effort scheduler  voice-sched-queue-shap;
    }
}
```

```
user@switch# show interfaces
```

```
ge-0/0/0 {
    unit 0 {
        family ethernet {
            filter {
                input voip_class;
            }
        }
    }
}
ge-0/0/1 {
    unit 0 {
        family ethernet {
            filter {
                input voip_class;
            }
        }
    }
}
ge-0/0/2 {
    unit 0 {
        family ethernet {
            filter {
                input video_class;
            }
        }
    }
}
ge-0/0/3 {
    unit 0 {
        family ethernet {
            filter {
```

```
                    input app_class;
            }
        }
    }
}
ge-0/0/4 {
    unit 0 {
        family ethernet {
            filter {
                input app_class;
            }
        }
    }
}
ge-0/0/5 {
    unit 0 {
        family ethernet {
            filter {
                input app_class;
            }
        }
    }
}
ge-0/0/6 {
    unit 0 {
        family ethernet {
            filter {
                input app_class;
            }
        }
    }
}
```

Verification

IN THIS SECTION

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

Verifying That the Defined Forwarding Classes Exist and Are Mapped to Queues

Purpose

Verify that the forwarding classes app, best-effort, db, erp, mail, network-control, video, and voice have been defined and mapped to queues.

Action

```
user@switch> show class-of-service forwarding-class
Forwarding class                   ID    Queue
  app                               0      5
  db                                1      2
  erp                               2      3
  best-effort                       3      0
  mail                              4      1
  voice                             5      6
  video                             6      4
  network-control                   7      7
```

Meaning

This output shows that the forwarding classes have been defined and mapped to appropriate queues.

Verifying That the Forwarding Classes Have Been Assigned to Schedulers

Purpose

Verify that the forwarding classes have been assigned to schedulers.

Action

```
user@switch> show class-of-service scheduler-map
Scheduler map: ethernet-cos-map, Index: 2
  Scheduler: voice-sched, Forwarding class: voice, Index: 22
    Shaping rate: 10 percent, Rate Limit: none, Buffer size: 10 percent,
    Priority: Strict-high
    Drop profiles:
      Loss priority   Protocol    Index   Name
      High            non-TCP       1     <default-drop-profile>
      High            TCP           1     <default-drop-profile>

  Scheduler: video-sched, Forwarding class: video, Index: 22
    Transmit rate: 10 percent, Rate Limit: none, Buffer size: 10 percent,
    Priority: low
    Drop profiles:
      Loss priority   Protocol    Index   Name
      High            non-TCP       1     <default-drop-profile>
      High            TCP           1     <default-drop-profile>

  Scheduler: app-sched, Forwarding class: app, Index: 22
    Transmit rate: 10 percent, Rate Limit: none, Buffer size: 10 percent,
    Priority: low
    Drop profiles:
      Loss priority   Protocol    Index   Name
      High            non-TCP       1     <default-drop-profile>
      High            TCP           1     <default-drop-profile>

  Scheduler: mail-sched, Forwarding class: mail, Index: 22
    Transmit rate: 5 percent, Rate Limit: none, Buffer size: 5 percent,
    Priority: low
    Drop profiles:
      Loss priority   Protocol    Index   Name
      High            non-TCP       1     <default-drop-profile>
      High            TCP           1     <default-drop-profile>

  Scheduler: db-sched, Forwarding class: db, Index: 22
    Transmit rate: 10 percent, Rate Limit: none, Buffer size: 10 percent,
    Priority: low
    Drop profiles:
      Loss priority   Protocol    Index   Name
      High            non-TCP       1     <default-drop-profile>
```

```
      High            TCP               1    <default-drop-profile>


  Scheduler: erp-sched, Forwarding class: erp, Index: 22
    Transmit rate: 10 percent, Rate Limit: none, Buffer size: 10 percent,
    Priority: low
    Drop profiles:
      Loss priority   Protocol     Index    Name
      High            non-TCP           1    <default-drop-profile>
      High            TCP               1    <default-drop-profile>


  Scheduler: be-sched, Forwarding class: best-effort, Index: 20
    Transmit rate: 35 percent, Rate Limit: none, Buffer size: 35 percent,
    Priority: low
    Drop profiles:
      Loss priority   Protocol     Index    Name
      High            non-TCP           1    <default-drop-profile>
      High            TCP               1    <default-drop-profile>


  Scheduler: nc-sched, Forwarding class: network-control, Index: 22
    Shaping rate: 5 percent, Rate Limit: none, Buffer size: 5 percent,
    Priority: Strict-high
    Drop profiles:
      Loss priority   Protocol     Index    Name
      High            non-TCP           1    <default-drop-profile>
      High            TCP               1    <default-drop-profile>
```

Meaning

This output shows that the forwarding classes have been assigned to schedulers.

Verifying That the Scheduler Map Has Been Applied to the Interfaces

Purpose

Verify that the scheduler map has been applied to the interfaces.

Action

```
user@switch> show class-of-service interface
...
Physical interface: ge-0/0/20, Index: 149
Queues supported: 8, Queues in use: 8
  Scheduler map: ethernet-cos-map, Index: 43366
  Input scheduler map: <default>, Index: 3
...
Physical interface: ge-0/0/21, Index: 150
Queues supported: 8, Queues in use: 8
  Scheduler map: ethernet-cos-map, Index: 15103
  Input scheduler map: <default>, Index: 5
...
```

Meaning

This output includes details of the interfaces to which the scheduler map (`ethernet-cos-map`) has been applied (`ge-0/0/20` and `ge-0/0/21`).

Verifying That Port Shaping Has Been Applied

Purpose

Verify that the port shaping has been applied to an interface.

Action

Following is the output before port shaping is applied to the interface `ge-0/0/0`, when there is egress traffic of 400 Mpbs exiting on that interface:

```
user@switch> show interfaces ge-0/0/0 extensive
Physical interface: ge-0/0/0, Enabled, Physical link is Up
  Interface index: 239, SNMP ifIndex: 548, Generation: 242
  Link-level type: Ethernet, MTU: 1514, Speed: Auto, Duplex: Auto, BPDU Error: None, MAC-REWRITE
Error: None, Loopback: Disabled, Source filtering: Disabled, Flow control: Enabled, Auto-
negotiation: Enabled, Remote fault: Online,
  Media type: Copper
  Device flags   : Present Running
  Interface flags: SNMP-Traps Internal: 0x0
```

```
  Link flags     : None
  CoS queues     : 8 supported, 8 maximum usable queues
  Hold-times     : Up 0 ms, Down 0 ms
  Current address: 00:23:9c:0b:ae:8d, Hardware address: 00:23:9c:0b:ae:8d
  Last flapped   : 2012-07-07 03:21:52 UTC (1d 18:02 ago)
  Statistics last cleared: 2012-07-07 23:54:34 UTC (21:29:59 ago)
  Traffic statistics:
   Input  bytes  :                    0                    0 bps
   Output bytes  :           2299853696            345934816 bps
   Input  packets:                    0                    0 pps
   Output packets:             17967609               337827 pps
    IPv6 transit statistics:
    Input  bytes  :                0
    Output bytes  :                0
    Input  packets:                0
    Output packets:                0
  Input errors:
    Errors: 0, Drops: 0, Framing errors: 0, Runts: 0, Policed discards: 0, L3 incompletes: 0, L2
channel errors: 0, L2 mismatch timeouts: 0, FIFO errors: 0, Resource errors: 0
  Output errors:
    Carrier transitions: 0, Errors: 0, Drops: 0, Collisions: 0, Aged packets: 0, FIFO errors: 0,
HS link CRC errors: 0, MTU errors: 0, Resource errors: 0
  Egress queues: 8 supported, 4 in use
  Queue counters:        Queued packets  Transmitted packets    Dropped packets
    0 best-effort                     0             18302337                  0
    1 assured-forw                    0                    0                  0
    5 expedited-fo                    0                    0                  0
    7 network-cont                    0                    0                  0
  Queue number:        Mapped forwarding classes
    0                  best-effort
    1                  assured-forwarding
    5                  expedited-forwarding
    7                  network-control
  Active alarms  : None
  Active defects : None
  MAC statistics:                      Receive        Transmit
    Total octets                             0      2299853696
    Total packets                            0        17967609
    Unicast packets                          0        17967609
    Broadcast packets                        0               0
    Multicast packets                        0               0
    CRC/Align errors                         0               0
    FIFO errors                              0               0
```

```
      MAC control frames                       0                   0
      MAC pause frames                         0                   0
      Oversized frames                         0
      Jabber frames                            0
      Fragment frames                          0
      Code violations                          0
  Autonegotiation information:
    Negotiation status: Complete
    Link partner:
        Link mode: Full-duplex, Flow control: Symmetric, Remote fault: OK, Link partner Speed:
1000 Mbps
    Local resolution:
        Flow control: Symmetric, Remote fault: Link OK
  Packet Forwarding Engine configuration:
    Destination slot: 1
  CoS information:
    Direction : Output
    CoS transmit queue           Bandwidth             Buffer Priority   Limit
                          %          bps     %           usec
      0 best-effort       95     950000000   95            NA      low     none
      7 network-control    5      50000000    5            NA      low     none
  Interface transmit statistics: Disabled


  Logical interface ge-1/0/10.0 (Index 69) (SNMP ifIndex 638) (Generation 138)
    Flags: SNMP-Traps 0x0 Encapsulation: ENET2
    Traffic statistics:
     Input  bytes  :                0
     Output bytes  :                0
     Input  packets:                0
     Output packets:                0
    Local statistics:
     Input  bytes  :                0
     Output bytes  :                0
     Input  packets:                0
     Output packets:                0
    Transit statistics:
     Input  bytes  :                0                    0 bps
     Output bytes  :                0                    0 bps
     Input  packets:                0                    0 pps
     Output packets:                0                    0 pps
    Protocol eth-switch, Generation: 163, Route table: 0
      Flags: Trunk-Mode
```

The Traffic statistics: field in this output shows that egress traffic is ~400 Mpbs (345,934,816 bps). When a port shaping of 100 Mbps is applied to the ge-0/0/0 interface, you see the following outputs for the **show interfaces ge-0/0/0 statistics** and the **show class-of-service interface ge-0/0/0** commands:

```
user@switch> show interfaces ge-0/0/0 statistics
Physical interface: ge-0/0/0, Enabled, Physical link is Up
  Interface index: 239, SNMP ifIndex: 548, Generation: 242
  Link-level type: Ethernet, MTU: 1514, Speed: Auto, Duplex: Auto, BPDU Error: None, MAC-REWRITE
Error: None, Loopback: Disabled, Source filtering: Disabled, Flow control: Enabled, Auto-
negotiation: Enabled, Remote fault: Online,
  Media type: Copper
  Device flags   : Present Running
  Interface flags: SNMP-Traps Internal: 0x0
  Link flags     : None
  CoS queues     : 8 supported, 8 maximum usable queues
  Hold-times     : Up 0 ms, Down 0 ms
  Current address: 00:23:9c:0b:ae:8d, Hardware address: 00:23:9c:0b:ae:8d
  Last flapped   : 2012-07-07 03:21:52 UTC (1d 18:10 ago)
  Statistics last cleared: 2012-07-07 23:54:34 UTC (21:37:58 ago)
  Traffic statistics:
   Input  bytes  :                    0                      0 bps
   Output bytes  :          15779512832              100223104 bps
   Input  packets:                    0                      0 pps
   Output packets:            123277444                  97874 pps
    IPv6 transit statistics:
     Input  bytes  :               0
     Output bytes  :               0
     Input  packets:               0
     Output packets:               0
   Input errors:
     Errors: 0, Drops: 0, Framing errors: 0, Runts: 0, Policed discards: 0, L3 incompletes: 0, L2
channel errors: 0, L2 mismatch timeouts: 0, FIFO errors: 0, Resource errors: 0
   Output errors:
     Carrier transitions: 0, Errors: 0, Drops: 0, Collisions: 0, Aged packets: 0, FIFO errors: 0,
HS link CRC errors: 0, MTU errors: 0, Resource errors: 0
   Egress queues: 8 supported, 4 in use
   Queue counters:       Queued packets  Transmitted packets    Dropped packets
     0 best-effort                   0            123350092           57012484
     1 assured-forw                  0                    0                  0
     5 expedited-fo                  0                    0                  0
     7 network-cont                  0                    0                  0
```

```
   Queue number:        Mapped forwarding classes
     0                  best-effort
     1                  assured-forwarding
     5                  expedited-forwarding
     7                  network-control
   Active alarms  : None
   Active defects : None
   MAC statistics:                    Receive         Transmit
     Total octets                         0       15779512832
     Total packets                        0         123277444
     Unicast packets                      0         123277444
     Broadcast packets                    0                 0
     Multicast packets                    0                 0
     CRC/Align errors                     0                 0
     FIFO errors                          0                 0
     MAC control frames                   0                 0
     MAC pause frames                     0                 0
     Oversized frames                     0
     Jabber frames                        0
     Fragment frames                      0
     Code violations                      0
   Autonegotiation information:
     Negotiation status: Complete
     Link partner:
        Link mode: Full-duplex, Flow control: Symmetric, Remote fault: OK, Link partner Speed:
1000 Mbps
     Local resolution:
        Flow control: Symmetric, Remote fault: Link OK
   Packet Forwarding Engine configuration:
     Destination slot: 1
   CoS information:
     Direction : Output
     CoS transmit queue            Bandwidth            Buffer Priority   Limit
                            %          bps     %           usec
     0 best-effort         95     95000000    95             NA    low    none
     7 network-control      5      5000000     5             NA    low    none
   Interface transmit statistics: Disabled


   Logical interface ge-1/0/10.0 (Index 69) (SNMP ifIndex 638) (Generation 138)
     Flags: SNMP-Traps 0x0 Encapsulation: ENET2
     Traffic statistics:
      Input  bytes  :                   0
      Output bytes  :                   0
```

```
   Input  packets:                0
   Output packets:                0
  Local statistics:
  Input  bytes  :                 0
  Output bytes  :                 0
  Input  packets:                 0
  Output packets:                 0
  Transit statistics:
  Input  bytes  :                 0                    0 bps
  Output bytes  :                 0                    0 bps
  Input  packets:                 0                    0 pps
  Output packets:                 0                    0 pps
   Protocol eth-switch, Generation: 163, Route table: 0
    Flags: Trunk-Mode


user@switch> show class-of-service interface ge-0/0/0
Physical interface: ge-0/0/0, Index: 165
Queues supported: 8, Queues in use: 4
  Shaping rate: 100000000 bps
  ...
  ...
```

Meaning

In the output for the `show interfaces ge-0/0/0 statistics` command, the Traffic statistics: field shows that egress traffic is ~100 Mbps (100,223,104 bps). The output for the `show class-of-service interface ge-0/0/0` command shows that the shaping rate is 100,000,000 bps, which indicates that a port shaping of 100 Mbps is applied to the ge-0/0/0 interface.

Verifying That Queue Shaping Has Been Applied

Purpose

Verify that the queue shaping has been applied to the best-effort queue.

Action

Following is the output before queue shaping is applied to the best-effort queue when there is egress traffic of 400 Mpbs exiting on that interface:

```
user@switch> show interfaces ge-0/0/2 extensive
Physical interface: ge-0/0/2, Enabled, Physical link is Up
  Interface index: 239, SNMP ifIndex: 548, Generation: 242
  Link-level type: Ethernet, MTU: 1514, Speed: Auto, Duplex: Auto, BPDU Error: None, MAC-REWRITE
Error: None, Loopback: Disabled, Source filtering: Disabled, Flow control: Enabled, Auto-
negotiation: Enabled, Remote fault: Online,
  Media type: Copper
  Device flags   : Present Running
  Interface flags: SNMP-Traps Internal: 0x0
  Link flags     : None
  CoS queues     : 8 supported, 8 maximum usable queues
  Hold-times     : Up 0 ms, Down 0 ms
  Current address: 00:23:9c:0b:ae:8d, Hardware address: 00:23:9c:0b:ae:8d
  Last flapped   : 2012-07-07 03:21:52 UTC (1d 18:02 ago)
  Statistics last cleared: 2012-07-07 23:54:34 UTC (21:29:59 ago)
  Traffic statistics:
   Input  bytes  :                    0                    0 bps
   Output bytes  :           2299853696            345934816 bps
   Input  packets:                    0                    0 pps
   Output packets:             17967609               337827 pps
   IPv6 transit statistics:
    Input  bytes  :                   0
    Output bytes  :                   0
    Input  packets:                   0
    Output packets:                   0
  Input errors:
    Errors: 0, Drops: 0, Framing errors: 0, Runts: 0, Policed discards: 0, L3 incompletes: 0, L2
channel errors: 0, L2 mismatch timeouts: 0, FIFO errors: 0, Resource errors: 0
  Output errors:
    Carrier transitions: 0, Errors: 0, Drops: 0, Collisions: 0, Aged packets: 0, FIFO errors: 0,
HS link CRC errors: 0, MTU errors: 0, Resource errors: 0
  Egress queues: 8 supported, 4 in use
  Queue counters:       Queued packets  Transmitted packets   Dropped packets
    0 best-effort                    0             18302337                 0
    1 assured-forw                   0                    0                 0
    5 expedited-fo                   0                    0                 0
    7 network-cont                   0                    0                 0
```

```
   Queue number:        Mapped forwarding classes
     0                  best-effort
     1                  assured-forwarding
     5                  expedited-forwarding
     7                  network-control
  Active alarms  : None
  Active defects : None
  MAC statistics:                      Receive        Transmit
    Total octets                           0      2299853696
    Total packets                          0        17967609
    Unicast packets                        0        17967609
    Broadcast packets                      0               0
    Multicast packets                      0               0
    CRC/Align errors                       0               0
    FIFO errors                            0               0
    MAC control frames                     0               0
    MAC pause frames                       0               0
    Oversized frames                       0
    Jabber frames                          0
    Fragment frames                        0
    Code violations                        0
  Autonegotiation information:
    Negotiation status: Complete
    Link partner:
        Link mode: Full-duplex, Flow control: Symmetric, Remote fault: OK, Link partner Speed:
1000 Mbps
    Local resolution:
        Flow control: Symmetric, Remote fault: Link OK
  Packet Forwarding Engine configuration:
    Destination slot: 1
  CoS information:
    Direction : Output
    CoS transmit queue              Bandwidth              Buffer Priority   Limit
                          %          bps       %             usec
    0 best-effort        95     950000000     95              NA     low    none
    7 network-control     5      50000000      5              NA     low    none
  Interface transmit statistics: Disabled


  Logical interface ge-1/0/10.0 (Index 69) (SNMP ifIndex 638) (Generation 138)
    Flags: SNMP-Traps 0x0 Encapsulation: ENET2
    Traffic statistics:
     Input  bytes  :                   0
     Output bytes  :                   0
```

```
   Input  packets:                    0
   Output packets:                    0
  Local statistics:
   Input  bytes  :                    0
   Output bytes  :                    0
   Input  packets:                    0
   Output packets:                    0
  Transit statistics:
   Input  bytes  :                    0                    0 bps
   Output bytes  :                    0                    0 bps
   Input  packets:                    0                    0 pps
   Output packets:                    0                    0 pps
  Protocol eth-switch, Generation: 163, Route table: 0
    Flags: Trunk-Mode
```

The Traffic statistics: field in this output shows that the egress traffic is ~400 Mpbs (345,934,816 bps). When a queue shaping of 30 Mbps is applied to the best-effort queue, you see the following output for the show interfaces ge-0/0/2 statistics and show class-of-service scheduler-map sched-map-be commands:

```
user@switch> show interfaces ge-0/0/2 statistics
Physical interface: ge-0/0/2, Enabled, Physical link is Up
  Interface index: 239, SNMP ifIndex: 548, Generation: 242
  Link-level type: Ethernet, MTU: 1514, Speed: Auto, Duplex: Auto, BPDU Error: None, MAC-REWRITE
Error: None, Loopback: Disabled, Source filtering: Disabled, Flow control: Enabled, Auto-
negotiation: Enabled, Remote fault: Online,
  Media type: Copper
  Device flags   : Present Running
  Interface flags: SNMP-Traps Internal: 0x0
  Link flags     : None
  CoS queues     : 8 supported, 8 maximum usable queues
  Hold-times     : Up 0 ms, Down 0 ms
  Current address: 00:23:9c:0b:ae:8d, Hardware address: 00:23:9c:0b:ae:8d
  Last flapped   : 2012-07-07 03:21:52 UTC (1d 18:29 ago)
  Statistics last cleared: 2012-07-08 21:46:22 UTC (00:04:56 ago)
  Traffic statistics:
   Input  bytes  :                    0                    0 bps
   Output bytes  :           5376128896             30097712 bps
   Input  packets:                    0                    0 pps
   Output packets:             42001003                29392 pps
   IPv6 transit statistics:
    Input  bytes  :                   0
    Output bytes  :                   0
```

```
   Input  packets:                    0
   Output packets:                    0
 Input errors:
   Errors: 0, Drops: 0, Framing errors: 0, Runts: 0, Policed discards: 0, L3 incompletes: 0, L2
channel errors: 0, L2 mismatch timeouts: 0, FIFO errors: 0, Resource errors: 0
 Output errors:
   Carrier transitions: 0, Errors: 0, Drops: 0, Collisions: 0, Aged packets: 0, FIFO errors: 0,
HS link CRC errors: 0, MTU errors: 0, Resource errors: 0
 Egress queues: 8 supported, 4 in use
 Queue counters:        Queued packets  Transmitted packets     Dropped packets
   0 best-effort                 0             41986978             57813642
   1 assured-forw                0                    0                    0
   5 expedited-fo                0                    0                    0
   7 network-cont                0                    0                    0
 Queue number:        Mapped forwarding classes
   0                  best-effort
   1                  assured-forwarding
   5                  expedited-forwarding
   7                  network-control
 Active alarms  : None
 Active defects : None
 MAC statistics:                 Receive        Transmit
   Total octets                       0      5376128896
   Total packets                      0        42001003
   Unicast packets                    0        42001003
   Broadcast packets                  0               0
   Multicast packets                  0               0
   CRC/Align errors                   0               0
   FIFO errors                        0               0
   MAC control frames                 0               0
   MAC pause frames                   0               0
   Oversized frames                   0
   Jabber frames                      0
   Fragment frames                    0
   Code violations                    0
 Autonegotiation information:
   Negotiation status: Complete
   Link partner:
       Link mode: Full-duplex, Flow control: Symmetric, Remote fault: OK, Link partner Speed:
1000 Mbps
   Local resolution:
       Flow control: Symmetric, Remote fault: Link OK
 Packet Forwarding Engine configuration:
```

```
    Destination slot: 1
  CoS information:
    Direction : Output
    CoS transmit queue              Bandwidth              Buffer Priority   Limit
                            %              bps    %            usec
    0 best-effort           r              r      r            NA      low    none
  Interface transmit statistics: Disabled


  Logical interface ge-1/0/10.0 (Index 69) (SNMP ifIndex 638) (Generation 138)
    Flags: SNMP-Traps 0x0 Encapsulation: ENET2
    Traffic statistics:
     Input  bytes  :                 0
     Output bytes  :                 0
     Input  packets:                 0
     Output packets:                 0
    Local statistics:
     Input  bytes  :                 0
     Output bytes  :                 0
     Input  packets:                 0
     Output packets:                 0
    Transit statistics:
     Input  bytes  :                 0                  0 bps
     Output bytes  :                 0                  0 bps
     Input  packets:                 0                  0 pps
     Output packets:                 0                  0 pps
    Protocol eth-switch, Generation: 163, Route table: 0
      Flags: Trunk-Mode
```

```
user@switch> show class-of-service scheduler-map sched-map-be

Scheduler map: sched-map-be, Index: 31271

  Scheduler: voice-sched-queue-shap, Forwarding class: best-effort, Index: 64106
    Transmit rate: remainder, Rate Limit: none, Buffer size: remainder,
    Buffer Limit: none, Priority: low
    Excess Priority: unspecified
    Shaping rate: 30000000 bps
    Drop profiles:
      Loss priority   Protocol    Index    Name
```

```
     High              non-TCP         1    <default-drop-profile>
     High              TCP             1    <default-drop-profile>
```

Meaning

In the output for the `show interfaces ge-0/0/2 statistics` command, the Traffic statistics: field shows that the egress traffic is ~30 Mbps (30,097,712 bps). The output for the `show class-of-service scheduler-map sched-map-be` command, shows that a shaping rate of 30,000,000 bps (that is 30 Mbps) is applied to the best-effort queue.

RELATED DOCUMENTATION

Configuring Firewall Filters (CLI Procedure)

CHAPTER 2

Junos OS EZQoS and J-Web

IN THIS CHAPTER

Understanding Junos OS EZQoS for CoS Configurations on EX Series Switches

Junos operating system (Junos OS) EZQoS on Juniper Networks EX Series Ethernet Switches eliminates the complexities involved in configuring *class of service* (CoS) across the network. EZQoS offers templates for key traffic classes.

Junos OS CoS allows you to divide traffic into classes and offer various levels of throughput and packet loss when congestion occurs. You can use CoS to ensure that different types of traffic (voice, video, and data) get the bandwidth and consideration they need to meet user expectations and business objectives.

Configuring CoS requires careful consideration of your service needs and thorough planning and design to ensure consistency across all switches in a CoS domain. To configure CoS manually, you must define and fine-tune all CoS components such as classifiers, *rewrite rules*, forwarding classes, schedulers, and scheduler-maps and then apply these components to the interfaces. Therefore, configuring CoS can be a fairly complex and time-consuming task.

EZQoS works by automatically assigning preconfigured values to all CoS parameters based on the typical application requirements. These preconfigured values are stored in a template with a unique name. You can change the preconfigured values of these parameters to suit your particular application needs.

For using EZQoS, you must identify which switch ports are being used for a specific application (such as VoIP, video, and data) and manually apply the corresponding application-specific EZQoS template to these switch ports.

> **(i)** **NOTE**: Currently, we provide an EZQoS template for configuring CoS for VoIP.

> **(i)** **NOTE**: We recommend that you do not use the term **EZQoS** for defining a classifier.

RELATED DOCUMENTATION

Configuring Junos OS EZQoS for CoS (CLI Procedure)

You use Junos OS EZQoS on EX Series switches to eliminate the complexities involved in configuring class of service (CoS) across the network. EZQoS offers templates for key traffic classes.

When you configure EZQoS on EX Series switches, preconfigured values are assigned to all CoS parameters based on the typical application requirements. These preconfigured values are stored in a template with a unique name.

> **(i)** **NOTE**: Currently, we provide an EZQoS template for configuring CoS for VoIP applications. The EZQoS VoIP template is stored in **/etc/config/ezqos-voip.conf**.

To configure EZQoS using the CLI:

1. Load the EZQoS configuration file (**/etc/config/ezqos-voip.conf**):

   ```
   [edit]
   user@switch# load merge /etc/config/ezqos-voip.conf
   ```

2. Apply the EZQoS group (**ezqos-voip**):

   ```
   [edit]
   user@switch# set apply-groups ezqos-voip
   ```

3. Apply the DSCP classifier (**ezqos-dscp-classifier**) to a Gigabit Ethernet interface (**ge-0/0/0**):

```
[edit class-of-service interfaces]
user@switch# set ge-0/0/0 unit 0 classifiers dscp ezqos-dscp-classifier
```

4. Apply the scheduler map (**ezqos-voip-sched-maps**) to a Gigabit Ethernet interface (**ge-0/0/1**):

```
[edit class-of-service interfaces]
user@switch# set ge-0/0/1 scheduler-map ezqos-voip-sched-maps
```

RELATED DOCUMENTATION

Configuring CoS (J-Web Procedure)

The Class of Service Configuration pages allow you to configure the Junos CoS components. You can configure forwarding classes for transmitting packets, define which packets are placed into each output queue, and schedule the transmission service level for each queue. After defining the CoS components you must assign classifiers to the required physical and logical interfaces.

Using the Class of Service Configuration pages, you can configure various CoS components individually or in combination to define particular CoS services.

To configure CoS components :

1. In the J-Web interface, select **Configure>Class of Service**.
2. On the Class of Service Configuration page, select one of the following options depending on the CoS component that you want to define. Enter information into the pages as described in the respective table:

   - To define or edit CoS value aliases, select **CoS Value Aliases** .

   - To define or edit forwarding classes and assign queues, select **Forwarding Classes**.

   - To define or edit classifiers, select **Classifiers** .

   - To define or edit rewrite rules, select **Rewrite Rules**.

- To define or edit schedulers, select **Schedulers**.

- To define or edit virtual channel groups, select **Interface Associations**.

3. Click **Apply** after completing configuration on any Configuration page.

RELATED DOCUMENTATION

CoS on Interfaces

IN THIS CHAPTER

Assigning CoS Components to Interfaces (CLI Procedure)

After you have defined the following CoS components, you must assign them to logical or physical interfaces.

- Forwarding classes—Assign only to logical interfaces.

- Classifiers—Assign only to logical interfaces.

- Scheduler maps—Assign to either physical or logical interfaces.

- Rewrite rules—Assign to either physical or logical interfaces.

You can assign a CoS component to a single interface or to multiple interfaces using wild cards.

To assign CoS components to interfaces:

- To assign CoS components to a single interface, associate a CoS component (for example a scheduler map named `ethernet-cos-map`) with an interface:

```
[edit class-of-service interfaces]
user@switch# set ge-0/0/20 scheduler-map ethernet-cos-map
```

- To assign a CoS component to multiple interfaces, associate a CoS component (for example, a rewrite rule named `customup-rw`) to all Gigabit Ethernet interfaces on the switch, use wild characters for the interface name and logical-interface (unit) number:

```
[edit class-of-service interfaces]
user@switch# set ge-* unit * rewrite-rules ieee-802.1 customup-rw
```

RELATED DOCUMENTATION

Assigning CoS Components to Interfaces (J-Web Procedure)

> **(i)** **NOTE**: This topic applies only to the J-Web Application package.

After you have defined CoS components on an EX Series switch, you must assign them to logical or physical interfaces. You can use the J-Web interface to assign scheduler maps to physical or logical interfaces and to assign forwarding classes or classifiers to logical interfaces.

To assign CoS components to interfaces:

1. Select **Configure** > **Class of Service** > **Assign to Interface**.

   > **(i)** **NOTE**: After you make changes to the configuration on this page, you must commit the changes immediately for them to take effect. To commit all changes to the active configuration, select **Commit Options** > **Commit**. See Using the Commit Options to Commit Configuration Changes (J-Web Procedure) for details about all commit options.

2. To configure an interface association, select an interface from the list and click **Edit**.

3. Select one of the following:

   - **Associate system default scheduler map**—Associates the interface with the default scheduler map.

- **Select the scheduler map**—Associates the interface with a configured scheduler map. Select the scheduler map from the list.

4. Click **OK**.

5. To manage a CoS assignment on a logical interface, Click one of the following options:

   - **Add**—Adds a CoS service to a logical interface on a specified physical interface. Enter information as described in Table 4 on page 54.

   - **Edit**—Modifies a CoS service assignment to a logical interface. Enter information as described in Table 4 on page 54.

   - **Delete**—Deletes the CoS service assignment to a logical interface.

Table 4: Assigning CoS Components to Logical Interfaces

Field	Function	Your Action
Unit	Specifies the name of a logical interface. Enables you to assign CoS components when you configure a logical interface on a physical interface.	Type the interface name. To assign CoS to all logical interfaces configured on this physical interface, type the wildcard character (*).
Forwarding Class	Assigns a predefined forwarding class to incoming packets on a logical interface.	To assign a forwarding class to an interface, select the forwarding class.
Classifiers	Enables you to apply classification maps to a logical interface. Classifiers assign a forwarding class and loss priority to an incoming packet based on its CoS value.	To assign a classification map to an interface, select an appropriate classifier for each CoS value type used on the interface.
Rewrite Rules	Enables you to alter the CoS values in outgoing packets to meet the requirements of the targeted peer. A rewrite rule examines the forwarding class and loss priority of a packet and sets its bits to a corresponding value specified in the rule.	To assign rewrite rules to the interface, select the appropriate rewrite rule for each CoS value type used on the interface.

RELATED DOCUMENTATION

Monitoring Interfaces That Have CoS Components

IN THIS SECTION

Purpose

> **NOTE**: This topic applies only to the J-Web Application package.

Use the monitoring functionality to display details about the physical and logical interfaces and the CoS components assigned to them.

Action

To monitor interfaces that have CoS components in the J-Web interface, select **Monitor** > **Class of Service** > **Interface Association**.

To monitor interfaces that have CoS components in the CLI, enter the following command:

```
show class-of-service interface interface
```

Meaning

summarizes key output fields for CoS interfaces.

Table 5: Summary of Key CoS Interfaces Output Fields

Field	Values	Additional Information
Interface	Name of a physical interface to which CoS components are assigned.	To display names of logical interfaces configured on this physical interface, click the plus sign (+).
Scheduler Map	Name of the scheduler map associated with this interface.	
Queues Supported	Number of queues you can configure on the interface.	
Queues in Use	Number of queues currently configured.	
Logical Interface	Name of a logical interface on the physical interface to which CoS components are assigned.	
Object	Category of an object—for example, **classifier**, **scheduler-map**, or **rewrite**.	
Name	Name that you have given to an object —for example, **ba-classifier**.	
Type	Type of an object—for example, **dscp** for a classifier.	
Index	Index of this interface or the internal index of a specific object.	

RELATED DOCUMENTATION

Assigning CoS Components to Interfaces (CLI Procedure) | **52**

Assigning CoS Components to Interfaces (J-Web Procedure) | **53**

Example: Configuring CoS on EX Series Switches | 11

CHAPTER 4

CoS Code-Point Aliases

IN THIS CHAPTER

Understanding CoS Code-Point Aliases

IN THIS SECTION

A code-point alias assigns a name to a pattern of code-point bits. You can use this name instead of the bit pattern when you configure other CoS components such as classifiers, drop-profile maps, and *rewrite rules*.

> **(i)** **NOTE**: This topic applies to all EX Series switches except the EX4600. Because the EX4600 uses a different chipset than other EX Series switches, the code-point aliases on EX4600 match those on QFX Series switches. For EX4600 code-point aliases, see *Understanding CoS Code-Point Aliases*.

Behavior aggregate classifiers use class-of-service (CoS) values such as Differentiated Services code points (DSCPs), IP precedence, and IEEE 802.1p bits to associate incoming packets with a particular CoS servicing level. On a switch, you can assign a meaningful name or alias to the CoS values and use this alias instead of bits when configuring CoS components. These aliases are not part of the specifications but are well known through usage. For example, the alias for DSCP 101110 is widely accepted as ef (expedited forwarding).

When you configure classes and define classifiers, you can refer to the markers by alias names. You can configure user-defined classifiers in terms of alias names. If the value of an alias changes, it alters the behavior of any classifier that references it.

This topic covers:

Default Code-Point Aliases

shows the default mappings between the bit values and standard aliases.

Table 6: Default Code-Point Aliases

CoS Value Types	Mapping
DSCP CoS Values	
ef	101110
af11	001010
af12	001100
af13	001110
af21	010010
af22	010100
af23	010110
af31	011010
af32	011100
af33	011110
af41	100010

Table 6: Default Code-Point Aliases *(Continued)*

CoS Value Types	Mapping
af42	100100
af43	100110
be	000000
cs1	001000
cs2	010000
cs3	011000
cs4	100000
cs5	101000
nc1/cs6	110000
nc2/cs7	111000

IEEE 802.1p CoS Values

be	000
be1	001
ef	100
ef1	101

Table 6: Default Code-Point Aliases *(Continued)*

CoS Value Types	Mapping
af11	010
af12	011
nc1/cs6	110
nc2/cs7	111
Legacy IP Precedence CoS Values	
be	000
be1	001
ef	010
ef1	011
af11	100
af12	101
nc1/cs6	110
nc2/cs7	111

RELATED DOCUMENTATION

Defining CoS Code-Point Aliases (CLI Procedure)

You can use code-point aliases to streamline the process of configuring CoS features on your EX Series switch. A code-point alias assigns a name to a pattern of code-point bits. You can use this name instead of the bit pattern when you configure other CoS components such as classifiers, drop-profile maps, and rewrite rules.

You can configure code-point aliases for the following CoS marker types:

- **dscp** and **dscp-ipv6**—Handles incoming IPv4 and IPv6 packets, respectively.

- **ieee-802.1**—Handles Layer 2 CoS.

- **inet-precedence**—Handles incoming IPv4 packets. IP precedence mapping requires only the higher order three bits of the DSCP field.

To configure a code-point alias for a specified CoS marker type (**dscp**), assign an alias (**my1**) to the code-point (**110001**):

```
  [edit class-of-service code-point-aliases]
 user@switch# set dscp my1 110001
```

The **my1** alias will be applicable for incoming IPv4 packets.

RELATED DOCUMENTATION

Defining CoS Code-Point Aliases (J-Web Procedure)

> **NOTE**: This topic applies only to the J-Web Application package.

You can use the J-Web interface to define CoS code-point aliases on an EX Series switch. By defining aliases, you can assign meaningful names to a particular set of bit values and refer to them when configuring CoS components.

To define CoS code-point aliases:

1. Select **Configure** > **Class of Service** > **CoS Value Aliases**.

   > **NOTE**: After you make changes to the configuration on this page, you must commit the changes immediately for them to take effect. To commit all changes to the active configuration, select **Commit Options** > **Commit**. See Using the Commit Options to Commit Configuration Changes (J-Web Procedure) for details about all commit options.

2. Click one of the following options:

   - **Add**—Adds a code-point alias. Enter information into the code point alias page as described in Table 7 on page 63.

   - **Edit**—Modifies an existing code-point alias. Enter information into the code point alias page as described in Table 7 on page 63.

   - **Delete**—Deletes an existing code-point alias.

Table 7 on page 63 describes the related fields.

Table 7: CoS Value Aliases Configuration Fields

Field	Function	Your Action
Code point name	Specifies the name for a code-point—for example, **af11** or **be**.	Enter a name.
Code point type	Specifies a code-point type. The code-point type can be DSCP or IP precedence.	Select a value.

Table 7: CoS Value Aliases Configuration Fields *(Continued)*

Field	Function	Your Action
Code point value bits	Specifies the CoS value for which an alias is defined. Changing this value alters the behavior of all classifiers that refer to this alias.	To specify a CoS value, type it in the appropriate format: • For DSCP CoS values, use the format xxxxxx, where x is 1 or 0—for example, **101110**. • For IP precedence CoS values, use the format xxx, where x is 1 or 0—for example, **111**.

RELATED DOCUMENTATION

Monitoring CoS Value Aliases

IN THIS SECTION

Purpose

> **NOTE**: This topic applies only to the J-Web Application package.

Use the monitoring functionality to display information about the CoS value aliases that the system is currently using to represent DSCP, IEEE 802.1p, and IPv4 precedence bits.

Action

To monitor CoS value aliases in the J-Web interface, select **Monitor** > **Class of Service** > **CoS Value Aliases**.

To monitor CoS value aliases in the CLI, enter the following command:

```
show class-of-service code-point-aliases
```

Meaning

Table 8 on page 65 summarizes key output fields for CoS value aliases.

Table 8: Summary of Key CoS Value Alias Output Fields

Field	Values	Additional Information
CoS Value Type	Type of the CoS value: • **dscp**—Examines Layer 3 packet headers for IP packet classification. • **ieee-802.1**—Examines Layer 2 packet headers for packet classification. • **inet-precedence**—Examines Layer 3 packet headers for IP packet classification.	To display aliases and bit patterns, click the plus sign (+).
CoS Value Alias	Name given to a set of bits—for example, **af11** is a name for **001010** bits.	
CoS Value	Set of bits associated with an alias.	

RELATED DOCUMENTATION

CoS Classifiers

IN THIS CHAPTER

Understanding CoS Classifiers

IN THIS SECTION

Packet classification associates incoming packets with a particular class-of-service (CoS) servicing level. Classifiers associate packets with a forwarding class and loss priority, and packets are associated to an output queue based on the forwarding class. You can define classifiers for the following interfaces:

- IPv4 and IPv6 traffic to network interfaces, aggregated Ethernet interfaces (also known as link aggregation groups (LAGs))

- On switches that support the ELS configuration style, inter-VLAN routing functions use an integrated routing and bridging (IRB) interface named irb

- On switches that do not support the ELS configuration style, inter-VLAN routing functions use a routed VLAN interface (RVI) named vlan

There are two general types of classifiers:

- Behavior aggregate (BA) classifiers

- Multifield (MF) classifiers

You can configure both a BA classifier and an MF classifier on an interface. If you do this, the BA classification is performed first and then the MF classification. If the two classification results conflict, the MF classification result overrides the BA classification result.

Behavior Aggregate Classifiers

BA classifiers are based on fixed-length fields in the packet header, which makes them computationally more efficient than MF classifiers. Therefore core devices that handle high traffic volumes are normally configured to perform BA classification. The BA classifier maps packets to a forwarding class and a loss priority. The forwarding class determines the output queue for a packet. The loss priority is used by a scheduler to control packet discard during periods of congestion.

These are the following types of BA classifiers:

- dscp—Differentiated Services Code Point (DSCP) for IP DiffServ. Handles incoming IPv4 packets.

- dscp-ipv6—Handles incoming IPv6 packets.

- ieee-802.1—Handles Layer 2 CoS (IEEE 802.1p).

- inet-precedence—Handles incoming IPv4 packets. IP precedence mapping requires only the upper three bits of the DSCP field.

A BA classifier takes a specified CoS value as either the literal bit pattern or as a defined alias and attempts to match it to the type of packet arriving on the interface. If the information in the packet's header matches the specified pattern, the packet is sent to the appropriate queue, defined by the forwarding class associated with the classifier.

Default Behavior Aggregate Classification

Juniper Networks Junos operating system (Junos OS) automatically assigns implicit default BA classifiers to logical interfaces based on the type of interface. lists different types of interfaces and the corresponding implicit default BA classification.

Table 9: Default BA Classification

Type of Interface	Default BA Classification
Trunk interfaces	`ieee8021p-default`
Layer 3 interface (IPv4)	`dscp-default`
Layer 3 interface (IPv6)	`dscp-ipv6-default`
Access interface	Untrusted
Routed VLAN interface (*RVI*)	No default classification

When you explicitly associate a BA classifier with a *logical interface*, you are overriding the implicit (default) BA classifier with an explicit BA classifier.

describes the BA classifier types you can configure on Layer 2 and Layer 3 interfaces.

Table 10: Allowed BA Classification

Type of Interface	Allowed BA Classification
Layer 2 interface	IEEE 802.1p, IP precedence, DSCP, DSCP IPv6
Layer 3 interface (IPv4)	IEEE 802.1p, IP precedence, DSCP
Layer 3 interface (IPv6)	IEEE 802.1p, IP precedence, DSCP IPv6

You cannot apply DSCP and IP precedence classifiers to the same interface. You also cannot apply IEEE 802.1p classifiers to an interface with classifiers of any other type. DSCP IPv6 classifiers can be applied to an interface with either DSCP or IP precedence classifiers, because they apply to different types of packets.

If you have not explicitly configured a classifier on a logical interface, the default classifiers are assigned and classification works as follows:

- To a logical interface configured with an IPv4 address, a DSCP classifier is assigned by default, and IPv4 and IPv6 packets are classified using the DSCP classifier.

- To logical interface configured with an IPv6 address, a DSCP IPv6 classifier is assigned by default, and IPv4 and IPv6 packets are classified using the DSCP IPv6 classifier.

You can configure IRB interfaces on switches that support the ELS configuration style, or routed VLAN interfaces on switches that do not support the ELS configuration style. After you do this, the User Priority (UP) bits in the incoming packets are rewritten according to the default IEEE 802.1p rewrite rule.

> **(i) NOTE**: By default, all BA classifiers classify traffic into either the best-effort forwarding class or the network-control forwarding class.

Multifield Classifiers

Multifield (MF) classifiers examine multiple fields in a packet such as source and destination addresses and source and destination port numbers of the packet. With MF classifiers, you set the forwarding class and loss priority of a packet based on *firewall filter* rules.

MF classification is normally performed at the network edge because of the general lack of support for DSCP or IP precedence classifiers in end-user applications. On an edge switch, an MF classifier provides the filtering functionality that scans through a variety of packet fields to determine the forwarding class for a packet. Typically, any classifier performs matching operations on the selected fields against a configured value.

Platform-Specific Classifier Behavior

Use the following table to review platform-specific behaviors for your platforms.

Table 11: Platform-Specific Classifier Behavior

Platform	Difference
EX4300	- On EX4300 switches, the three classifiers (DSCP, DSCP IPv6, and IEEE 802.1p) can co-exist on an L2 interface along with a fixed classifier. BA classification takes precedence over fixed classification.

Defining CoS Classifiers (CLI Procedure)

Packet classification associates incoming packets with a particular CoS servicing level. Classifiers associate packets with a forwarding class and loss priority and assign packets to output queues based on the associated forwarding class. Junos OS supports two general types of classifiers:

- Behavior aggregate (BA) classifier—Examine the CoS value in the packet header. The value in this single field determines the CoS settings applied to the packet. BA classifiers allow you to set the forwarding class and loss priority of a packet based on the Differentiated Services code point (DSCP) value, IP precedence value, or IEEE 802.1p value. EX Series switches except EX4300 switches support two types of loss priorities: `high` and `low`. EX4300 switches support three types of loss priorities: `high`, `medium-high`, and `low`.

  You can configure BA classifiers for the following CoS marker types:

  - **dscp** and **dscp-ipv6**—Handles incoming IPv4 and IPv6 packets, respectively.

  - **ieee-802.1**—Handles Layer 2 CoS.

  - **inet-precedence**—Handles incoming IPv4 packets. IP precedence mapping requires only the higher order three bits of the DSCP field.

- Multifield (MF) classifier—Examine multiple fields in the packet such as source and destination addresses and source and destination port numbers of the packet. With MF classifiers, you set the forwarding class and loss priority of a packet based on firewall filter rules.

The following example describes how to configure a BA classifier (**ba-classifier**) as the default DSCP map for handling IPv4 traffic and to apply the BA classifier to either a specific Gigabit Ethernet interface or to all the Gigabit Ethernet interfaces on the switch. The BA classifier assigns loss priorities, as shown in Table 12 on page 71, to incoming packets in the four forwarding classes.

You can use the same procedure to set MF classifiers (except that you would use firewall filter rules).

Table 12: BA-classifier Loss Priority Assignments

Forwarding Class	For CoS Traffic Type	ba-classifier Assignment
be	Best-effort traffic	High-priority code point: **000001**
ef	Expedited-forwarding traffic	High-priority code point: **101110**
af	Assured-forwarding traffic	High-priority code point: **001100**
nc	Network-control traffic	High-priority code point: **110001**

To configure a DSCP BA classifier named **ba-classifier** as the default DSCP map:

- Associate code point **000001** with forwarding class **be** and loss priority **high:**

```
[edit class-of-service classifiers]
user@switch# set dscp ba-classifier import default forwarding-class be loss-priority high
code-points 000001
```

- Associate code point **101110** with forwarding class **ef** and loss priority **high:**

```
[edit class-of-service classifiers]
user@switch# set dscp ba-classifier forwarding-class ef loss-priority high code-points 101110
```

- Associate code point **001100** with forwarding class **af** and loss priority **high:**

```
[edit class-of-service classifiers]
user@switch# set dscp ba-classifier forwarding-class af loss-priority high code-points 001100
```

- Associate code point **110001** with forwarding class **nc** and loss priority **high:**

```
[edit class-of-service classifiers]
user@switch# set dscp ba-classifier forwarding-class nc loss-priority high code-points 110001
```

- Apply the classifier to a specific interface or to all Gigabit Ethernet interfaces on the switch.

- To apply the classifier to a specific interface:

```
[edit class-of-service interfaces]
user@switch# set ge-0/0/0 unit 0 classifiers dscp ba-classifier
```

- To apply the classifier to all Gigabit Ethernet interfaces on the switch, use wildcards for the interface name and the logical-interface (unit) number:

```
[edit class-of-service interfaces]
user@switch# set ge-* unit * classifiers dscp ba-classifier
```

RELATED DOCUMENTATION

Defining CoS Classifiers (J-Web Procedure)

(i) **NOTE**: This topic applies only to the J-Web Application package.

You can use the J-Web interface to define CoS classifiers on an EX Series switch. Classifiers examine the CoS value or alias of an incoming packet and assign the packet a level of service by setting its forwarding class and loss priority.

To define CoS classifiers:

1. Select **Configure** > **Class of Service** > **Classifiers**.

   (i) **NOTE**: After you make changes to the configuration on this page, you must commit the changes immediately for them to take effect. To commit all changes to the active

configuration, select **Commit Options** > **Commit**. See Using the Commit Options to Commit Configuration Changes (J-Web Procedure) for details about all commit options.

2. Click one of the following options:

- **Add**—Adds a classifier. Enter information into the classifier page as described in Table 13 on page 73.

- **Edit**—Modifies an existing classifier. Enter information into the classifier page as described in Table 13 on page 73.

- **Delete**—Deletes an existing classifier.

Table 13: Classifiers Configuration Fields

Field	Function	Your Action
Classifier Name	Specifies the name for a classifier.	To name a classifier, type the name—for example, **ba-classifier**.
Classifier Type	Specifies the type of classifier: **dscp**, **ieee-802.1**, or **inet-precedence**.	Select a value from the list.

Table 13: Classifiers Configuration Fields *(Continued)*

Field	Function	Your Action

Code Point Mapping	Sets the forwarding classes and the packet loss priorities (PLPs) for specific CoS values and aliases.	To add a code point mapping:
		
		1. Click **Add**.
		
		2. Select the code point.
		
		3. Select a forwarding class from the following list:
		
		   • **expedited-forwarding**—Provides low loss, low delay, low jitter, assured bandwidth, and end-to-end service. Packets can be forwarded out of sequence or dropped.
		
		   • **best-effort**—Provides no special CoS handling of packets. Typically, RED drop profile is aggressive and no loss priority is defined.
		
		   • **assured-forwarding**—Provides high assurance for packets within the specified service profile. Excess packets are dropped.
		
		   • **network-control**—Packets can be delayed but not dropped.
		
		4. Select the loss priority.
		
		   To assign a loss priority, select one:
		
		   • **high**—Packet has a high loss priority.
		
		   • **low**—Packet has a low loss priority.

RELATED DOCUMENTATION

Example: Configuring Multidestination (Multicast, Broadcast, DLF) Classifiers

IN THIS SECTION

Packet classification associates incoming packets with a particular CoS servicing level. Behavior aggregate (BA) classifiers examine the CoS value in the packet header to determine the CoS settings applied to the packet. BA classifiers allow you to set the forwarding class and loss priority of a packet based on the incoming CoS value.

On supported platforms, you can apply classifiers to multidestination traffic globally or to a specific interface. If you apply multidestination classifiers both globally and to a specific interface, the classifications on the interface take precedence.

Multidestination classifiers apply to all of the switch interfaces and handle multicast, broadcast, and destination lookup fail (DLF) traffic. You cannot apply a multidestination classifier to a single interface or to a range of interfaces.

Unicast and multidestination traffic must use different classifiers.

Configuring Multidestination Classifiers

Step-by-Step Procedure

To configure a multicast IEEE 802.1 BA classifier named `ba-mcast-classifier`:

1. Associate code point `000` with forwarding class `mcast` and loss priority `low`:

```
[edit class-of-service classifiers]
user@switch# set ieee-802.1 ba-mcast-classifier forwarding-class mcast loss-priority low code-
points 000
```

2. Configure the classifier as a multidestination classifier:

```
[edit class-of-service]
user@switch# set multi-destination classifiers ieee-802.1 ba-mcast-classifier
```

Requirements

This example uses the following hardware and software components:

- One supported Junos device

- Any supported Junos device

Overview

Junos supports three general types of classifiers:

- Behavior aggregate or CoS value traffic classifiers—Examine the CoS value in the packet header. The value in this single field determines the CoS settings applied to the packet. BA classifiers allow you to set the forwarding class and loss priority of a packet based on the CoS value.

- Fixed classifiers. Fixed classifiers classify all ingress traffic on a physical interface into one forwarding class, regardless of the CoS bits in the VLAN header or the DSCP bits in the packet header.

- Multifield traffic classifiers—Examine multiple fields in the packet such as source and destination addresses and source and destination port numbers of the packet. With multifield classifiers, you set the forwarding class and loss priority of a packet based on firewall filter rules.

Multidestination classifiers apply to all of the switch interfaces and handle multicast, broadcast, and destination lookup fail (DLF) traffic. You cannot apply a multidestination classifier to a single interface or to a range of interfaces.

> (i) **NOTE**: You must assign unicast traffic and multicast traffic to different classifiers. One classifier cannot include both unicast and multicast forwarding classes. A multidestination classifier can include only forwarding classes for multicast traffic.

The following example describes how to configure a BA classifier called `ba-mcast-classifier`, which is applied to all of the switch interfaces. The BA classifier assigns loss priorities, as shown in Table 14 on page 77, to incoming packets in the multidestination forwarding class.

You can also use firewall filters to set multifield classifiers.

Table 14: BA-mcast-classifier Loss Priority Assignments

Multicast Forwarding Class	Traffic Type	ba-mcast-classifier Assignment
`mcast`	Best-effort multicast traffic	Low loss priority code point: `000`

Verification

IN THIS SECTION

- Verifying the IEEE 802.1 Multidestination Classifier | **77**
- Verifying the Multidestination Classifier Configuration | **78**
- Platform-Specific Multidestination Classifier Behavior | **78**

To verify the multidestination classifier configuration, perform these tasks:

Verifying the IEEE 802.1 Multidestination Classifier

Purpose

Verify that the classifier `ba-mcast-classifier` is configured as the IEEE 802.1 multidestination classifier:

Action

Verify the results of the classifier configuration using the operational mode command `show configuration class-of-service multi-destination classifiers ieee-802.1`:

```
user@switch> show configuration class-of-service multi-destination classifiers ieee-802.1
ba-mcast-classifier;
```

Verifying the Multidestination Classifier Configuration

Purpose

Verify that you configured the multidestination classifier with the correct forwarding classes, loss priorities, and code points.

Action

List the classifier configuration using the operational mode command `show configuration class-of-service classifiers ieee-802.1 ba-mcast-classifier`:

```
user@switch> show configuration class-of-service classifiers ieee-802.1 ba-mcast-classifier
    forwarding-class mcast {
        loss-priority low code-points 000;
    }
```

Platform-Specific Multidestination Classifier Behavior

Use the following table to review platform-specific behaviors for your platforms.

Table 15: Platform-Specific Multidestination Classifier Behavior

Platform	Difference
EX4300	• You cannot apply multidestination classifiers to a range of interfaces on an EX4300 switch.

RELATED DOCUMENTATION

Example: Configuring Unicast Classifiers

Defining CoS BA Classifiers (DSCP, DSCP IPv6, IEEE 802.1p)

Monitoring CoS Classifiers

Understanding CoS Classifiers

Understanding CoS Classifiers

Understanding Applying CoS Classifiers and Rewrite Rules to Interfaces

Configuring and Applying IEEE 802.1ad Classifiers

If you apply an IEEE 802.1 classifier to a logical interface, this classifier takes precedence and is not compatible with any other classifier type. You can set the forwarding class and loss priority for traffic on the basis of the three IEEE 802.1p bits (three bits in either the inner virtual LAN (VLAN) tag or the outer VLAN tag) and the drop eligible indicator (DEI) bit. You can apply the default map or customize one or more of the default values.

You then apply the classifier to the interface on which you configure IEEE 802.1ad frame formats.

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

1. Define the custom IEEE 802.1ad map:

   a. Create the classifier by specifying a name for it and defining it as an IEEE-802.1ad (DEI) classifier.

   ```
   [edit]
   user@host# edit class-of-service classifiers ieee-802.1ad dot1p_dei_class
   ```

   b. Assign the forwarding class and loss priority to the code-point alias.

   ```
   [edit class-of-service classifiers ieee-802.1ad dot1p_dei_class]
   user@host# set forwarding-class best-effort loss-priority low code-points [0000 1101]
   ```

2. Apply the classifier to the logical interface:

   a. Specify the interface to which you want to apply the classifier.

   ```
   [edit]
   user@host# edit class-of-service interfaces ge-2/0/0 unit 0
   ```

   b. Specify the name of the classifier you want to apply to the interface.

   ```
   [edit class-of-service interfaces ge-2/0/0 unit 0]
   user@host# set classifiers ieee-802.1ad dot1p_dei_class
   ```

3. Verify the custom IEEE 802.1ad map configuration:

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

```
class-of-service {
    classifiers {
        ieee-802.1ad dot1p_dei_class {
            forwarding-class best-effort {
                loss-priority low code-points [ 0000 1101 ];
            }
        }
    }
}
```

```
class-of-service {
    interfaces {
        ge-2/0/0 {
            unit 0 {
                classifiers {
                    ieee-802.1ad dot1p_dei_class;
                }
            }
        }
    }
}
```

RELATED DOCUMENTATION

Understanding How Behavior Aggregate Classifiers Prioritize Trusted Traffic

Applying Behavior Aggregate Classifiers to Logical Interfaces

Change the Default Queuing and Marking of Host Outbound Traffic

IN THIS SECTION

You can modify the default forwarding class and DSCP bits used in the ToS field of *host outbound traffic* (packets generated by the Routing Engine). You can also modify the queue assignment of the forwarding class to which you assign host outbound traffic.

Host Outbound Traffic Classification Overview

TCP-related packets, such as BGP or LDP, use queue 3 (network control) for retransmitted traffic. Changing the defaults for Routing Engine sourced traffic does not affect transit or incoming traffic. The changes apply to all packets relating to Layer 3 and Layer 2 protocols, but not MPLS EXP bits or IEEE 802.1p bits. This feature applies to all application-level traffic such as FTP or ping operations as well.

The forwarding class selected is global to the device. That is, the traffic is placed in the selected forwarding class on all egress interfaces. In the case of a restricted interface, the Routing Engine sourced traffic flows through the restricted queue.

The forwarding class selected must be properly configured on all interfaces.

To change the default forwarding class and DSCP bits for Routing Engine sourced traffic, include the `host-outbound-traffic` statement at the `[edit class-of-service]` hierarchy level:

```
[edit class-of-service]
host-outbound-traffic {
    forwarding-class class-name;
    dscp-code-point value;
}
```

The following example places all Routing Engine sourced traffic into the network control forwarding class (assigned to queue 3 by default) with a DSCP value of 101010:

```
[edit class-of-service]
host-outbound-traffic {
    forwarding-class network-control;
    dscp-code-point 101010;
}
```

You can also modify the queue assignment of the forwarding class. For example:

```
[edit class-of-service]
forwarding-classes {
    class network-control queue-num 5;
}
```

Platform-Specific Host Outbound Traffic Classification Behavior

Use Feature Explorer to confirm platform and release support for host outbound traffic classification.

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

Platform	Difference
ACX7000 Series Routers	ACX7000 Series Routers support only the `forwarding-class` *class-name* option.

RELATED DOCUMENTATION

Understanding How Forwarding Classes Assign Classes to Output Queues

Default Routing Engine Protocol Queue Assignments

Default DSCP and DSCP IPv6 Classifiers

Monitoring CoS Classifiers

Purpose

> ⓘ **NOTE**: This topic applies only to the J-Web Application package.

Use the monitoring functionality to display the mapping of incoming CoS values to the forwarding class and loss priority for each classifier.

Action

To monitor CoS classifiers in the J-Web interface, select **Monitor** > **Class of Service** > **Classifiers**.

To monitor CoS classifiers in the CLI, enter the following CLI command:

```
show class-of-service classifier
```

Meaning

Table 16 on page 83 summarizes key output fields for CoS classifiers.

Table 16: Summary of Key CoS Classifier Output Fields

Field	Values	Additional Information
Classifier Name	Name of a classifier.	To display classifier assignments, click the plus sign (+).

Table 16: Summary of Key CoS Classifier Output Fields *(Continued)*

Field	Values	Additional Information
CoS Value Type	The classifiers are displayed by type: • **dscp**—All classifiers of the DSCP type. • **ieee-802.1**—All classifiers of the IEEE 802.1 type. • **inet-precedence**—All classifiers of the IP precedence type.	
Index	Internal index of the classifier.	
Incoming CoS Value	CoS value of the incoming packets, in bits. These values are used for classification.	
Assign to Forwarding Class	Forwarding class that the classifier assigns to an incoming packet. This class affects the forwarding and scheduling policies that are applied to the packet as it transits the switch.	
Assign to Loss Priority	Loss priority value that the classifier assigns to the incoming packet based on its CoS value.	

RELATED DOCUMENTATION

Troubleshooting a CoS Classifier Configuration for a TCAM Space Error

Problem

Description

When a CoS classifier configuration exceeds the amount of available ternary content addressable memory (TCAM) space, the switch returns the following system log message:

```
<number_of_rules_being_added> rules for <filter_name> class <filter_class> will not be
installed, key: <bind_point>. no space in tcam db(<shared_pool_information>)
```

The switch returns this message during the commit operation if the number of classifiers defined in the CoS configuration or the number of bind points (interfaces) to which classifiers are bound causes the CoS configuration to exceed the amount of available TCAM space. However, the commit operation for the CoS configuration is completed in the CLI module.

Solution

When a CoS configuration exceeds the amount of available TCAM table space, you must either define fewer classifiers or bind them to fewer interfaces, or both, so that the space requirements for the CoS configuration do not exceed the available space in TCAM.

To delete classifier definitions and bind points in a CoS configuration, and to apply a new CoS classifier definition to fewer bind points:

1. Delete either the CoS classifier definition or the bind points:

    - To delete the CoS classifier definition:

- For behavioral classifiers:

```
[edit class-of-service]
user@switch# delete classifier dscp d1
```

- For multifield classifiers:

```
[edit]
user@switch# delete interfaces ge-3/0/2 unit 0 family ethernet-switching filter input
ipacl
```

This command deletes a multifield classifier defined for a port. Similarly, you can delete a multifield classifier defined for a VLAN or router.

You can also delete terms defined in a single multifield classifier:

```
[edit]
user@switch# delete firewall family inet filter f1 term t1
```

In both these examples (for behavioral and multifield classifiers), the assumption is that too many classifier definitions resulted in the error message.

- To delete the bind points:

```
[edit class-of-service]
user@switch# delete class-of-service interfaces ge-0/0/0
user@switch# delete class-of-service interfaces ge-0/0/1
user@switch# delete class-of-service interfaces ge-0/0/2
user@switch# delete class-of-service interfaces ge-0/0/3
user@switch# delete class-of-service interfaces ge-0/0/4
user@switch# delete class-of-service interfaces ge-0/0/5
user@switch# delete class-of-service interfaces ge-0/0/6
user@switch# delete class-of-service interfaces ge-0/0/7
user@switch# delete class-of-service interfaces ge-0/0/8
```

Here the assumption is that too many bind points (nine) in the configuration resulted in the error message.

2. Commit the operation:

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

3. Define fewer classifiers in the CoS configuration or bind classifiers to fewer interfaces, or both, so that the CoS classifier configuration does not exceed the amount of available TCAM space on the switch:

- To define CoS classifiers:

  - For behavioral classifiers:

    ```
    [edit]
    user@switch# set class-of-service classifiers dscp d2 forwarding-class fc1 loss-
    priority low code-points 000001
    user@switch# set class-of-service classifiers dscp d2 forwarding-class fc2 loss-
    priority low code-points 000010
    user@switch# set class-of-service classifiers dscp d2 forwarding-class fc3 loss-
    priority low code-points 000011
    user@switch# set class-of-service classifiers dscp d2 forwarding-class fc4 loss-
    priority low code-points 000100
    user@switch# set class-of-service classifiers dscp d2 forwarding-class fc5 loss-
    priority low code-points 000101
    user@switch# set class-of-service classifiers dscp d2 forwarding-class fc6 loss-
    priority low code-points 000110
    user@switch# set class-of-service classifiers dscp d2 forwarding-class fc7 loss-
    priority low code-points 000111
    ```

  - For multifield Classifiers:

    ```
    [edit]
    user@switch# set firewall family inet filter f1 term t1 from protocol tcp
    user@switch# set firewall family inet filter f1 term t1 then loss-priority high
    user@switch# set firewall family inet filter f1 term t1 then forwarding-class best-
    effort
    user@switch# set firewall family inet filter f1 term t2 from protocol udp
    user@switch# set firewall family inet filter f1 term t2 then loss-priority high
    user@switch# set firewall family inet filter f1 term t2 then forwarding-class assured-
    forwarding
    user@switch# set firewall family inet filter f1 term t3 from source-port ssh
    ```

```
user@switch# set firewall family inet filter f1 term t3 then loss-priority low
user@switch# set firewall family inet filter f1 term t3 then forwarding-class fc8
user@switch#set class-of-service forwarding-classes best-effort, assured-forwarding,
fc8
```

- To bind classifiers to fewer interfaces:

```
[edit]
user@switch# set class-of-service interfaces ge-0/0/0 unit 0 classifiers dscp d2
user@switch# set class-of-service interfaces ge-0/0/1 unit 0 classifiers dscp d2
user@switch# set class-of-service interfaces ge-0/0/2 unit 0 forwarding-class best-effort
user@switch# set class-of-service interfaces ge-0/0/3 unit 0 forwarding-class assured-
forwarding
user@switch# set class-of-service interfaces ge-0/0/4 unit 0 forwarding-class fc8
```

4. Commit the operation:

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

5. Check system log for an error message. If an error message is not logged, then your classifier configuration has not exceeded the TCAM space limit.

   If an error message is logged, then repeat this procedure by defining fewer classifiers or binding classifiers to fewer bind points.

RELATED DOCUMENTATION

CoS Rewrite

IN THIS CHAPTER

Understanding CoS Rewrite Rules

IN THIS SECTION

As packets enter or exit a network, edge switches might be required to alter the class-of-service (CoS) settings of the packets. This topic describes how to use *rewrite rules* to alter the CoS settings. It covers:

This topic covers:

How Rewrite Rules Work

Rewrite rules set the value of the CoS bits within a packet's header. Each rewrite rule reads the current forwarding class and loss priority associated with the packet, locates the chosen CoS value from a table,

and writes this CoS value into the packet header. For rewrites to occur, rewrite rules must be explicitly assigned to an interface.

On EX Series switches, you can define rewrite rules for IPv4 and IPv6 traffic to network interfaces, aggregated Ethernet interfaces (also known as link aggregation groups (LAGs)), routed VLAN interfaces (RVIs), Layer 3 interfaces, and Layer 3 VLAN-tagged sub-interfaces. Multiple rewrite rules of different types can be assigned to a single interface.

On EX4300 switches, you cannot configure separate DSCPv4 and DSCPv6 rewrite rules on network interfaces, aggregated Ethernet interfaces, Layer 3 interfaces, and integrated routing and bridging (IRB) interfaces. If you configure a DSCPv4 rewrite rule on an interface to rewrite IPv4 traffic, then the same rewrite rule is applied to IPv6 traffic also on that interface, and vice versa. You can define only DSCPv4 rewrite rules on integrated routing and bridging (IRB) interfaces and Layer 3 VLAN-tagged logical interfaces.

In effect, the rewrite rule performs the reverse function of the behavior aggregate (BA) classifier, which is used when the packet enters the switch. As the packet leaves the switch, the final CoS action is generally the application of a rewrite rule.

You configure rewrite rules to alter CoS values in outgoing packets on the outbound interfaces of an edge switch to meet the policies of a targeted peer. This allows the downstream switch in a neighboring network to classify each packet into the appropriate service group.

> **NOTE**: When an IP precedence rewrite rule is active, bits 3, 4, and 5 of the type-of-service (ToS) byte are always reset to zero when code points are rewritten.

Default Rewrite Rule

To define a rewrite rule on an interface, you can either create your own rewrite rule and enable it on the interface or enable a default rewrite rule. See "Defining CoS Rewrite Rules (CLI Procedure)" on page 91.

Table 17 on page 91 shows the default rewrite-rule mappings. These are based on the default bit definitions of Differentiated Services code point (DSCP), IEEE 802.1p, and IP precedence values and the default forwarding classes. You can configure multiple CoS rewrite rules for DSCP, IP precedence and IEEE 802.1p.

> **NOTE**: By default, rewrite rules are not assigned to an interface. You must explicitly assign a user-defined or system-defined rewrite rule to an interface for the rewrites to occur.

When the CoS values of a packet match the forwarding class and packet-loss-priority (PLP) values, the switch rewrites markings on the packet based on the rewrite table.

Table 17: Default Packet Header Rewrite Mappings

Map from Forwarding Class	PLP Value	Map to DSCP/IEEE 802.1p/IP Precedence Value
expedited-forwarding	low	ef
expedited-forwarding	high	ef
assured-forwarding	low	af11
assured-forwarding	high	af12 (DSCP)
best-effort	low	be
best-effort	high	be
network-control	low	nc1/cs6
network-control	high	nc2/cs7

Defining CoS Rewrite Rules (CLI Procedure)

You configure rewrite rules to alter CoS values in outgoing packets on the outbound interfaces of an EX Series switch to match the policies of a targeted peer. Policy matching allows the downstream routing platform or switch in a neighboring network to classify each packet into the appropriate service group.

To configure a CoS rewrite rule, create the rule by giving it a name and associating it with a forwarding class, loss priority, and a code point, thus creating a rewrite table, and you can enable the rewrite rule on an interface. On EX Series switches except EX4300 switches, you can also enable a rewrite rule on routed VLAN interfaces (RVIs). On EX4300 switches, you can also enable rewrite rules on integrated routing and bridging (IRB) interfaces. If you need to customize a rewrite rule, you can create a customized rewrite rule using a firewall filter configuration. You can configure CoS rewrite rules for DSCP, IP precedence and IEEE 802.1p.

You can configure rewrite rules for the following CoS marker types:

- **dscp** and **dscp-ipv6**—Handles incoming IPv4 and IPv6 packets, respectively. On EX4300 switches, you cannot configure DSCP IPv4 and DSCP IPv6 rewrite rules on the same interface. If you configure a DSCP IPv4 rewrite rule on an interface to rewrite IPv4 traffic, then the same rewrite rule is applied to IPv6 traffic also on that interface, and vice versa.

- **ieee-802.1**—Handles Layer 2 CoS.

- **inet-precedence**—Handles incoming IPv4 packets. IP precedence mapping requires only the higher order three bits of the DSCP field.

> **(i)** **NOTE**: To replace an existing rewrite rule on the interface with a new rewrite rule of the same type, first explicitly remove the rewrite rule and then apply the new rule.

To create IEEE 802.1p rewrite rules and enable them on Layer 2 interfaces:

- To create an IEEE 802.1p rewrite rule named `customup-rw` in the rewrite table for all Layer 2 interfaces:

```
[edit class-of-service rewrite-rules]
user@switch# set ieee-802.1 customup-rw forwarding-class be loss-priority low code-point 000
user@switch# set ieee-802.1 customup-rw forwarding-class be loss-priority high code-point 001
user@switch# set ieee-802.1 customup-rw forwarding-class af loss-priority low code-point 010
user@switch# set ieee-802.1 customup-rw forwarding-class af loss-priority high code-point 011
user@switch# set ieee-802.1 customup-rw forwarding-class ef loss-priority low code-point 100
user@switch# set ieee-802.1 customup-rw forwarding-class ef loss-priority high code-point 101
user@switch# set ieee-802.1 customup-rw forwarding-class nc loss-priority low code-point 110
user@switch# set ieee-802.1 customup-rw forwarding-class nc loss-priority high code-point 111
```

- To enable an IEEE 802.1p rewrite rule named customup-rw on a Layer 2 interface:

```
[edit]
user@switch# set class-of-service interfaces ge-0/0/0 unit 0 rewrite-rules ieee-802.1
customup-rw
```

(On EX4300 switches) To enable an IEEE 802.1p rewrite rule named customup-rw on a Layer 2
interface:

```
[edit]
user@switch# set class-of-service interfaces ge-0/0/0 rewrite-rules ieee-802.1 customup-rw
```

- To enable an IEEE 802.1p rewrite rule named customup-rw on all Gigabit Ethernet interfaces on the
  switch, use wildcards for the interface name and logical-interface (unit) number:

```
[edit]
user@switch# set class-of-service interfaces ge-* unit * rewrite-rules customup-rw
```

(On EX4300 switches) To enable an IEEE 802.1p rewrite rule named customup-rw on all Gigabit
Ethernet interfaces on the switch, use wildcards for the interface name:

```
[edit]
user@switch# set class-of-service interfaces ge-* rewrite-rules customup-rw
```

RELATED DOCUMENTATION

Defining CoS Rewrite Rules (J-Web Procedure)

NOTE: This topic applies only to the J-Web Application package.

You can use the J-Web interface to define CoS rewrite rules. Use the rewrite rules to alter the CoS
values in outgoing packets to meet the requirements of the targeted peer. A rewrite rule examines the
forwarding class and loss priority of a packet and sets its bits to a corresponding value specified in the
rule.

To define rewrite rules:

1. Select **Configure** > **Class of Service** > **Rewrite Rules**.

> **NOTE**: After you make changes to the configuration on this page, you must commit the changes immediately for them to take effect. To commit all changes to the active configuration, select **Commit Options** > **Commit**. See Using the Commit Options to Commit Configuration Changes (J-Web Procedure) for details about all commit options.

2. Select one of the following options:

   - **Add**—Adds a rewrite rule. Enter information into the rewrite rule page as described in Table 18 on page 94.

   - **Edit**—Modifies an existing rewrite rule. Enter information into the rewrite rule page as described in Table 18 on page 94.

   - **Delete**—Deletes an existing rewrite rule.

Table 18: Rewrite Rules Configuration Page Summary

Field	Function	Your Action
Rewrite Rule Name	Specifies the name for the rewrite rule.	To name a rule, type the name—for example, **rewrite-dscps**.
Rewrite rule type	Specifies the type of rewrite rule: **dscp**, **ieee-802.1**, or **inet-precedence**.	Select a value from the list.

Table 18: Rewrite Rules Configuration Page Summary *(Continued)*

Field	Function	Your Action
Code Point Mapping	Rewrites outgoing CoS values of a packet based on the forwarding class and loss priority. Allows you to remove a code point mapping entry.	To configure a CoS value assignment, follow these steps: To add a code point mapping: 1. Click **Add**. 2. Select the code point. 3. Select a forwarding class from the following list: • **expedited-forwarding**—Provides low loss, low delay, low jitter, assured bandwidth, and end-to-end service. Packets can be forwarded out of sequence or dropped. • **best-effort**—Provides no special CoS handling of packets. Typically, RED drop profile is aggressive and no loss priority is defined. • **assured-forwarding**—Provides high assurance for packets within the specified service profile. Excess packets are dropped. • **network-control**—Packets can be delayed but not dropped. 4. Select the loss priority. To assign a loss priority, select one: • **high**—Packet has a high loss priority. • **low**—Packet has a low loss priority. To edit an existing code point mapping, select it and click **Edit**. To remove a code point mapping entry, select it and click **Remove**.

Classifiers and Rewrite Rules at the Global, Physical, and Logical Interface Levels Overview

IN THIS SECTION

Depending on your platform, CoS supports classification and rewrite at the global, physical, and logical interface levels.

At a global level, depending on your platform, you can define EXP classification.

At a physical and logical interface level, depending on your platform, you can define the following features:

- DSCP, DSCP-IPV6, and IPv4 precedence classifiers

- DSCP, DSCP-IPV6, and IPv4 precedence rewrites

- EXP classifiers

- EXP rewrites

- IEEE 802.1p and IEEE 802.1ad classifiers (inner and outer)

- IEEE 802.1p and IEEE 802.1ad rewrites (outer)

The IEEE 802.1ad classifier uses IEEE 802.1p and DEI bits together.

> **NOTE**: You cannot configure both IEEE 802.1p and IEEE 802.1ad classifiers together.

To configure global EXP classifiers, include the **classfiers exp** *classifier-name* statement at the **[edit class-of-service system-defaults]** hierarchy level.

To configure classifiers or *rewrite rules* at the physical interface, include either the **classifiers** statement or the **rewrite-rules** statement at the **[edit class-of-service interfaces** *interface-name*] hierarchy level.

To configure fixed classifiers or rewrite rules at the logical interface, include the **forwarding-class** *fc* or the **rewrite-rules** statement at the **[edit class-of-service interfaces** *interface-name* **unit** *number*] hierarchy level.

To configure EXP rewrite at the logical interface, include the **[edit class-of-service interfaces** *interface-name* **unit** *number* **rewrite-rules exp** *rewrite-rule*] statement.

To display classifiers configured under **system-defaults**, enter the **show class-of-service system-defaults** command.

To display classifiers and rewrite rules bound to physical interfaces, enter the **show class-of-service interfaces** *interface-name* command.

Platform-Specific Classifier Behavior

Use the following table to review platform-specific behaviors for your platforms.

Table 19: Platform-Specific Classifier Behavior

Platform	Difference
ACX6360	The ACX6360 router does not support L2 (IEEE802.1p and IEEE802.1ad) classifiers.
ACX7000 Series	- ACX7000 Series routers support classifiers of all types (Inet-Prec/DSCP/DSCP-v6/EXP/ IEEE-802.1p/IEEE-802.1ad) only at the logical interface level. - ACX7000 Series routers do not support global classifiers.

Platform-Specific Rewrite Rule Behavior

Use the following table to review platform-specific behaviors for your platforms.

Table 20: Platform-Specific Rewrite Rule Behavior

Platform	Difference
ACX6360	The ACX6360 router does not support rewrite rules.
ACX7000 Series	ACX7000 Series routers support rewrite rules of all types (Inet-Prec/DSCP/DSCP-v6/EXP/IEEE-802.1p/IEEE-802.1ad) only at the logical interface level.

RELATED DOCUMENTATION

Configuring Classifiers and Rewrite Rules at the Global and Physical Interface Levels

Applying Classifiers and Rewrite Rules at the Global, Physical, and Logical Interface Levels

IN THIS SECTION

- Rewrite Rules on ACX7000 Series Routers | **100**

Depending on your platform, CoS supports classification and rewrite at the global, physical, and logical interface levels.

To apply a global EXP classifier, include the following statements at the **[edit class-of-service] system-defaults** hierarchy level.

```
[edit class-of-service]
user@device# set system-defaults classifiers exp classifier-name
```

On supported platforms, CoS supports one global system default classifier of the EXP type, as shown in the following example:

```
[edit class-of-service]
{
    system-defaults {
        classifiers {
            exp exp-classf-core;
        }
    }
}
```

To configure classifiers and rewrite rules at the physical interface level, include the following statements at the **[edit class-of-service] interfaces** hierarchy level.

```
[edit class-of-service]
interfaces {
    interface-name
        classifiers dscp classifier-name
        classifiers inet-precedence classifier-name
        classifiers ieee-802.1 [vlan-tag (outer | inner)] classifier-name
        rewrite-rules dscp rewrite-name
        rewrite-rules inet-prec  rewrite-name
        rewrite-rules ieee-802.1  rewrite-name
}
```

The following example shows classifiers and rewrite rules configured on physical and logical interfaces:

```
ge-0/1/0 {
    unit 0 {
        rewrite-rules {
            exp custom-exp;
        }
    }
    classifiers {
        dscp d1;
        ieee-802.1 ci;
    }
    rewrite-rules {
        dscp default;
    }
```

```
    }
    ge-0/1/2 {
        classifiers {
            ieee-802.1 ci;
        }
        rewrite-rules {
            ieee-802.1 ri;
        }
    }
    ge-0/1/3 {
        unit 0 {
            rewrite-rules {
                exp custom-exp2;
            }
        }
    }
    ge-0/1/7 {
        classifiers {
            dscp d1;
        }
    }
    ge-0/1/8 {
        classifiers {
            dscp d1;
        }
    }
```

Rewrite Rules on ACX7000 Series Routers

ACX7000 Series routers do not support rewrite functionality on ethernet-bridge and ethernet-ccc encapsulation. For rewrite rules to work on an ACX7000 Series router, the router always needs tagged packets to be present, and the logical interface must be VLAN aware. For example, only packet forwarding will work with the following configuration:

```
[edit]
set interfaces et-0/0/6 encapsulation ethernet-ccc
set interfaces et-0/0/6 unit 0 family ccc
```

For both forwarding and rewrite to work on a logical interface, use the following configuration:

```
[edit]
set interfaces et-0/0/0 flexible-vlan-tagging
set interfaces et-0/0/0 encapsulation flexible-ethernet-services
set interfaces et-0/0/0 unit 0 encapsulation vlan-ccc
set interfaces et-0/0/0 unit 0 vlan-id-list [1-4094]
set interfaces et-0/0/0 unit 0 family ccc
```

RELATED DOCUMENTATION

Classifiers and Rewrite Rules at the Global, Physical and Logical Interface Levels Overview

Host Outbound Traffic IEEE802.1p Rewrite

SUMMARY

This topic describes how to rewrite the IEEE 802.1p field in the Ethernet frame header for host outbound traffic, that is, traffic generated by the Routing Engine. You can set a global value for the priority code point that applies to all host outbound traffic. Additionally, or alternatively, you can specify that rewrite rules are applied to all host outbound traffic on egress logical interfaces. These are rules that you have previously configured to set the IEEE 802.1p field for data traffic on those interfaces.

IN THIS SECTION

- Configure a Global Default IEEE 802.1p Value for All Host Outbound Traffic | **102**
- Apply Egress Interface Rewrite Rules to the IEEE 802.1p Field for All Host Outbound Traffic on an Interface | **102**
- Platform-Specific Host Outbound Traffic Rewrite Behavior | **104**

To configure the IEEE 802.1p field settings:

1. (Optional) Specify a global default value for the IEEE 802.1p field for all host outbound traffic.

2. (Optional) Specify that the IEEE 802.1p rewrite rules for the egress logical interfaces are applied to all host outbound traffic on those interfaces.

Configure a Global Default IEEE 802.1p Value for All Host Outbound Traffic

SUMMARY

This task describes how to configure a global default value for the IEEE 802.1p field for all host outbound traffic.

To configure a global default value for the IEEE 802.1p field:

1. Specify the value. The value can be a three-bit binary number or code point alias.

```
[edit class-of-service host-outbound-traffic ieee-802.1]
user@host# set default value
```

2. For example, specify that a value of 010 is applied to all host outbound traffic:

```
[edit class-of-service host-outbound-traffic ieee-802.1]
user@host# set default 010
```

Apply Egress Interface Rewrite Rules to the IEEE 802.1p Field for All Host Outbound Traffic on an Interface

SUMMARY

This topic describes how to apply rewrite rules for egress logical interfaces to the IEEE 802.1p field for all host outbound traffic on those interfaces.

This task requires separately configured rewrite rules that map packet loss priority information to the code point value in the 802.1p field for data traffic on egress logical interfaces. See *Rewriting Packet Headers to Ensure Forwarding Behavior*.

To configure the rewrite rules:

1. Configure the CoS rewrite rules to map the forwarding class to the desired value for the 802.1p field. See *Configuring Rewrite Rules*.

2. Associate the rewrite rules to the desired egress logical interfaces. See *Applying Rewrite Rules to Output Logical Interfaces*.

3. (Optional) Configure the forwarding class for host outbound traffic. Do not configure this forwarding class if you want to use the default forwarding class assignment (input classification). See *Overriding the Input Classification*.

To configure the rewrite rules to apply to the host outbound traffic IEEE 802.1p field:

> (i) **NOTE**: Enabling IEEE 802.1p rewrite rules for host outbound traffic without creating any corresponding IEEE 802.1p rewrite rules sets the IEEE 802.1p code point to be automatically 000 for all host generated traffic that exits that interface.

Configure the rewrite rules:

```
[edit class-of-service host-outbound-traffic ieee-802.1]
user@host# set rewrite-rules
```

For example:

```
[edit class-of-service]
rewrite-rules {
    ieee-802.1 rewrite_foo {
        forwarding-class network-control {
            loss-priority low code-point 101;
        }
    }
}
interfaces {
    et-1/0/0 {
        unit 100 {
            rewrite-rules {
                ieee-802.1 rewrite_foo vlan-tag outer-and-inner;
            }
        }
    }
}
host-outbound-traffic {
    forwarding-class network-control;
    ieee-802.1 {
        rewrite-rules;
```

```
    }
  }
```

Platform-Specific Host Outbound Traffic Rewrite Behavior

Use Feature Explorer to confirm platform and release support for 802.1p rewrite rules for host outbound traffic.

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

Platform	Difference
ACX7000 Series	ACX7000 Series routers support only the `default` option to configure a global default value for the IEEE 802.1p field for all host outbound traffic.
PTX10000 Series	PTX10000 Series routers support only the `default` option to configure a global default value for the IEEE 802.1p field for all host outbound traffic.

RELATED DOCUMENTATION

Rewriting Packet Headers to Ensure Forwarding Behavior

Monitoring CoS Rewrite Rules

IN THIS SECTION

Purpose

> **NOTE**: This topic applies only to the J-Web Application package.

Use the monitoring functionality to display information about CoS value rewrite rules, which are based on the forwarding class and loss priority.

Action

To monitor CoS rewrite rules in the J-Web interface, select **Monitor** > **Class of Service** > **Rewrite Rules**.

To monitor CoS rewrite rules in the CLI, enter the following command:

```
show class-of-service rewrite-rules
```

Meaning

Table 21 on page 105 summarizes key output fields for CoS rewrite rules.

Table 21: Summary of Key CoS Rewrite Rules Output Fields

Field	Values	Additional Information
Rewrite Rule Name	Names of rewrite rules.	
CoS Value Type	Rewrite rule type: • **dscp**—For IPv4 DiffServ traffic. • **exp**—For MPLS traffic. • **ieee-802.1**—For Layer 2 traffic. • **inet-precedence**—For IPv4 traffic.	To display forwarding classes, loss priorities, and rewritten CoS values, click the plus sign (+).
Index	Internal index for this particular rewrite rule.	

Table 21: Summary of Key CoS Rewrite Rules Output Fields *(Continued)*

Field	Values	Additional Information
Forwarding Class	Forwarding class that is used to determine CoS values for rewriting in combination with loss priority.	Rewrite rules are applied to CoS values in outgoing packets based on forwarding class and loss priority setting.
Loss Priority	Loss priority that is used to determine CoS values for rewriting in combination with forwarding class.	
Rewrite CoS Value To	Value that the CoS value is rewritten to.	

RELATED DOCUMENTATION

CHAPTER 7

Forwarding Classes

Understanding CoS Forwarding Classes

Class-of-Service (CoS) forwarding classes can be thought of as output queues. In effect, the result of classifying packets is the identification of an output queue for a particular packet. For a classifier to assign an output queue to a packet, it must associate the packet with one of the following forwarding classes:

- best-effort (be)—Provides no service profile. Loss priority is typically not carried in a CoS value.

- expedited-forwarding (ef)—Provides a low loss, low latency, low *jitter*, assured bandwidth, end-to-end service.

- assured-forwarding (af)—Provides a group of values you can define and includes four subclasses: AF1, AF2, AF3, and AF4, each with two drop probabilities: low and high.

- network-control (nc)—Supports protocol control and thus is typically high priority.

- multicast best-effort (mcast-be)—Provides no service profile for multicast packets.

- multicast expedited forwarding (mcast-ef)—Supports high-priority multicast packets.

- multicast assured-forwarding (mcast-af)—Provides two drop profiles; high, and low, for multicast packets.

- multicast network-control (mcast-nc)—Supports high-priority multicast packets that are not delay-sensitive.

> **(i)** **NOTE**: The forwarding classes multicast expedited-forwarding, multicast assured-forwarding, multicast network-control, and multicast best-effort are applicable only to Juniper Networks EX4300 Ethernet Switches.

Juniper Networks EX Series Ethernet Switches support up to 16 forwarding classes, thus allowing granular packet classification. For example, you can configure multiple classes of expedited forwarding (EF) traffic such as EF, EF1, and EF2.

EX Series switches support up to eight output queues, except EX4300 switches, which support 12 output queues. Therefore, if you configure more forwarding classes that the number of queues supported, you must map multiple forwarding classes to one or more output queues.

This topic describes:

Default Forwarding Classes

Table 22 on page 109 shows the four default forwarding classes defined for unicast traffic, and Table 23 on page 109 shows the default forwarding classes defined for multicast traffic.

> **(i)** **NOTE**: The default forwarding classes for multicast traffic are applicable only to EX4300 switches.

You can rename the forwarding classes associated with the queues supported on your switch. Assigning a new class name to an output queue does not alter the default classification or scheduling that is applicable to that queue. However, because CoS configurations can be quite complicated, we recommend that you avoid altering the default class names or queue number associations.

Table 22: Default Forwarding Classes for Unicast Traffic

Forwarding Class Name	Comments
best-effort (be)	The software does not apply any special CoS handling to packets with 000000 in the DiffServ field. This is a backward compatibility feature. These packets are usually dropped under congested network conditions.
expedited-forwarding (ef)	The software delivers assured bandwidth, low loss, low delay, and low delay variation (jitter) end-to-end for packets in this service class. The software accepts excess traffic in this class, but in contrast to the assured forwarding class, the out-of-profile expedited-forwarding class packets can be forwarded out of sequence or dropped.
assured-forwarding (af)	The software offers a high level of assurance that the packets are delivered as long as the packet flow from the customer stays within a certain service profile that you define. The software accepts excess traffic, but it applies a tail drop profile to determine that excess packets are dropped, and not forwarded. Two drop probabilities (low and high) are defined for this service class.
network-control (nc)	The software delivers packets in this service class with a high priority. (These packets are not delay-sensitive.) Typically, these packets represent routing protocol hello or keep alive messages. Because loss of these packets jeopardizes proper network operation, packet delay is preferable to packet discard for these packets.

Table 23: Default Forwarding Classes for Multicast Traffic on EX4300 Switches

Forwarding Class Name	Comments
multicast best-effort (mcast-be)	The software does not apply any special CoS handling to multicast packets. These packets are usually dropped under congested network conditions.

Table 23: Default Forwarding Classes for Multicast Traffic on EX4300 Switches *(Continued)*

Forwarding Class Name	Comments
multicast expedited-forwarding (mcast-ef)	The software delivers assured bandwidth, low loss, low delay, and low delay variation (jitter) end-to-end for multicast packets in this service class. The software accepts excess traffic in this class, but in contrast to the multicast assured forwarding class, out-of-profile multicast expedited-forwarding class packets can be forwarded out of sequence or dropped.
multicast assured-forwarding (mcast-af)	The software offers a high level of assurance that the multicast packets are delivered as long as the packet flow from the customer stays within a certain service profile that you define. The software accepts excess traffic, but it applies a tail drop profile to determine if the excess packets are dropped and not forwarded. Two drop probabilities (low and high) are defined for this service class.
multicast network-control (mcast-nc)	The software delivers packets in this service class with a high priority. (These packets are not delay-sensitive.) Typically, these packets represent routing protocol hello or keep alive messages. Because loss of these packets jeopardizes proper network operation, packet delay is preferable to packet discard for these packets.

The following rules govern queue assignment:

- CoS configurations that specify more queues than the switch can support are not accepted. If you commit such a configuration, the commit fails and a message displays that states the number of queues available.

- All default CoS configurations are based on queue number. The name of the forwarding class that is displayed in the default configuration for a queue number is that of the forwarding class currently associated with that queue.

RELATED DOCUMENTATION

Defining CoS Forwarding Classes (CLI Procedure)

Forwarding classes allow you to group packets for transmission. Based on forwarding classes, you assign packets to output queues.

By default, four categories of forwarding classes are defined: best effort, assured forwarding, expedited forwarding, and network control. EX Series switches support up to 16 forwarding classes.

You can configure forwarding classes in one of the following ways:

- Using `class` statement—You can configure up to 16 forwarding classes and you can map multiple forwarding classes to single queue.

- Using `queue` statement—You can configure up to 8 forwarding classes and you can map one forwarding class to one queue.

  This example uses the `class` statement to configure forwarding classes.

To configure CoS forwarding classes, map the forwarding classes to queues:

```
[edit class-of-service forwarding-classes]
user@switch# set class be queue-num 0
user@switch# set class ef queue-num 1
user@switch# set class af queue-num 2
user@switch# set class nc queue-num 3
user@switch# set class ef1 queue-num 4
user@switch# set class ef2 queue-num 5
user@switch# set class af1 queue-num 6
user@switch# set class nc1 queue-num 7
```

RELATED DOCUMENTATION

Defining CoS Forwarding Classes (J-Web Procedure)

> ⓘ **NOTE**: This topic applies only to the J-Web Application package.

You can define CoS forwarding classes on an EX Series switch using the J-Web interface. Assigning a forwarding class to a queue number affects the scheduling and marking of a packet as it transits a switch.

To define forwarding classes:

1. Select **Configure** > **Class of Service** > **Forwarding Classes**.

> ⓘ **NOTE**: After you make changes to the configuration on this page, you must commit the changes immediately for them to take effect. To commit all changes to the active configuration, select **Commit Options** > **Commit**. See Using the Commit Options to Commit Configuration Changes (J-Web Procedure) for details about all commit options.

2. Select one of the following options:

   • **Add**—Adds a forwarding class. Enter information into the forwarding class page as described in Table 24 on page 112.

   • **Edit**—Modifies an existing forwarding class. Enter information into the forwarding class page as described in Table 24 on page 112.

   • **Delete**—Deletes an existing forwarding class.

Table 24: Forwarding Classes Configuration Fields

Field	Function	Your Action
Forwarding Class Summary		

Table 24: Forwarding Classes Configuration Fields *(Continued)*

Field	Function	Your Action
Queue #	Specifies the internal queue numbers to which forwarding classes are assigned. By default, if a packet is not classified, it is assigned to the class associated with queue 0. You can have more than one forwarding class to a queue number.	To specify an internal queue number, select an integer from 0 through 11, appropriate for your platform as follows: **NOTE**: For EX2300 and EX2300-C switches, a maximum of eight egress queues are supported per port. To specify an internal queue number select an integer from 0 through 7.
Forwarding Class Name	Specifies the forwarding class names assigned to specific internal queue numbers. By default, four forwarding classes are assigned to queue numbers 0 (best-effort), 1 (assured-forwarding), 5 (expedited-forwarding), and 7 (network-connect).	Type the name—for example, be-class.

RELATED DOCUMENTATION

Monitoring CoS Forwarding Classes

Purpose

> ⓘ **NOTE**: This topic applies only to the J-Web Application package.

View the current assignment of CoS forwarding classes to queues on the switch.

Action

To monitor CoS forwarding classes in the J-Web interface, select **Monitor** > **Class of Service** > **Forwarding Classes**.

To monitor CoS forwarding classes in the CLI, enter the following CLI command:

```
show class-of-service forwarding-class
```

Meaning

Table 25 on page 115 summarizes key output fields for CoS forwarding classes.

Table 25: Summary of Key CoS Forwarding Class Output Fields

Field	Values
Forwarding Class	Names of forwarding classes assigned to queue numbers. The following are the default forwarding classes: • **best-effort**—Provides no special CoS handling of packets. Loss priority is typically not carried in a CoS value. • **expedited-forwarding**—Provides low loss, low delay, low jitter, assured bandwidth, and end-to-end service. • **assured-forwarding**—Provides high assurance for packets within the specified service profile. Excess packets are dropped. • **network-control**—Packets can be delayed but not dropped. EX4300 switches have the following additional default forwarding classes: • **mcast-be**—Provides no special CoS handling of packets. • **mcast-ef**—Provides low loss, low delay, low jitter, assured bandwidth, and end-to-end service. • **mcast-af**—Provides high assurance for packets within the specified service profile. Excess packets are dropped. • **mcast-nc**—Provides multicast network-control traffic.
Queue	Queue number corresponding to the forwarding class name. The default forwarding classes are assigned as follows: • **best-effort**—0 • **expedited-forwarding**—5 • **assured-forwarding**—1 • **network-control**—7 • **mcast-be**—2 • **mcast-ef**—4 • **mcast-af**—6

RELATED DOCUMENTATION

CHAPTER 8

Flow Control

Understanding Priority-Based Flow Control

Priority-based flow control (PFC), IEEE standard 802.1Qbb, is a link-level flow control mechanism. The flow control mechanism is similar to that used by IEEE 802.3x Ethernet PAUSE, but it operates on individual priorities. Instead of pausing all traffic on a link, PFC allows you to selectively pause traffic according to its class.

This topic describes:

Reliability of Packet Delivery in Standard Ethernet Networks and in Layer 2 Networks

Standard Ethernet does not guarantee that a packet injected into the network will arrive at its intended destination. Reliability is provided by upper-layer protocols. Generally, a network path consists of multiple hops between the source and destination. A problem arises when transmitters send packets faster than receivers can accept them. When receivers run out of available buffer space to hold incoming flows, they silently drop additional incoming packets. This problem is generally resolved by upper-layer protocols that detect the drops and request retransmission.

Applications that require reliability in Layer 2 must have flow control that includes feedback from a receiver to a sender regarding buffer availability. Using IEEE 802.3x Ethernet PAUSE control frames, a receiver can generate a MAC control frame and send a PAUSE request to a sender when a specified threshold of receiver buffer has been filled to prevent buffer overflow. Upon receiving a PAUSE request, the sender stops transmission of any new packets until the receiver notifies the sender that it has sufficient buffer space to accept them again. The disadvantage of using Ethernet PAUSE is that it operates on the entire link, which might be carrying multiple traffic flows. Some traffic flows do not need flow control in Layer 2, because they are carrying applications that rely on upper-layer protocols for reliability. PFC enables you to configure Layer 2 flow control selectively for the traffic that requires it, such as Fibre Channel over Ethernet (FCoE) traffic, without impacting other traffic on the link. You can also enable PFC for other traffic types, such as iSCSI.

Calculations for Buffer Requirements When Using PFC PAUSE

The receive buffer must be large enough to accommodate all data that is received while the system is responding to a PFC PAUSE frame.

When you calculate buffer requirements, consider the following factors:

- Processing and queuing delay of the PFC PAUSE—In general, the time to detect the lack of sufficient buffer space and to transmit the PFC PAUSE is negligible. However, delays can occur if the switch detects a reduction in buffer space just as the transmitter is beginning to transmit a maximum length frame.

- Propagation delay across the media—The delay amount depends on the length and speed of the physical link.

- Response time to the PFC PAUSE frame

- Propagation delay across the media on the return path

> 🛈 **NOTE**: We recommend that you configure at least 20 percent of the buffer size for the queue that is using PFC and that you do not specify the **exact** option.
>
> Because it is mandatory to explicitly configure a certain percentage of buffer size for PFC, you must also explicity configure some buffer size for any other forwarding classes that you are planning to use (including the default forwarding classes and the user-defined forwarding classes). The percentage that you allocate depends on the usage of the respective classes.

How PFC and Congestion Notification Profiles Work With or Without DCBX

PFC can be applied to an interface regardless of whether the Data Center Bridging Capability Exchange protocol (DCBX) is enabled.

However, automatic control and advertisement of PFC requires DCBX:

- When DCBX is enabled—DCBX detects the data center bridging (DCB) neighbor's PFC configuration, uses autonegotiation to advertise local and peer PFC configuration, and then enables or disables PFC depending on whether the configurations are compatible or not. When PFC is enabled, it uses the congestion notification profile, which you have configured and applied to the interface.

- When DCBX is not enabled—*Class of service* (CoS) triggers PFC when the incoming frame has a User Priority (UP) field that matches the three-bit pattern specified for the congestion notification profile.

To manually control the use of PFC on the interface regardless of the configuration of the peer data center devices, you can explicitly change the configuration of DCBX on the interface to disable PFC autonegotiation. See *Disabling DCBX to Disable PFC Autonegotiation on EX Series Switches (CLI Procedure)*. When PFC autonegotiation is disabled, PFC is triggered by the congestion notification profile for PFC regardless of the configuration of the DCB peer.

> **NOTE**: PFC functions effectively only when the peer devices connected to the local interface are also using PFC and are configured compatibly with the local interface. PFC must be symmetrical—if PFC is not configured to use the same traffic class (code point) on both the local and the peer interface, it does not have any impact on the traffic.

Table 26 on page 119 shows the one-to-one mapping between the UP field of an IEEE 802.1Q tagged frame, the traffic class, and the egress queue. In addition to setting a PFC congestion notification profile on an ingress port, you must set a forwarding class to match the priority specified in the PFC congestion notification profile and to forward the frame to the appropriate queue.

Juniper Networks EX Series Ethernet Switches support up to six traffic classes and allow you to associate those classes with six different congestion notification profiles. (The switches support up to 16 forwarding classes.)

Table 26: Input for PFC Congestion Notification Profile and Mapping to Traffic Class and Egress Queue

UP Field of IEEE-802.1Q Tagged Frame	Traffic Class	Egress Queue
000	TC 0	queue 0
001	TC 1	queue 1
010	TC 2	queue 2

Table 26: Input for PFC Congestion Notification Profile and Mapping to Traffic Class and Egress Queue
(Continued)

UP Field of IEEE-802.1Q Tagged Frame	Traffic Class	Egress Queue
011	TC 3	queue 3
100	TC4	queue 4
101	TC 5	queue 5

RELATED DOCUMENTATION

Understanding Data Center Bridging Capability Exchange Protocol for EX Series Switches

Example: Configuring an FCoE Transit Switch

Configuring Priority-Based Flow Control for an EX Series Switch (CLI Procedure) **| 120**

schedulers

congestion-notification-profile

Configuring Priority-Based Flow Control for an EX Series Switch (CLI Procedure)

You can configure priority-based flow control (PFC) to apply link-level flow control on a specific traffic class so that different types of traffic can efficiently use the same network interface card (NIC). You must configure PFC for all interfaces carrying Fibre Channel over Ethernet (FCoE) traffic. You can also configure PFC on interfaces carrying other traffic types, such as Internet small computer system interface (iSCSI) traffic. Using PFC is optional for traffic types other than FCoE.

> *(i)* **NOTE**:
> - PFC is supported only on 10-Gigabit Ethernet interfaces.

- If you are using PFC for a non-FCoE DCBX application, use the same 802.1p code points for the PFC congestion notification profile and for the application map that is carrying that application traffic.

Data Center Bridging Capability Exchange protocol (DCBX) is enabled by default on all 10-Gigabit Ethernet interfaces. DCBX enables or disables PFC on the local interface depending on whether the PFC configuration on that interface is the same as the PFC configuration of the connected interface on the data center bridging (DCB) peer.

> **NOTE**: When you configure PFC, we recommend that you:
>
> - Configure at least 20 percent of the buffer for the queue that is using PFC.
>
> - Configure an appropriate percent of the buffer for any other forwarding classes (default forwarding classes and the user-defined forwarding classes) that you are using.
>
> - Do not specify the `exact` option when configuring the buffer for the queue that is using PFC.
>
> - Configure the `loss-priority` statement to `low` for a traffic class that is using PFC.
>
> - Verify that the PFC configurations of the local interfaces are the same as the PFC configurations of the connected interfaces on the DCB peer. See *show dcbx neighbors*.

EX Series switches support up to six congestion notification profiles for PFC.

To configure PFC:

1. Configure a congestion notification profile, specifying the name of the profile and specifying the three-bit pattern of the User Priority bits in an incoming frame that will trigger the priority-based flow control on that traffic class:

```
[edit class-of-service]
user@switch# set congestion-notification-profile profile-name  input ieee-802.1 code-point
up-bits pfc
```

2. Disable standard Ethernet flow control on the interfaces that will be used for the traffic class that you have selected for PFC:

```
[edit interfaces]
user@switch# set interface-name  ether-options no-flow-control
```

> ⓘ **NOTE**: You cannot apply PFC to interfaces that are using standard Ethernet flow control. You must first disable flow control on those interfaces.

3. Bind the congestion notification profile to the interfaces that will be used for the traffic class that you have selected for PFC:

```
[edit class-of-service]
user@switch# set interfaces interface-name  congestion-notification-profile profile-name
```

4. Create a CoS classifier for a traffic class that will use PFC:

```
[edit class-of-service]
user@switch# set classifiers ieee-802.1 classifier-name import default
```

5. Configure this traffic class (*classifier-name*) to use a user-defined or default forwarding class with a low loss priority value and specify the 802.1p code points::

```
[edit class-of-service]
user@switch# set classifiers ieee-802.1 classifier-name forwarding-class class-name loss-
priority low code-points 3 bit-patterns
```

6. Bind the *classifier-name* classifier to all interfaces that require PFC:

```
[edit class-of-service]
user@switch# set interfaces interface-name unit logical-unit-number classifiers ieee-802.1
classifier-name
```

7. Assign the specified forwarding-class to an egress queue:

```
[edit class-of-service]
user@switch# set forwarding-classes class-name queue-number
```

8. Set a scheduler for this queue, allocating at least 20 percent of the buffer to be used for FCoE traffic:

```
[edit class-of-service]
user@switch# set schedulers scheduler-name  buffer-size percent
```

9. Set a scheduler to allocate buffer space for forwarding classes carrying other traffic:

> **NOTE**: You must explicitly allocate some buffer space for the other forwarding classes. The default allocation of buffer space for forwarding classes is overridden when you manually configure the requisite amount of buffer space for the FCoE traffic.

```
[edit class-of-service]
user@switch# set scheduler-name  buffer-size percent
```

10. Configure a scheduler map that associates the specified scheduler with the specified forwarding class:

```
[edit class-of-service]
user@switch# set scheduler-maps map-name forwarding-class class-name scheduler scheduler-
name
```

For example:

```
[edit class-of-service]
user@switch# set scheduler-maps pfc-map forwarding-class af2 scheduler pfc-sched
user@switch# set scheduler-maps pfc-map forwarding-class best-effort scheduler default-
sched user@switch# set scheduler-maps pfc-map forwarding-class network-control scheduler
default-sched
user@switch# set scheduler-maps pfc-map forwarding-class expedited-forwarding scheduler
default-sched
```

11. Assign the scheduler map to the egress interface:

```
[edit class-of-service]
user@switch# set interfaces interface-name scheduler-map pfc-map
```

RELATED DOCUMENTATION

Example: Configuring an FCoE Transit Switch

Understanding Priority-Based Flow Control | **117**

congestion-notification-profile

CoS Queue Schedulers and Scheduler Maps

IN THIS CHAPTER

Understanding CoS Schedulers

IN THIS SECTION

You use class-of-service (CoS) schedulers to define the properties of output queues on Juniper devices. These properties include the amount of interface bandwidth assigned to the queue, the size of the memory buffer allocated for storing packets, the priority of the queue, and the drop profiles associated with the queue.

You associate the schedulers with forwarding classes by means of scheduler maps. You can then associate each scheduler map with an interface, thereby configuring the queues, packet schedulers, and tail drop processes that operate according to this mapping.

Default Schedulers

Each forwarding class has an associated scheduler priority.

- On EX Series switches (other than Juniper Networks EX4300 Ethernet Switches):

  - Two forwarding classes—best-effort (queue 0) and network-control (queue 7)—are used in the default configuration. By default on these switches, the best-effort forwarding class (queue 0) receives 95 percent of the bandwidth and the buffer space for the output link, and the network-control forwarding class (queue 7) receives 5 percent. The default drop profile causes the buffer to fill completely and then to discard all incoming packets until it has free space.

  - The expedited-forwarding (queue 5) and assured-forwarding (queue 1) classes have no scheduler because no resources are assigned to queue 5 or queue 1, by default. However, you can manually configure resources to be assigned to the expedited-forwarding and assured-forwarding classes.

- On EX4300 switches:

  - Four forwarding classes—best-effort (queue 0), multicast best-effort (queue 8), network-control (queue 3), and multicast network-control (queue 11)—are used in the default configuration. By default, all the multicast traffic flows through the multicast best-effort queue. EX4300 switches support 12 queues (0–11). The default scheduler transmission rates for queues 0 through 11 are 75, 0, 0, 5, 0, 0, 0, 0, 15, 0, 0 and 5 percent, respectively, of the total available bandwidth.

  - The expedited-forwarding (queue 1) and assured-forwarding (queue 2) classes have no scheduler because no resources are assigned to queue 1 or queue 2, by default. However, you can manually configure resources to be assigned to the expedited-forwarding and assured-forwarding classes.

Also by default, any queue can exceed the assigned bandwidth if additional bandwidth is available from other queues. When a forwarding class does not fully use the allocated transmission bandwidth, the remaining bandwidth can be used by other forwarding classes if they have a traffic load that exceeds their allocated bandwidth.

Excess Rate (EX4300 Switches)

> **NOTE**: Only EX4300 switches support the `excess-rate` option.

Excess rate traffic determines the percentage of the excess bandwidth to share when a queue receives traffic in excess of its bandwidth allocation. By default, the excess bandwidth is shared in the ratio of the

transmit rates. You can control this distribution by configuring the `excess-rate` statement at the `[edit class-of-service schedulers` *scheduler-name*`]` hierarchy. You can specify the excess rate sharing in percentage.

Transmission Rate

Transmission-rate control determines the actual traffic bandwidth for each forwarding class you configure. The transmission rate is specified in bits per second. Each queue is allocated some portion of the bandwidth of the interface. This bandwidth can be a fixed value, such as 1 megabit per second (Mbps), a percentage of the total available bandwidth, or the rest of the available bandwidth. In case of congestion, the configured transmission rate is guaranteed for the queue. Transmission-rate control allows you to ensure that each queue receives the bandwidth appropriate for its level of service.

Scheduler Buffer Size

To control congestion at the output stage, you can configure the delay-buffer bandwidth by using the *buffer-size (Schedulers) configuration statement*. The delay-buffer bandwidth provides packet buffer space to absorb burst traffic up to the specified duration of delay. When the specified delay buffer becomes full, packets with 100 percent drop probability are dropped from the tail of the buffer.

For each scheduler on EX Series switches, you can configure the buffer size as one of the following:

- The exact buffer size.

- A percentage of the total buffer.

- The remaining buffer available. The remainder is the buffer percentage that is not assigned to other queues. For example, if you assign 40 percent of the delay buffer to queue 0, allow queue 2 to keep the default allotment of 20 percent, allow queue 7 to keep the default allotment of 5 percent, and assign the remainder to queue 3, then queue 3 uses 35 percent of the delay buffer.

- On EX Series switches (other than Juniper Networks EX4300 Ethernet Switches):

  - The default scheduler transmission rates for queues 0 through 7 are 95, 0, 0, 0, 0, 0, 0, and 5 percent, respectively, of the total available bandwidth. The default buffer-size percentages for queues 0 through 7 are 95, 0, 0, 0, 0, 0, 0, and 5 percent, respectively, of the total available buffer.

- On EX4300 switches:

  - The default scheduler transmission rates for queues 0 through 11 are 75, 0, 0, 5, 0, 0, 0, 0, 15, 0, 0 and 5 percent, respectively, of the total available buffer. The default buffer-size percentages for queues 0 through 11 are 75, 0, 0, 5, 0, 0, 0, 0, 15, 0, 0 and 5 percent, respectively, of the total available buffer.

  - You can configure the buffer size as a temporal value on Juniper Networks EX4300 Ethernet Switches also.

Priority Scheduling

Priority scheduling determines the order in which an interface transmits traffic from queues, thus ensuring that queues containing important traffic are provided faster access.

Priority scheduling is accomplished through a procedure in which the scheduler examines the priority of the queue. EX Series switches support two levels of transmission priority:

- Low—The scheduler determines whether the individual queue is within its defined bandwidth profile or not. This binary decision, which is re-evaluated on a regular time cycle, involves comparing the amount of data transmitted by the queue against the bandwidth allocated to it by the scheduler. If the transmitted amount is less than the allocated amount, the queue is considered to be in profile. A queue is out of profile when the amount of traffic that it transmits is larger than the queue's allocated limit. An out-of-profile queue is transmitted only if bandwidth is available. Otherwise, it is buffered.

- Strict-high—A strict-high priority queue receives preferential treatment over a low-priority queue. Unlimited bandwidth is assigned to a strict-high priority queue.

- On EX Series switches (other than Juniper Networks EX4300 Ethernet Switches):

  - A queue from a set of queues is selected based on the shaped deficit weighted round robin (SDWRR) algorithm, which operates within the set.

  - Queues are scheduled according to the queue number, starting with the highest queue, 7, with decreasing priority down through queue 0. Traffic in higher-numbered queues is always scheduled prior to traffic in lower-numbered queues. In other words, if there are two high-priority queues, the queue with the higher queue number is processed first.

- On EX4300 switches:

  - The weighted deficit round-robin (WDRR) algorithm is used to select a queue from a set of queues.

  - You can configure multiple strict-high priority queues on an interface and an EX4300 switch processes these queues in a round-robin method.

Packets in low-priority queues are transmitted only when strict-high priority queues are empty.

Scheduler Drop-Profile Maps

Drop-profile maps associate drop profiles with a scheduler. A drop-profile map sets the drop profile for a specific packet loss priority (PLP) and protocol type. The inputs for a drop-profile map are the PLP and the protocol type. The output is the drop profile.

Scheduler Maps

A scheduler map associates a specified forwarding class with a scheduler configuration. After configuring a scheduler, you must include it in a scheduler map and then associate the scheduler map with an output interface.

On EX Series switches, if you configure more than the supported number of scheduler maps on a switch or for a port group in a line card, an error is logged in the system log. On any interface in a port group on a line card or on a switch, if you configure a scheduler map that causes the number of scheduler maps for that port group to exceed the maximum number supported, the default scheduler map is bound to that interface. We recommend that you check the system log for errors after the commit operation to verify that you have not configured more than the maximum permitted number of scheduler maps.

> (i) **NOTE**: On EX Series switches, you cannot configure a scheduler map on an individual interface that is a member of a link aggregation group (LAG). Instead, you must configure the scheduler map on the LAG itself (that is, on the aggregated Ethernet (**ae**) interface).

RELATED DOCUMENTATION

Defining CoS Schedulers and Scheduler Maps (CLI Procedure)

IN THIS SECTION

You use schedulers to define the class-of-service (CoS) properties of output queues. These properties include the amount of interface bandwidth assigned to the queue, the size of the memory buffer allocated for storing packets, the priority of the queue, and the drop profiles associated with the queue.

You associate the schedulers with forwarding classes by means of scheduler maps. You can then associate each scheduler map with an interface, thereby configuring the queues and packet schedulers that operate according to this mapping.

> **NOTE**: On EX Series switches, you cannot configure a scheduler map on an individual interface that is a member of a link aggregation group (LAG). Instead, you must configure the scheduler map on the LAG itself (that is, on the aggregated Ethernet (ae) interface).

You can associate up to four user-defined scheduler maps with an interface.

This topic describes:

Configuring a Scheduler and a Scheduler Map

You can define the properties for an output queue by configuring a scheduler. You can then define a scheduler map to associate a forwarding class with a scheduler.

To configure a scheduler and a scheduler map:

1. Create a scheduler, and assign one or more output queue properties to it:

```
[edit class-of-service]
user@switch# set schedulers scheduler-name output-queue-properties
```

For various properties that you can define for an output queue, see the *schedulers* hierarchy.

2. Configure a scheduler map that associates the scheduler with the forwarding class:

```
[edit class-of-service]
user@switch# set scheduler-maps map-name forwarding-class class-name scheduler scheduler-name
```

Assigning a Scheduler Map to Interfaces

After defining a scheduler map, you can assign the scheduler map to one or more interfaces. You can also assign the scheduler map to multiple interfaces by using a wildcard representation of the interface or Virtual Chassis Ports (VCPs).

Following are sample syntaxes and examples for assigning a scheduler map to a single or to multiple interfaces:

- To assign the scheduler map to one interface:

```
[edit class-of-service interfaces]
user@switch# set interface-name scheduler-map map-name
```

- To assign the scheduler map to more than one interface, you can use a wildcard representation of the interface:

```
[edit class-of-service interfaces]
user@switch# set wild-card-representation-of-interface-name scheduler-map map-name
```

For example, following is the configuration to assign the be-map scheduler map to all Gigabit Ethernet interfaces (ge-*):

```
[edit class-of-service interfaces]
user@switch# set ge-* scheduler-map be-map
```

- To assign the scheduler map to all VCPs:

```
[edit class-of-service interfaces]
user@switch# set wild-card-representation-of-vcp scheduler-map map-name
```

For example, following is the configuration to assign the be-map scheduler map to all VCPs:

```
[edit class-of-service interfaces]
user@switch# set vcp-* scheduler-map be-map
```

RELATED DOCUMENTATION

Defining CoS Schedulers (J-Web Procedure)

> **NOTE**: This topic applies only to the J-Web Application package.

You can use the J-Web interface to define CoS schedulers on an EX Series switch. Using schedulers, you can assign attributes to queues and thereby provide congestion control for a particular class of traffic. These attributes include the amount of interface bandwidth, memory buffer size, transmit rate, and schedule priority.

To configure schedulers:

1. Select **Configure** > **Class of Service** > **Schedulers**.

   > **NOTE**: After you make changes to the configuration on this page, you must commit the changes immediately for them to take effect. To commit all changes to the active configuration, select **Commit Options** > **Commit**. See Using the Commit Options to Commit Configuration Changes (J-Web Procedure) for details about all commit options.

2. Select one of the following options:

   - **Add**—Adds a scheduler. Enter information into the Schedulers page as described in Table 27 on page 132.

   - **Edit**—Modifies an existing scheduler. Enter information into the Schedulers page as described in Table 27 on page 132.

   - **Delete**—Deletes an existing scheduler.

Table 27: Schedulers Configuration Page

Field	Function	Your Action
Scheduler name	Specifies the name for a scheduler.	To name a scheduler, type the name—for example, **be-scheduler**.

Table 27: Schedulers Configuration Page *(Continued)*

Field	Function	Your Action
Scheduling priority	Sets the transmission priority of the scheduler, which determines the order in which an output interface transmits traffic from the queues. You can set the scheduling priority at different levels in the order of increasing priority from low to high. A high-priority queue with a high transmission rate might lock out lower-priority traffic.	To set a priority, select one: • **low**—Packets in this queue are transmitted last. • **strict-high**—Packets in this queue are transmitted first. To specify no scheduling priority, select the blank check box.

Table 27: Schedulers Configuration Page *(Continued)*

Field	Function	Your Action
Buffer size	Defines the size of the delay buffer. By default, queues 0 through 11 are allotted the following percentages of the total available buffer space: • Queue 0—75 percent • Queue 1—0 percent • Queue 2—0 percent • Queue 3—5 percent • Queue 4—0 percent • Queue 5—0 percent • Queue 6—0 percent • Queue 7—0 percent • Queue 8—15 percent • Queue 9—0 percent • Queue 10—0 percent • Queue 11—5 percent **NOTE**: A large buffer size value correlates with a greater possibility of packet delays. Such a value might not be practical for sensitive traffic such as voice or video.	To define a delay buffer size for a scheduler, select the appropriate option: • To specify no buffer size, select the blank check box. • To specify buffer size as a percentage of the total buffer, select **Percent** and type an integer from 1 through 100. • To specify buffer size as the remaining available buffer, select **Remainder**.

Table 27: Schedulers Configuration Page *(Continued)*

Field	Function	Your Action
Shaping rate	Specifies the rate at which queues transmit packets.	- To specify shaping rate as a percentage, select **Percent** and type an integer from 1 through 100. - To specify shaping rate as a number, select **Rate** and enter a value. - To specify no shaping rate, select the blank check box.

Table 27: Schedulers Configuration Page *(Continued)*

Field	Function	Your Action
Transmit rate	Defines the transmission rate of a scheduler. The transmit rate determines the traffic bandwidth from each forwarding class you configure. By default, queues 0 through 11 are allotted the following percentages of the transmission capacity: • Queue 0—75 percent • Queue 1—0 percent • Queue 2—0 percent • Queue 3—5 percent • Queue 4—0 percent • Queue 5—0 percent • Queue 6—0 percent • Queue 7—0 percent • Queue 8—15 percent • Queue 9—0 percent • Queue 10—0 percent • Queue 11—5 percent	To define a transmit rate, select the appropriate option: • To enforce the exact transmission rate, select **Rate** and enter a value. • To specify the remaining transmission capacity, select **Remainder Available**. • To specify a percentage of transmission capacity, select **Percent** and type an integer from 1 through 100. • To specify no transmit rate, select the blank check box.

RELATED DOCUMENTATION

Defining CoS Schedulers and Scheduler Maps (CLI Procedure) | **129**

Example: Configuring CoS on EX Series Switches | **11**

Monitoring CoS Scheduler Maps | **138**

Defining CoS Scheduler Maps (J-Web Procedure)

> **i** **NOTE**: This topic applies only to the J-Web Application package.

You can use the J-Web interface to configure CoS scheduler maps on an EX Series switch.

> **i** **NOTE**: On EX Series switches, you cannot configure a scheduler map on an individual interface that is a member of a link aggregation group (LAG). Instead, you must configure the scheduler map on the LAG itself—that is, on the aggregated Ethernet (**ae**) interface.

To configure scheduler maps:

1. Select **Configure** > **Class of Service** > **Scheduler Maps**.

   > **i** **NOTE**: After you make changes to the configuration on this page, you must commit the changes immediately for them to take effect. To commit all changes to the active configuration, select **Commit Options** > **Commit**. See Using the Commit Options to Commit Configuration Changes (J-Web Procedure) for details about all commit options.

2. Select one of the following options:

   - **Add**—Adds a scheduler map. Enter information into the scheduler map page as described in Table 28 on page 137.

   - **Edit**—Modifies an existing scheduler map. Enter information into the scheduler map page as described in Table 28 on page 137.

   - **Delete**—Deletes an existing scheduler map.

Table 28: Scheduler Maps Configuration Fields

Field	Function	Your Action
Scheduler Map Name	Specifies the name for a scheduler map.	To name a map, type the name—for example, **be-scheduler-map**.

Table 28: Scheduler Maps Configuration Fields *(Continued)*

Field	Function	Your Action
Scheduler Mapping	Enables you to associate a preconfigured scheduler with a forwarding class. After scheduler maps have been applied to an interface, they affect the hardware queues and packet schedulers.	To associate a scheduler with a forwarding class, locate the forwarding class and select the scheduler in the box next to it. For example, for the **best-effort** forwarding class, select the configured scheduler from the list.

RELATED DOCUMENTATION

Defining CoS Schedulers (J-Web Procedure) **|** **132**

Defining CoS Schedulers and Scheduler Maps (CLI Procedure) **|** **129**

Example: Configuring CoS on EX Series Switches **|** **11**

Monitoring CoS Scheduler Maps **|** **138**

Monitoring CoS Scheduler Maps

IN THIS SECTION

- Purpose **|** **138**
- Action **|** **139**
- Meaning **|** **139**

Purpose

> **(i)** **NOTE**: This topic applies only to the J-Web Application package.

Use the monitoring functionality to display assignments of CoS forwarding classes to schedulers.

Action

To monitor CoS scheduler maps in the J-Web interface, select **Monitor** > **Class of Service** > **Scheduler Maps**.

To monitor CoS scheduler maps in the CLI, enter the following CLI command:

```
show class-of-service scheduler-map
```

Meaning

Table 29 on page 139 summarizes key output fields for CoS scheduler maps.

Table 29: Summary of Key CoS Scheduler Maps Output Fields

Field	Values	Additional Information
Scheduler Map	Name of a scheduler map.	For details, click the plus sign (+).
Index	Index of a specific object—scheduler maps, schedulers, or drop profiles.	
Scheduler Name	Name of a scheduler.	
Forwarding Class	Forwarding classes this scheduler is assigned to.	
Transmit Rate	Configured transmit rate of the scheduler in bits per second (bps). The rate value can be either of the following: • A percentage—The scheduler receives the specified percentage of the total interface bandwidth. • **remainder**—The scheduler receives the remaining bandwidth of the interface after bandwidth allocation to other schedulers.	

Table 29: Summary of Key CoS Scheduler Maps Output Fields *(Continued)*

Field	Values	Additional Information
Buffer Size	Delay buffer size in the queue or the amount of transmit delay (in milliseconds). The buffer size can be either of the following: • A percentage—The buffer is a percentage of the total buffer allocation. • **remainder**—The buffer is sized according to what remains after other scheduler buffer allocations.	
Priority	Scheduling priority of a queue: • **strict-high**—Packets in this queue are transmitted first. • **low**—Packets in this queue are transmitted last.	
Excess rate	The percentage of excess bandwidth traffic to share.	
Drop Profiles	Name and index of a drop profile that is assigned to a specific loss priority and protocol pair.	
Loss Priority	Packet loss priority corresponding to a drop profile.	
Protocol	Transport protocol corresponding to a drop profile.	
Drop Profile Name	Name of the drop profile.	

Table 29: Summary of Key CoS Scheduler Maps Output Fields *(Continued)*

Field	Values	Additional Information
Index	Index of a specific object—scheduler maps, schedulers, or drop profiles.	

RELATED DOCUMENTATION

Defining CoS Schedulers and Scheduler Maps (CLI Procedure) | **129**

Defining CoS Schedulers (J-Web Procedure) | **132**

Example: Configuring CoS on EX Series Switches | **11**

2
PART

Congestion Management, Tail Drop Profiles, Queue Shaping, and Explicit Congestion Notification (ECN)

Congestion Management

Understanding CoS Congestion Management

A congestion in a network occurs because of various parameters and some packets must be dropped to avoid congestion and to facilitate easy flow of traffic in the network. On Juniper Networks EX Series Ethernet Switches, *class of service* (CoS) provides congestion management mechanisms for a switch to drop arriving packets based on certain parameters when a queue is full. Based on the EX Series switch that you are using, packets are dropped depending on either the priority of a packet or on both priority and drop probability of a packet.

You can specify parameters at the `[edit class-of-service drop-profiles]` hierarchy level for dropping packets and reference the parameters in a scheduler configuration.

Weighted Tail Drop Congestion Management

Weighted tail drop (WTD) is a congestion management mechanism to drop packets from the tail of the queue when the queue reaches a certain buffer capacity (that is, fill level). Hence the name *weighted tail drop*. The packets that are dropped are based on priority and are those marked with a packet loss priority (PLP) of *high*. You can configure a WTD profile (a WTD mechanism) usually on edge devices in a network.

When you configure a WTD profile, you are essentially setting the value for queue fullness. The queue fullness represents a percentage of the memory, known as delay-buffer bandwidth, that is used to store packets in relation to the total amount of memory that has been allocated for that specific queue. The delay-buffer bandwidth provides packet buffer space to absorb burst traffic up to the specified duration of delay. When the specified delay buffer becomes full, packets are dropped from the tail of the buffer.

By default, if you do not configure any drop profile, WTD profile is in effect and functions as the primary mechanism for managing congestion.

(i) **NOTE**: The default WTD profile associated with the packets whose PLP is *low* cannot be modified. You can configure custom drop profile only for those packets whose PLP is *high*.

Weighted Random Early Detection Congestion Management

In weighted random early detection (WRED) congestion management, random packets with a PLP of low or high are gradually dropped (based on drop probability) when the queue reaches a certain buffer capacity.

(i) **NOTE**: The WRED mechanism is supported only on Juniper Networks EX4300 standalone switches, EX4300 Virtual Chassis, EX4600 standalone switches, and EX9200 standalone switches.

Following are the different implementations of WRED:

- Segmented Drop Profile

- Interpolated Drop Profile

From a high level, segmented drop profile is a stair-step-like drop profile, whereas interpolated drop profile is a smother (curve) drop profile. Figure 3 on page 145 and Figure 4 on page 146 show a graphical representation of segmented and interpolated drop profiles. Regardless of the implementation, a drop profile represents a graph where the x-axis represents the percentage of fill level (l) and the y-axis represents the percentage of drop probability (p). The origin (0,0) represents the drop profile in which the drop probability is 0 percent when the queue fullness is 0 percent, and the point (100,100) represents that the drop probability is 100 percent when the queue fullness is 100 percent. Although the formation of graph lines in Figure 3 on page 145 and Figure 4 on page 146 is different, the application of the profile is the same. When a packet reaches the head of the queue, a random number between 0 and 100 is calculated. This random number is plotted against the drop profile graph using the current queue fullness of that particular queue. When the random number falls above the graph line, the packet is transmitted. When the number falls below the graph line, the packet is dropped from the network.

The following sections discuss the WRED drop profile implementations and parameters.

Segmented Drop Profile

In a segmented drop profile configuration, you can define multiple data points for fill level and drop probability. shows a graphical representation of a segmented drop profile.

Figure 3: Graphical Representation of a Segmented Drop Profile
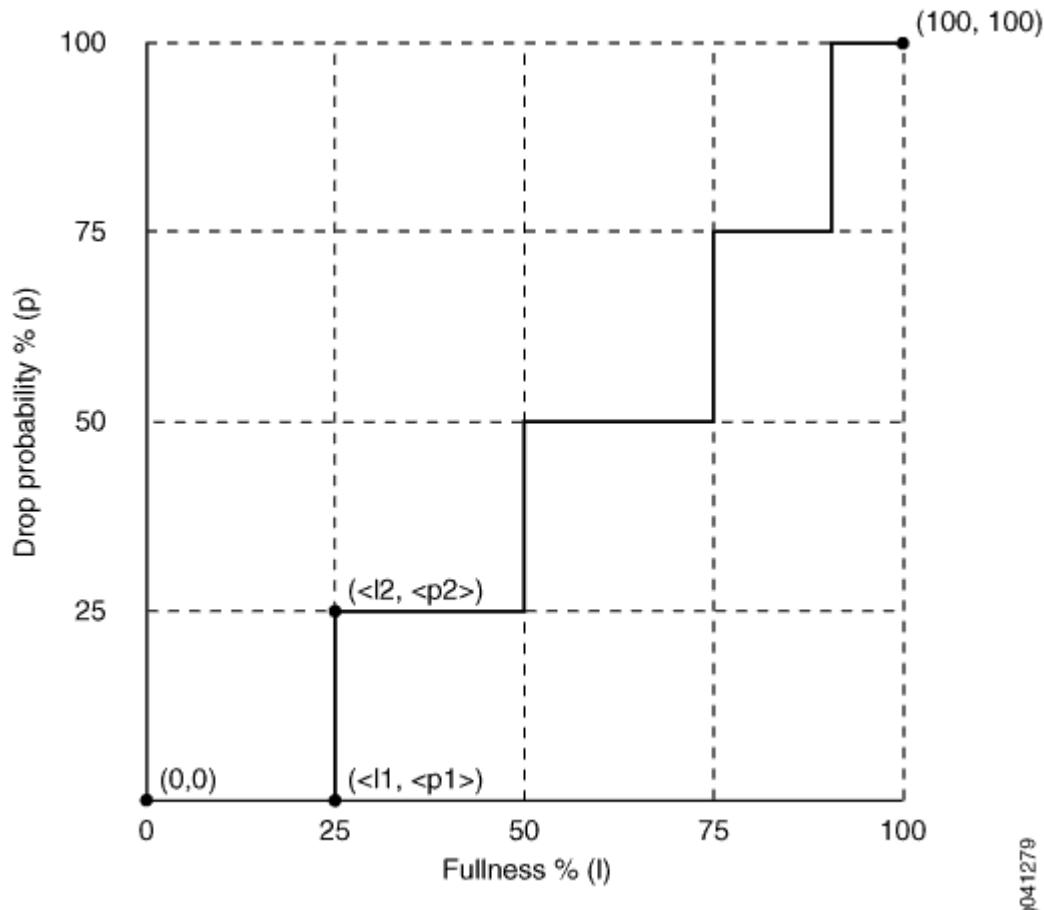
To create the profile's graph line, the software begins at the bottom-left corner of the graph, representing a 0 percent fill level and a 0 percent drop probability (that is the point (0,0)). The configuration draws a line directly to the right until it reaches the first defined fill level (that is, 25 percent represented in the graph on the x-axis). The software then continues the line vertically until the first drop probability is reached (that is, 25 percent represented in the graph in the y-axis). This process is repeated for all of the defined fill levels and drop probabilities until the top-right corner of the graph is reached (that is point (100,100) in the graph).

Interpolated Drop Profile

An interpolated drop profile configuration forms a smoother graph line compared to the graph in a segmented drop profile configuration. In this method of congestion management also, a switch uses multiple drop profile values to drop incoming packets to reduce congestion in the output queue.

Following are interpolated drop profile configurations on EX Series switches:

Interpolated Drop Profile Configuration on EX Series Switches Except EX4300 Switches

An interpolated drop profile on all EX Series switches except EX4300 switches automatically generates 64 pairs of data points on the graph beginning at (0, 0) and ending at (100, 100). Along the way, the graph line intersects specific data points that you define for fullness and drop probability.

Figure 4 on page 146 shows a graphical representation of an interpolated drop profile.

Figure 4: Graphical Representation of an Interpolated Drop Profile on EX Series Switches Except EX4300 Switches
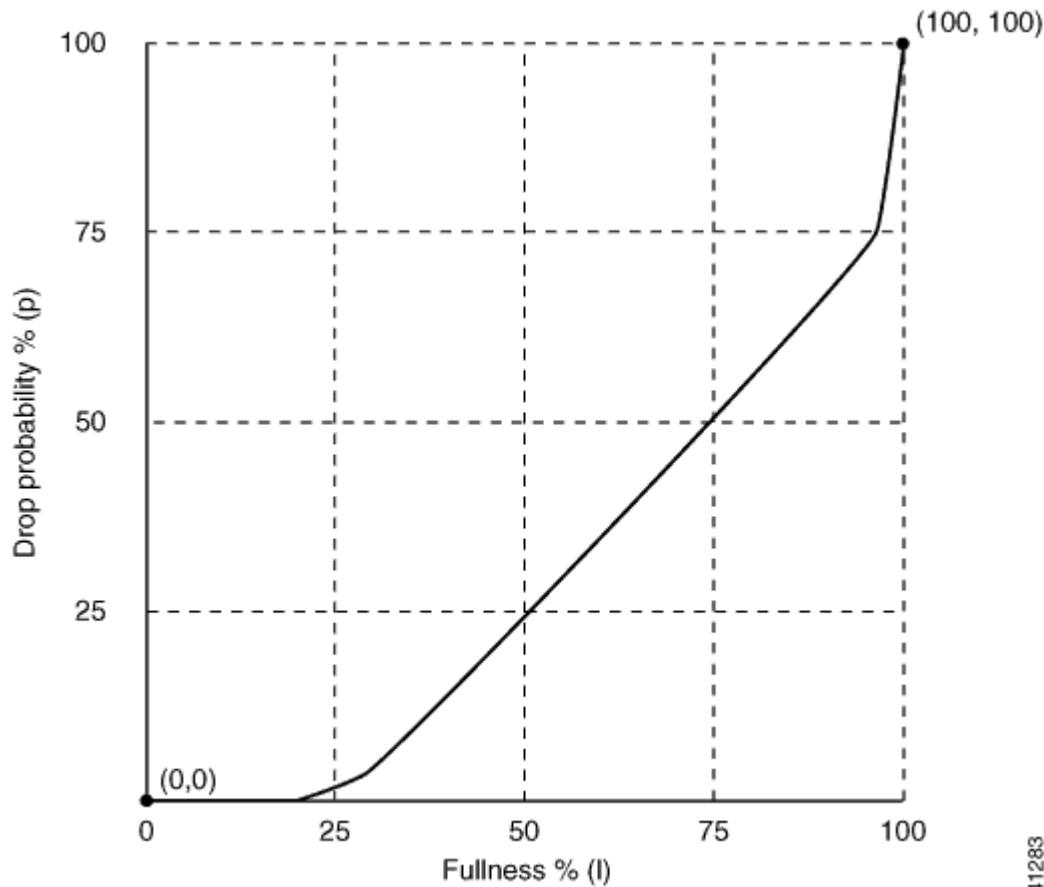
Interpolated Drop Profile Configuration on EX4300 Switches

On EX4300 switches, you can set two queue fill levels and two drop probabilities in each drop profile. The two fill levels and the two drop probabilities create two pairs of values. The first fill level and the first drop probability create one value pair. The second fill level and the second drop probability create the second value pair.

> **NOTE**: You can configure a maximum of 64 drop profiles on EX4300 switches.

The first fill level value specifies the percentage of queue fullness at which packets begin to drop, known as the drop start point. Until the queue reaches this level of fullness, no packets are dropped. The second fill level value specifies the percentage of queue fullness at which all packets are dropped, known as the drop endpoint.

The first drop probability value is always 0 (zero). This pairs with the drop start point and specifies that until the queue fullness level reaches the first fill level, no packets drop. When the queue fullness exceeds the drop start point, packets begin to drop until the queue exceeds the second fill level, when all packets drop. The second drop probability value, known as the maximum drop rate, specifies the likelihood of dropping packets when the queue fullness reaches the drop end point. As the queue fills from the drop start point to the drop end point, packets drop in a smooth, linear pattern (called an interpolated graph) as shown in Figure 5 on page 147. After the drop end point, all packets drop.

Figure 5: Tail-Drop Profile Packet Drop on EX4300 Switches
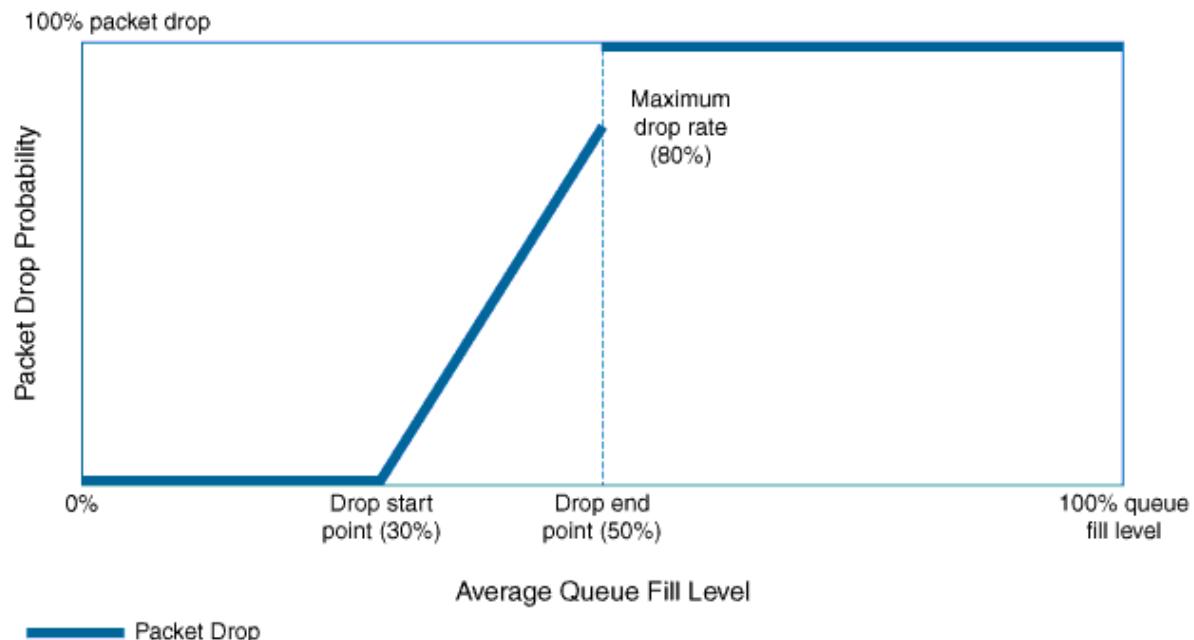
The thick line in Figure 5 on page 147 shows the packet drop characteristics for a sample tail drop profile. At the drop start point, the queue reaches a fill level of 30 percent. At the drop end point, the queue fill level reaches 50 percent, and the maximum drop rate is 80 percent.

No packets drop until the queue fill level reaches the drop start point of 30 percent. When the queue reaches the 30 percent fill level, packets begin to drop. As the queue fills, the percentage of packets dropped increases in a linear fashion. When the queue fills to the drop endpoint of 50 percent, the rate of packet drop has increased to the maximum drop rate of 80 percent. When the queue fill level exceeds the drop endpoint of 50 percent, all of the packets drop until the queue fill level drops below 50 percent.

Drop Profile Parameters

You can specify the following two values in drop profile configuration:

- Fill level—The queue fullness value, which represents a percentage of the memory used to store packets in relation to the total amount of memory allocated to the queue.

- Drop probability—The percentage value that corresponds to the likelihood that an individual packet is dropped.

Configuring CoS Congestion Management (CLI Procedure)

An effective congestion management mechanism is imperative to ensure smooth flow of traffic in a network and also to ensure minimum packet drops in the network. Class of service (CoS) provides congestion management methods that allow you to define parameters based on which packets can be dropped when the output queue is full. These parameters vary depending on the EX Series switch that you are using in a network.

You can specify parameters for dropping packets at the `[edit class-of-service drop-profiles]` hierarchy level and reference them in a scheduler configuration. The parameters that you can specify are `fill-level` and `drop-probability`. The first parameter defines the delay-buffer bandwidth, which provides packet buffer space to absorb burst traffic up to the specified duration of delay. When the specified delay buffer becomes full, packets with 100 percent drop probability are dropped from the head of the buffer. The second parameter represents a percentage value that correlates to the likelihood that an individual packet is dropped from the network.

Depending on the switch on which you are configuring a drop profile, you can configure either a weighted tail drop (WTD) profile or a weighted random early detection (WRED) profile.

Configuring a Weighted Tail Drop Profile

A weighted tail drop (WTD) is a congestion management mechanism in which packets are dropped from the tail of the queue when the queue reaches a certain buffer capacity (that is, the fill level), and hence the name weighted tail drop. When that level is reached, packets marked with a packet loss priority (PLP) of high are prevented from entering the queue (that is, they are discarded).

To configure a WTD profile, create a drop profile name and assign a fill level:

```
[edit class-of-service drop-profiles]
user@switch# set profile-name fill-level percentage
```

Following is a sample WTD profile in which the fill level is set to 80 percent:

```
[edit class-of-service drop-profiles]
user@switch# set wtd-profile fill-level 80
```

Configuring a Weighted Random Early Detection Drop Profile

A WRED drop profile enables you to define multiple data points for fill level and drop probability so that packets are dropped at various levels of queue fullness, and for various drop probabilities. Unlike the WTD drop profile that can be defined only for packets with a PLP of high, WRED can be defined for packets with a PLP of high and also for packets with a PLP of low.

> **NOTE**: The WRED drop profile is supported only on EX4300 standalone switches and EX4300 Virtual Chassis.

WRED has two implementations: segmented and interpolated. From a high level, segmented is a stair-step-like drop profile, whereas interpolated is a smother (curve) drop profile. For a graphical representation of both these implementations, see . Although the formation of graph lines is different for both these implementations, the

application of the profile is the same. On EX Series switches except EX4300 switches, when a packet reaches the head of the queue, a random number between 0 and 100 is calculated. This random number is plotted against the drop profile using the current queue fullness of that particular queue. When the random number falls above the graph line, the packet is transmitted. When the number falls below the graph line, the packet is dropped from the network.

For information about congestion management on EX4300 switches, see "Understanding CoS Congestion Management" on page 143.

> **NOTE**: On EX4300 switches, you cannot enable WRED on multidestination (multicast) queues. You can enable WRED only on unicast queues.

Following is the procedure to define a segmented and an interpolated drop profiles:

- To configure a segmented drop profile, specify multiple data points for fill level (l) and drop probability (p) as follows:

```
[edit class-of-service drop-profiles]
user@switch# set profile-name fill-level percentage-l1 drop-probability percentage-p1
user@switch# set profile-name fill-level percentage-l2 drop-probability percentage-p2
user@switch# set profile-name fill-level percentage-l3 drop-probability percentage-p3
user@switch# set profile-name fill-level percentage-l4 drop-probability percentage-p4
```

Following is a sample segmented drop profile:

```
[edit class-of-service drop-profiles]
user@switch# set seg-prof fill-level 20 drop-probability 25
user@switch# set seg-prof fill-level 40 drop-probability 50
user@switch# set seg-prof fill-level 60 drop-probability 75
user@switch# set seg-prof fill-level 80 drop-probability 100
```

- To configure an interpolated drop profile on EX Series switches except EX4300 switches, specify multiple data points for fill level (l) and drop probability (p) using the `interpolate` statement as follows:

```
[edit class-of-service drop-profiles ]
user@switch# set profile-name interpolate fill-level percentage-l1 drop-probability percentage-l1
user@switch# set profile-name interpolate fill-level percentage-l2 drop-probability percentage-l2
user@switch# set profile-name interpolate fill-level percentage-l3 drop-probability
```

```
percentage-p3
user@switch# set profile-name interpolate fill-level percentage-l4 drop-probability
percentage-p4
```

Following is a sample interpolated drop profile:

```
[edit class-of-service drop-profiles]
user@switch# set inter-prof interpolate fill-level 20 drop-probability 25
user@switch# set inter-prof interpolate fill-level 40 drop-probability 50
user@switch# set inter-prof interpolate fill-level 60 drop-probability 75
user@switch# set inter-prof interpolate fill-level 80 drop-probability 100
```

- To configure an interpolated drop profile EX4300 switches, specify two data points for fill level (l) and drop probability (p) by using the `interpolate` statement as follows:

```
[edit class-of-service drop-profiles ]
user@switch# set profile-name interpolate fill-level percentage-l1 fill-level percentage-l2
drop-probability percentage-l1 percentage-l2
```

Following is a sample interpolated drop profile:

```
[edit class-of-service drop-profiles]
user@switch# set inter-prof interpolate fill-level 20 fill-level 80 drop-probability 25 drop-
probability 100
```

RELATED DOCUMENTATION

Example: Configuring CoS on EX Series Switches | **11**

Understanding CoS Congestion Management | **143**

Tail Drop Profiles

IN THIS CHAPTER

Understanding CoS Tail Drop Profiles

Tail drop profile is a congestion management mechanism that allows switch to drop arriving packets when queue buffers become full or begin to overflow.

Tail drop profiles define the meanings of the loss priorities. When you configure tail drop profiles you are essentially setting the value for queue fullness. The queue fullness represents a percentage of the memory used to store packets in relation to the total amount that has been allocated for that specific queue.

The queue fullness defines the delay-buffer bandwidth, which provides packet buffer space to absorb burst traffic up to the specified duration of delay. Once the specified delay buffer becomes full, packets with 100 percent drop probability are dropped from the tail of the buffer.

You specify drop probabilities in the drop profile section of the CoS configuration hierarchy and reference them in each scheduler configuration.

By default, if you do not configure any drop profile, tail drop profile is in effect and functions as the primary mechanism for managing congestion. In the default tail drop profile, when the fill level is 0 percent, the drop probability is 0 percent. When the fill level is 100 percent, the drop probability is 100 percent.

> **NOTE**: The default drop profile associated with the packets whose loss priority is low cannot be modified. You can configure custom drop profile only for those packets whose loss priority is high.

Configuring CoS Tail Drop Profiles (CLI Procedure)

Tail drop is a simple and effective traffic congestion avoidance mechanism. When you apply this mechanism to manage congestion, packets are dropped when the output queue is full.

To configure CoS tail-drop profiles, create a drop profile name (be-dp) and assign a fill level (25):

```
[edit class-of-service drop-profiles]
user@switch# set be-dp fill-level 25
```

Defining CoS Drop Profiles (J-Web Procedure)

> (i) **NOTE**: This topic applies only to the J-Web Application package.

To configure CoS drop profiles:

1. Select **Configure** > **Class of Service** > **Drop Profile**.

   > (i) **NOTE**: After you make changes to the configuration on this page, you must commit the changes immediately for them to take effect. To commit all changes to the active configuration, select **Commit Options** > **Commit**. See Using the Commit Options to Commit Configuration Changes (J-Web Procedure) for details about all commit options.

2. Select one of the following options:

- **Add**—Adds a drop profile. Enter information into the drop profiles page as described in Table 30 on page 154.

- **Edit**—Modifies an existing drop file. Enter information into the drop profiles page as described in Table 30 on page 154.

- **Delete**—Deletes an existing drop profile.

Table 30: Drop Profiles Configuration parameters

Field	Function	Your Action
Drop Profile Name	Specifies the name for a drop profile.	Type the name.
Drop profile graph	Specifies the drop profile graph type	Select one: **Segmented** or **Interpolated**.
Drop profile values	Specifies values for the following two parameters of the drop profile: the queue fill level and the drop probability. The queue fill level represents a percentage of the memory used to store packets in relation to the total amount that has been allocated for that specific queue. The drop probability is a percentage value that correlates to the likelihood that an individual packet is dropped from the network.	To add new values: 1. Click **Add**. 2. Enter the fill level. 3. Enter the drop probability. 4. Click **OK**. To edit an existing value, click **Edit** and modify the fill level and drop probability. To delete a value, select it and click **Delete**.

RELATED DOCUMENTATION

Monitoring CoS Drop Profiles | **155**

Example: Configuring CoS on EX Series Switches | **11**

Monitoring CoS Drop Profiles

Purpose

> **(i)** **NOTE**: This topic applies only to the J-Web Application package.

Use the monitoring functionality to view data point information for each CoS random early detection (RED) drop profile.

Action

To monitor CoS RED drop profiles in the J-Web interface, select **Monitor** > **Class of Service** > **RED Drop Profiles**.

To monitor CoS RED drop profiles in the CLI, enter the following CLI command:

```
show class-of-service drop-profile
```

Meaning

summarizes the key output fields for CoS RED drop profiles.

Table 31: Summary of the Key Output Fields for CoS Red Drop Profiles

Field	Values	Additional Information
RED Drop Profile Name	Name of the RED drop profile. A drop profile consists of pairs of values between 0 and 100, one for queue buffer fill level and the other for drop probability, that determine the relationship between a buffer's fullness and the likelihood it will drop packets.	To display profile values, click the plus sign (+).
Graph RED Profile	Links to a graph of a RED curve that the system uses to determine the drop probability based on queue buffer fullness.	The x axis represents the queue buffer fill level, and the y axis represents the drop probability.
Type	Type of a specific drop profile: • **interpolated**—The two coordinates (x and y) of the graph are interpolated to produce a smooth profile. • **segmented**—The two coordinates (x and y) of the graph are represented by line fragments to produce a segmented profile.	
Index	Internal index of this drop profile.	
Fill Level	Percentage fullness of a buffer queue. This value is the x coordinate of the RED drop profile graph.	
Drop Probability	Drop probability of a packet corresponding to a specific queue buffer fill level. This value is the y coordinate of the RED drop profile graph.	

RELATED DOCUMENTATION

Defining CoS Drop Profiles (J-Web Procedure) **| 153**

Example: Configuring CoS on EX Series Switches **| 11**

Queue Shaping

IN THIS CHAPTER

Understanding Port Shaping and Queue Shaping for CoS

IN THIS SECTION

When the amount of traffic on a switch's network exceeds the maximum bandwidth, packets are lost because of congestion in the network. The excess traffic in the network must be handled carefully to ensure minimum or no data loss in the network. A class-of-service (CoS) configuration includes several parameters that classify traffic into different queues and also define packet loss priorities (PLPs) to ensure smooth transmission of data in the network. You can use these configuration parameters to control or shape traffic for a specific port on a switch or for a specific CoS queue. While port shaping defines the maximum bandwidth allocated to an interface, queue shaping defines a limit on excess-bandwidth usage for each queue.

Port Shaping

Port shaping enables you to shape the aggregate traffic through a port or channel to a rate that is less than the line rate. You can configure interfaces to shape traffic based on the rate-limited bandwidth of the total interface bandwidth. This allows you to shape the output of the interface so that the interface

transmits less traffic than it is capable of transmitting. For port shaping, you specify the shaping rate as the peak rate at which traffic can pass through the interface. You specify the rate as a value in bits per second (bps) either as a decimal number or as a decimal number followed by the abbreviation k (1000), m (1,000,000), or g (1,000,000,000) and the value can range from 1000 through 160,000,000,000 bps.

By default, shaping is not configured on an interface. If you do not configure a shaping rate on an interface, the default shaping rate is 100 percent, which is the equivalent to no shaping configured for that interface.

On modern EX series switches that support Enhanced Layer 2 Software (ELS), when you configure a shaping rate on an ae interface, the traffic is equally divided among the members of the ae interface. For example, consider an interface, ae0, that consists of three interfaces: ge-0/0/0, ge-0/0/1, and ge-0/0/2. If you configure a shaping rate of X Mpbs on ae0, traffic up to the rate of X/3 Mpbs flows through each of the three interfaces. This is known as scale mode.

> **NOTE**: On older EX swithes that don't support ELS, when you configure a shaping rate on an aggregated Ethernet (ae) interface, all members of the ae interface are shaped at the configured shaping rate. For example, consider an interface ae0 that consists of three interfaces: ge-0/0/0, ge-0/0/1, and ge-0/0/2. If you configure a shaping rate of X Mpbs on ae0, traffic up to the rate of X Mpbs flows through each of the three interfaces. Therefore, the total traffic flowing through ae0 can be at the rate of 3X Mbps. This is replicate mode.

Queue Shaping

Queue shaping throttles the rate at which queues transmit packets. For example, using queue shaping, you can rate-limit a strict-priority queue so that the strict-priority queue does not lock out (or starve) low-priority queues. Similarly, for any queue, you can configure queue shaping.

You can specify queue shaping as the maximum rate at which traffic can pass through the queue or as a percentage of the available bandwidth. On EX Series switches except EX4300 switches, you can specify the rate as a value between 3200 and 160,000,000,000 bps and the percentage as a value from 0 to 100 percent. On EX4300 switches, you can specify the rate as a value between 8000 and 160,000,000,000 bps and the percentage as a value from 0 to 100 percent.

RELATED DOCUMENTATION

Understanding CoS Schedulers | **125**

Defining CoS Schedulers and Scheduler Maps (CLI Procedure) | **129**

Configuring Shaping for CoS (CLI Procedure)

Port shaping and queue shaping enable you to limit traffic on an interface or queue, respectively, so that you can control the amount of traffic passing through the interface or the queue. Port shaping enables you to shape the aggregate traffic through an interface to a rate that is less than the line rate for that interface. When you configure port shaping on an interface, you are essentially specifying a value that indicates the maximum amount of traffic that can pass through the interface. This value must be less than the maximum bandwidth for that interface. Queue shaping enables you to throttle the rate at which a queue transmits packets. When you configure queue shaping, you can specify either as the maximum rate at which traffic can pass through the queue or as a percentage of the available bandwidth.

Configuring Port Shaping for CoS on an EX Series Switch

You can configure port shaping on network interfaces, aggregated Ethernet interfaces (also known as link aggregation groups (LAGs)), and loopback interfaces.

To configure port shaping on an interface:

1. Ensure that the interface on which you want to configure port shaping is up and running.
2. Assign a shaping rate for the interface:

```
[edit]
user@switch# set class-of-service interfaces interface-name shaping-rate value
```

The value indicates the maximum amount of traffic (in bps) that can pass through the interface. This value must be less than the maximum bandwidth for that interface.

Configuring Queue Shaping for CoS on an EX Series Switch

Queue shaping enables you to restrict the rate at which queues transmit traffic. You can configure queue shaping on any queue supported by CoS on an EX Series switch that supports up to eight output queues and 16 forwarding classes. Forwarding classes can be thought of as output queues. In effect, the result of classifying packets into forwarding classes is the identification of an output queue for a particular

packet. For a classifier to assign an output queue to a packet, it must associate the packet with one of the forwarding classes discussed in "Understanding CoS Forwarding Classes" on page 107.

To configure queue shaping:

1. Ensure that the interface on which you want to configure queue shaping is up and running.

2. Configure queue shaping:

   a. Define a scheduler and assign a shaping rate to the scheduler:

      ```
      [edit]
      user@switch# set class-of-service schedulers scheduler-name shaping-rate (rate | percent
      percentage)
      ```

      You can assign a *rate* (a value in bits per second (bps)) or a percentage value for shaping-rate.

   b. Define a scheduler map and assign a forwarding class and scheduler (that you defined in the previous step) to the scheduler map:

      ```
      [edit]
      user@switch# setset class-of-service scheduler-maps scheduler-map-name forwarding-class
      class-name scheduler scheduler-name
      ```

   c. Assign the scheduler map to an interface:

      ```
      [edit]
      user@switch# set class-of-service interfaces interface-name scheduler-map scheduler-map-
      name-map-name
      ```

RELATED DOCUMENTATION

Applying a Shaping Rate to Physical Interfaces Overview

On supported platforms, you can configure physical interfaces to shape traffic based on the rate-limited bandwidth of the total interface bandwidth. This allows you to shape the output of the physical interface, so that the interface transmits less traffic than it is physically capable of carrying.

If you do not configure a shaping rate on the physical interface, the default physical interface bandwidth is based on the channel bandwidth and the time slot allocation.

In general, the physical interface speed is the basis for calculating the various queue parameters for a physical interface such as delay buffer size, weighted round- robin (WRR) weight, drop profile, and so forth. However, when you apply a shaping rate by including the `shaping-rate` statement, the shaping rate on that physical interface becomes the basis for calculating all the queue parameters for that physical interface.

The minimum shaping rate value for the physical interface depends on the platform and line card. The maximum value of shaping rate is limited by the maximum transmission rate of the interface.

RELATED DOCUMENTATION

Configuring the Shaping Rate for Physical Interfaces

Configuring the Shaping Rate for Physical Interfaces

To configure the shaping rate on the physical interface, either include the `shaping-rate` statement at the `[edit class-of-service interfaces interface-name]` hierarchy level or include the `output-traffic-control-profile` statement at the `[edit class-of-service interfaces interface-name]` hierarchy level.

You can specify a peak bandwidth rate 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). For physical interfaces, the range is from 1000 through 6,400,000,000,000 bps.

The maximum value of `shaping-rate` is limited by the maximum transmission rate of the interface.

The following are two sample configurations for applying a shaping rate of 5 Gbps on a physical interface (xe-4/0/0):

Apply a shaping rate at the `[edit class-of-service interfaces interface-name]` hierarchy:

```
[edit class-of-service]
interfaces {
```

```
    xe-4/0/0 {
        shaping-rate 5g;
    }
}
```

Apply a shaping rate using a traffic control profile:

```
[edit class-of-service]
traffic-control-profiles {
    shaping-output {
        shaping-rate 5g;
    }

}
interfaces {
    xe-4/0/0 {
        output-traffic-control-profile shaping-output;
    }
}
```

To view the results of your configuration, issue the following show commands:

- **show class-of-service interface *interface-name***

- **show interfaces *interface-name* extensive**

RELATED DOCUMENTATION

Applying a Shaping Rate to Physical Interfaces Overview

Explicit Congestion Notification (ECN)

IN THIS CHAPTER

CoS Explicit Congestion Notification (ECN)

SUMMARY

Use ECN to improve transport efficiency between two IP endpoints by marking packets to signal congestion instead of dropping them.

IN THIS SECTION

Overview

Explicit congestion notification (ECN) enables end-to-end congestion notification between two endpoints on IP based networks. ECN improves transport efficiency between two IP endpoints by marking packets to signal congestion instead of dropping them for RED (random early detection). The two endpoints are an ECN-enabled sender and an ECN-enabled receiver. ECN must be enabled on both endpoints and on all of the intermediate devices between the endpoints for ECN to work properly. Any device in the transmission path that does not support ECN breaks the end-to-end ECN functionality.

> *(i)* **NOTE**: ECN marking happens at the IP header level. ECN works well with both TCP and UDP/ROCEv2 traffic.

ECN notifies networks about congestion with the goal of reducing packet loss and delay by making the sending device decrease the transmission rate until the congestion clears, without dropping packets. RFC 3168, *The Addition of Explicit Congestion Notification (ECN) to IP*, defines ECN.

ECN is disabled by default. You can enable ECN on any queue and the transport protocol chooses whether to use it. Normally, you enable ECN only on queues that handle best-effort traffic because other traffic types use different methods of congestion notification—lossless traffic uses priority-based flow control (PFC) and strict-high priority traffic receives all of the port bandwidth it requires up to the point of a configured maximum rate.

You enable ECN on individual output queues (as represented by forwarding classes) by enabling ECN in the queue scheduler configuration, mapping the scheduler to forwarding classes (queues), and then applying the scheduler to interfaces.

There are two types of ECN: Static ECN and Dynamic ECN (D-ECN). Static ECN requires you to manually configure the thresholds that trigger congestion notifications, and the thresholds stay the same until you change the setting. Dynamic ECN adjusts the thresholds automatically based on real-time conditions such as queue length and traffic patterns.

On supported devices, both ECN versions can be enabled on your device at the same time, but only one version can be assigned to a particular queue at a time.

> *(i)* **NOTE**: For ECN to work on a queue, you must also apply a weighted random early detection (WRED) packet drop profile to the queue.

How ECN Works

Without ECN, devices respond to network congestion by dropping IP packets. Dropped packets signal the network that congestion is occurring. Devices on the IP network respond to packet drops by reducing the packet transmission rate to allow the congestion to clear. However, the packet drop method of congestion notification and management has some disadvantages. For example, packets are dropped and must be retransmitted. Also, bursty traffic can cause the network to reduce the transmission rate too much, resulting in inefficient bandwidth utilization.

Instead of dropping packets to signal network congestion, ECN marks packets to signal network congestion, without dropping the packets. For ECN to work, all of the devices in the path between two ECN-enabled endpoints must have ECN enabled. ECN is negotiated during the establishment of the connection between the endpoints.

ECN-enabled devices determine the queue congestion state based on the WRED packet drop profile configuration applied to the queue, so each ECN-enabled queue must also have a WRED drop profile. If a queue fills to the level at which the WRED drop profile has a packet drop probability greater than zero (0), the device might mark a packet as experiencing congestion. Whether or not a device marks a packet as experiencing congestion is the same probability as the drop probability of the queue at that fill level.

ECN communicates whether or not congestion is experienced by marking the two least-significant bits in the differentiated services (DiffServ) field in the IP header. The most significant six bits in the DiffServ field contain the Differentiated Services Code Point (DSCP) bits. The state of the two ECN bits signals whether or not the packet is an ECN-capable packet and whether or not congestion has been experienced.

ECN-capable senders mark packets as ECN-capable. If a sender is not ECN-capable, it marks packets as not ECN-capable. If an ECN-capable packet experiences congestion at the egress queue of a device, the device marks the packet as experiencing congestion. When the packet reaches the ECN-capable receiver (destination endpoint), the receiver echoes the congestion indicator to the sender (source endpoint) by sending a packet marked to indicate congestion.

After receiving the congestion indicator from the receiver, the source endpoint reduces the transmission rate to relieve the congestion. This is similar to the result of TCP congestion notification and management, but instead of dropping the packet to signal network congestion, ECN marks the packet and the receiver echoes the congestion notification to the sender. Because the packet is not dropped, the packet does not need to be retransmitted.

ECN Bits in the DiffServ Field

The two ECN bits in the DiffServ field provide four codes that determine if a packet is marked as an ECN-capable transport (ECT) packet, meaning that both endpoints of the transport protocol are ECN-capable, and if there is congestion experienced (CE), as shown in the table below:

Table 32: ECN Bit Codes

ECN Bits (Code)	Meaning
00	Non-ECT—Packet is marked as not ECN-capable
01	ECT(1)—Endpoints of the transport protocol are ECN-capable
10	ECT(0)—Endpoints of the transport protocol are ECN-capable
11	CE—Congestion experienced

Codes 01 and 10 have the same meaning: the sending and receiving endpoints of the transport protocol are ECN-capable. There is no difference between these codes.

End-to-End ECN Behavior

After the sending and receiving endpoints negotiate ECN, the sending endpoint marks packets as ECN-capable by setting the DiffServ ECN field to ECT(1) (01) or ECT(0) (10). Every intermediate device between the endpoints must have ECN enabled or it does not work.

When a packet traverses a device and experiences congestion at an output queue that uses the WRED packet drop mechanism, the device marks the packet as experiencing congestion by setting the DiffServ ECN field to CE (11). Instead of dropping the packet (as with TCP congestion notification), the device forwards the packet.

> ⓘ **NOTE**: At the egress queue, the WRED algorithm determines whether or not a packet is drop eligible based on the queue fill level (how full the queue is). If a packet is drop eligible and marked as ECN-capable, the packet can be marked CE and forwarded. If a packet is drop eligible and is not marked as ECN-capable, it might be dropped. See "WRED Drop Profile Control of ECN Thresholds" on page 169 for more information about the WRED algorithm.

When the packet reaches the receiver endpoint, the CE mark tells the receiver that there is network congestion. The receiver then sends (echoes) a message to the sender that indicates there is congestion on the network. The sender acknowledges the congestion notification message and reduces its transmission rate. The diagram below summarizes how ECN works to mitigate network congestion:

Figure 6: Explicit Congestion Notification
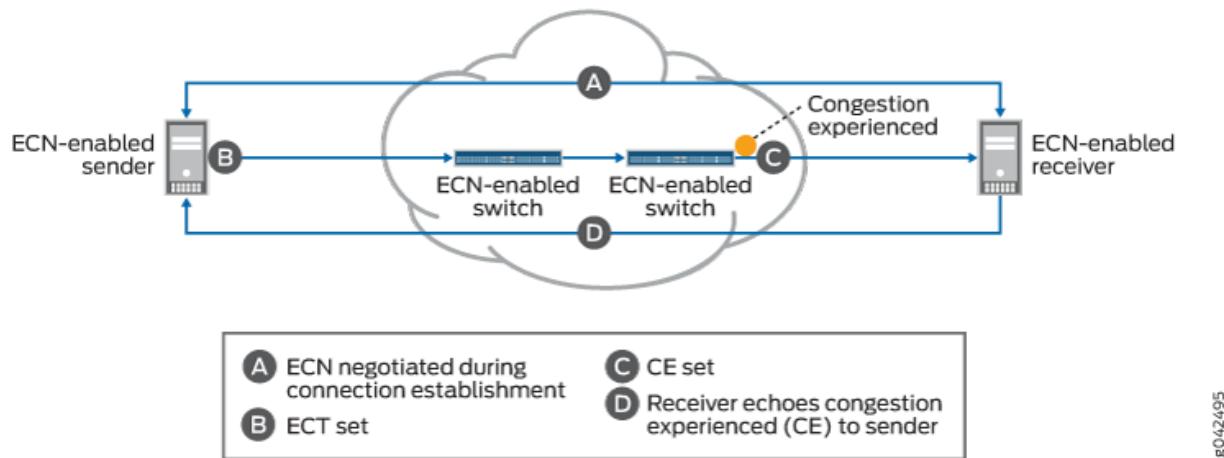
End-to-end ECN Behavior

End-to-end ECN behavior includes:

1. The ECN-capable sender and receiver negotiate ECN capability during the establishment of their connection.

2. After successful negotiation of ECN capability, the ECN-capable sender sends IP packets with the ECT field set to the receiver.

   > (*i*) **NOTE**: You must enable ECN on *all* of the intermediate devices in the path between the sender and the receiver.

3. If the WRED algorithm on a device egress queue determines that the queue is experiencing congestion and the packet is drop eligible, the device can mark the packet as "congestion experienced" (CE) to indicate to the receiver that there is congestion on the network. If the packet has already been marked CE (congestion has already been experienced at the egress of another device), the device forwards the packet with CE marked.

   If there is no congestion at the egress queue, the device forwards the packet and does not change the ECT-enabled marking of the ECN bits, so the packet is still marked as ECN-capable but not as experiencing congestion.

4. The receiver receives a packet marked CE to indicate that congestion was experienced along the congestion path.

5. The receiver echoes (sends) a packet back to the sender with the ECE bit (bit 9) marked in the flag field of the TCP/UDP header. The ECE bit is the ECN echo flag bit, which notifies the sender that there is congestion on the network.

6. The sender reduces the data transmission rate and sends a packet to the receiver with the CWR bit (bit 8) marked in the flag field of the TCP/UDP header. The CWR bit is the congestion window reduced flag bit, which acknowledges to the receiver that the congestion experienced notification was received.

7. When the receiver receives the CWR flag, the receiver stops setting the ECE bit in replies to the sender.

The table below summarizes the behavior of traffic on ECN-enabled queues.

Table 33: Traffic Behavior on ECN-Enabled Queues

Incoming IP Packet Marking of ECN Bits	ECN Configuration on the Output Queue	Action if WRED Algorithm Determines Packet is Drop Eligible	Outgoing Packet Marking of ECN Bits
Non-ECT (00)	Does not matter	Drop.	No ECN bits marked
ECT (10 or 01)	ECN disabled	Drop	Packet dropped—no ECN bits marked
ECT (10 or 01)	ECN enabled	Do not drop. Mark packet as experiencing congestion (CE, bits 11).	Packet marked ECT (11) to indicate congestion
CE (11)	ECN disabled	Drop	Packet dropped—no ECN bits marked
CE (11)	ECN enabled	Do not drop. Packet is already marked as experiencing congestion, forward packet without changing the ECN marking.	Packet marked ECT (11) to indicate congestion

When an output queue is not experiencing congestion as defined by the WRED drop profile mapped to the queue, all packets are forwarded, and no packets are dropped.

ECN Compared to PFC and Ethernet PAUSE

ECN is an end-to-end network congestion notification mechanism for IP traffic. Priority-based flow control (PFC) (IEEE 802.1Qbb) and Ethernet PAUSE (IEEE 802.3X) are different types of congestion management mechanisms.

ECN requires that an output queue must also have an associated WRED packet drop profile. Output queues used for traffic on which PFC is enabled should not have an associated WRED drop profile. Interfaces on which Ethernet PAUSE is enabled should not have an associated WRED drop profile.

PFC is a peer-to-peer flow control mechanism to support lossless traffic. PFC enables connected peer devices to pause flow transmission during periods of congestion. PFC enables you to pause traffic on a specified type of flow on a link instead of on all traffic on a link. For example, you can (and should) enable PFC on lossless traffic classes such as the `fcoe` forwarding class. Ethernet PAUSE is also a peer-to-peer flow control mechanism, but instead of pausing only specified traffic flows, Ethernet PAUSE pauses all traffic on a physical link.

With PFC and Ethernet PAUSE, the sending and receiving endpoints of a flow do not communicate congestion information to each other across the intermediate devices. Instead, PFC controls flows between two PFC-enabled peer devices (for example, devices) that support data center bridging (DCB) standards. PFC works by sending a pause message to the connected peer when the flow output queue becomes congested. Ethernet PAUSE simply pauses all traffic on a link during periods of congestion and does not require DCB.

PFC works this way: if a device output queue fills to a certain threshold, the device sends a PFC pause message to the connected peer device that is transmitting data. The pause message tells the transmitting device to pause transmission of the flow. When the congestion clears, the device sends another PFC message to tell the connected peer to resume transmission. (If the output queue of the transmitting device also reaches a certain threshold, that device can in turn send a PFC pause message to the connected peer that is transmitting to it. In this way, PFC can propagate a transmission pause back through the network.)

See *Understanding CoS Flow Control (Ethernet PAUSE and PFC)* for more information. You can also refer to *Understanding PFC Functionality Across Layer 3 Interfaces*.

WRED Drop Profile Control of ECN Thresholds

IN THIS SECTION

- How to Apply a WRED Drop Profile to an Output Queue: ETS Hierarchical Scheduling | **171**
- How to Apply a WRED Drop Profile to an Output Queue: Port Scheduling | **171**

You apply WRED drop profiles to forwarding classes (which are mapped to output queues) to control how the device marks ECN-capable packets. A scheduler map associates a drop profile with a scheduler and a forwarding class, and then you apply the scheduler map to interfaces to implement the scheduling properties for the forwarding class on those interfaces.

Drop profiles define queue fill level (the percentage of queue fullness) and drop probability (the percentage probability that a packet is dropped) pairs. When a queue fills to a specified level, traffic that matches the drop profile has the drop probability paired with that fill level. When you configure a drop profile, you configure pairs of fill levels and drop probabilities to control how packets drop at different levels of queue fullness.

The first fill level and drop probability pair is the drop start point. Until the queue reaches the first fill level, packets are not dropped. When the queue reaches the first fill level, packets that exceed the fill level have a probability of being dropped that equals the drop probability paired with the fill level.

The last fill level and drop probability pair is the drop end point. When the queue reaches the last fill level, all packets are dropped unless they are configured for ECN.

> **NOTE**: Lossless queues (forwarding class configured with the `no-loss` packet drop attribute) and strict-high priority queues do not use drop profiles. Lossless queues use PFC to control the flow of traffic. Strict-high priority queues receive all of the port bandwidth they require up to the configured maximum bandwidth limit.

Different devices support different amounts of fill level/drop probability pairs in drop profiles.

> **NOTE**: Do not configure the last fill level as 100 percent.

The drop profile configuration affects ECN packets as follows:

- Drop start point—ECN-capable packets might be marked as congestion experienced (CE).

- Drop end point—ECN-capable packets are always marked CE.

As a queue fills from the drop start point to the drop end point, the probability that an ECN packet is marked CE is the same as the probability that a non-ECN packet is dropped if you apply the drop profile to best-effort traffic. As the queue fills, the probability of an ECN packet being marked CE increases, just as the probability of a non-ECN packet being dropped increases when you apply the drop profile to best-effort traffic.

At the drop end point, all ECN packets are marked CE, but the ECN packets are not dropped. When the queue fill level exceeds the drop end point, all ECN packets are marked CE. At this point, all non-ECN packets are dropped. ECN packets (and all other packets) are tail-dropped if the queue fills completely.

How to Apply a WRED Drop Profile to an Output Queue: ETS Hierarchical Scheduling

To configure a WRED packet drop profile and apply it to an output queue (using hierarchical scheduling on devices that support ETS):

1. Configure a drop profile using the statement `set class-of-service drop-profiles` *`profile-name`* `interpolate fill-level` *`drop-start-point`* `fill-level` *`drop-end-point`* `drop-probability 0 drop-probability` *`percentage`*.

2. Map the drop profile to a queue scheduler using the statement `set class-of-service schedulers` *`scheduler-name`* `drop-profile-map loss-priority (low | medium-high | high) protocol any drop-profile` *`profile-name`*. The name of the drop-profile is the name of the WRED profile configured in Step 1.

3. Map the scheduler, which Step 2 associates with the drop profile, to the output queue using the statement `set class-of-service scheduler-maps` *`map-name`* `forwarding-class` *`forwarding-class-name`* `scheduler` *`scheduler-name`*. The forwarding class identifies the output queue. Forwarding classes are mapped to output queues by default, and can be remapped to different queues by explicit user configuration. The scheduler name is the scheduler configured in Step 2.

4. Associate the scheduler map with a traffic control profile using the statement `set class-of-service traffic-control-profiles` *`tcp-name`* `scheduler-map` *`map-name`*. The scheduler map name is the name configured in Step 3.

5. Associate the traffic control profile with an interface using the statement `set class-of-service interface` *`interface-name`* `forwarding-class-set` *`forwarding-class-set-name`* `output-traffic-control-profile` *`tcp-name`*. The output traffic control profile name is the name of the traffic control profile configured in Step 4.

   The interface uses the scheduler map in the traffic control profile to apply the drop profile (and other attributes, including the enable ECN attribute) to the output queue (forwarding class) on that interface. Because you can use different traffic control profiles to map different schedulers to different interfaces, the same queue number on different interfaces can handle traffic in different ways.

How to Apply a WRED Drop Profile to an Output Queue: Port Scheduling

You can configure a WRED packet drop profile and apply it to an output queue on devices that support port scheduling (ETS hierarchical scheduling is either not supported or not used). To configure a WRED packet drop profile and apply it to an output queue on devices that support port scheduling (ETS hierarchical scheduling is either not supported or not used):

1. Configure a drop profile using the statement `set class-of-service drop-profiles` *`profile-name`* `interpolate fill-level` *`level1 level2 ... level32`* `drop-probability` *`probability1 probability2 ... probability32`*. You can specify as few as two fill level/drop probability pairs or as many as 32 pairs.

2. Map the drop profile to a queue scheduler using the statement `set class-of-service schedulers` *scheduler-name* `drop-profile-map loss-priority (low | medium-high | high) drop-profile` *profile-name*. The name of the drop-profile is the name of the WRED profile configured in Step 1.

3. Map the scheduler, which Step 2 associates with the drop profile, to the output queue using the statement `set class-of-service scheduler-maps` *map-name* `forwarding-class` *forwarding-class-name* `scheduler` *scheduler-name*. The forwarding class identifies the output queue. Forwarding classes are mapped to output queues by default, and can be remapped to different queues by explicit user configuration. The scheduler name is the scheduler configured in Step 2.

4. Associate the scheduler map with an interface using the statement `set class-of-service interfaces` *interface-name* `scheduler-map` *scheduler-map-name*.

   The interface uses the scheduler map to apply the drop profile (and other attributes) to the output queue mapped to the forwarding class on that interface. Because you can use different scheduler maps on different interfaces, the same queue number on different interfaces can handle traffic in different ways.

Dynamic ECN

IN THIS SECTION

- Support, Limitations, and Notes | **172**

Dynamic ECN enhances the ECN feature set by providing a way to automate the thresholds triggering a congestion notification event. Junos OS Evolved monitors real-time conditions like queue length and traffic patterns to evaluate whether or not a threshold should be adjusted. This results in a faster response to congestion events than static ECN, and improves congestion control efficiency.

D-ECN is more difficult to implement than static ECN, and requires active monitoring. You should evaluate your network conditions and configuration to decide if D-ECN is the best fit for your network.

You can check Feature Explorer to see if your device supports D-ECN.

Support, Limitations, and Notes

If the WRED algorithm that is mapped to a queue does not find a packet drop eligible, then the ECN configuration and ECN bits marking does not matter. The packet transport behavior is the same as when ECN is not enabled.

ECN is disabled by default. Normally, you enable ECN only on queues that handle best-effort traffic, and you do not enable ECN on queues that handle lossless traffic or strict-high priority traffic.

ECN supports the following:

- IPv4 and IPv6 packets

- Untagged, single-tagged, and double-tagged packets

- The outer IP header of IP tunneled packets (but not the inner IP header)

ECN does not support the following:

- IP packets with MPLS encapsulation

- The inner IP header of IP tunneled packets (however, ECN works on the outer IP header)

- Multicast, broadcast, and destination lookup fail (DLF) traffic

- Non-IP traffic

> **(i)** **NOTE**: To apply a WRED drop profile to non-ECT traffic, configure a multifield (MF) classifier to assign non-ECT traffic to a different output queue that is not ECN-enabled, and then apply the WRED drop profile to that queue.

ECN Packets per Queue

IN THIS SECTION

Overview

IN THIS SECTION

Explicit congestion notification (ECN) enables two endpoint devices on TCP/IP-based networks to send end-to-end congestion notifications to each other. Without ECN, devices respond to network congestion by dropping TCP/IP packets. The dropped packets signal the occurrence of network

congestion. In contrast, ECN marks packets to signal network congestion without dropping the packets. ECN reduces packet loss by making the sending device decrease the transmission rate until the congestion clears.

Packets may be delayed as the device decreases the transmission rate until congestion clears. To account for how many packets are delayed, you can use the `show interfaces queue` command to view the amount of ECN congestion experienced (CE) traffic in the queue.

Benefits

- Identify the packets that have experienced congestion.

- Helps in identifying if traffic is going to reach the queue buffer limits.

- Enables quick troubleshooting of network congestion points.

Configuration

1. Configure ECN.

   ECN is disabled by default. For how to configure ECN, see Example: Configuring Static and Dynamic ECN.

2. Enable ECN on both endpoints and on all of the intermediate devices between the endpoints.

   ECN must be enabled this way for ECN to work properly. Any device in the transmission path that does not support ECN breaks the end-to-end ECN functionality.

3. Use the `show interfaces queue` command to view the amount of traffic that has experienced congestion.

   The `ECN-CE packets` field shows the number of packets that have experienced congestion, while the `ECN-CE bytes` field shows the number of total bytes in those packets.

   The per-queue ECN counters `ECN-CE packets` and `ECN-CE bytes` only count packets that experienced congestion on the local switch.

   For example:

```
show interfaces queue et-0/0/5 forwarding-class network-control

Physical interface: et-0/0/5, up, Physical link is Up
  Interface index: 1262, SNMP ifIndex: 974
Forwarding classes: 12 supported, 9 in use
Egress queues: 12 supported, 9 in use
Queue: 3, Forwarding classes: network-control1
  Queued:
    Packets              :             15239998                856158 pps
    Bytes                :           2225039708             999992904 bps
```

```
Transmitted:
  Packets             :          15239998            856158 pps
  Bytes               :        2225039708         999992904 bps
   Tail-dropped packets :                0                 0 pps
   Tail-dropped bytes  :                0                 0 bps
   RED-dropped packets  :                0                 0 pps
   RED-dropped bytes   :                0                 0 bps
   ECN-CE packets      :          8577686            482043 pps
   ECN-CE bytes        :        1252342156          70378315 bps
```

On-Chip Buffer with ECN on PTX10002-36QDD Routers

PTX10002-36QDD routers support fine-grained control over buffer allocation and congestion management. This allows you to configure buffers in such a way that they remain "on-chip"; that is, buffered packets remain on the main traffic chip rather than be stored on a separate buffering chip. On-chip buffering is optimized for low-latency, high-throughput scenarios.

VOQs with a buffer size of 40µs, at the configured target rate, will be always on-chip. On-chip VOQs do not participate in on/off chip decisions.

> (i) **NOTE**: The default buffer size value for unconfigured queues is 1 byte, which makes the corresponding VOQ also on-chip.

When you have a VOQ with an OCB and then configure an ECN profile for the queue, after 100% ECN marking probability is reached at the defined queue occupancy, the queue stays on-chip.

As an example, shown below, say you want to mark traffic after a queue grows to 8µs. Assuming the RTT over the router fabric is 6µs, configure an ECN profile with 100% drop probability at a fill level of 35% (35% of 40µs is 14us = 8µs buffer size + 6µs RTT). The queue stays on chip at 14µs occupancy, and the network will have 36µs to react before the queue fills up and starts tail-dropping.

```
[edit class-of-service]
set drop-profile dp1 fill-level 35 drop-probability 100
set schedulers s1 explicit-congestion-notification
set schedulers s1 drop-profile-map loss-priority low protocol any drop-profile dp1
set schedulers s1 buffer-size temporal 40
set schedulers s1 priority low
set scheduler-maps map1 forwarding-class fc1 scheduler s1
set interfaces xe-0/0/1 scheduler-map map1
```

Platform-Specific Behavior

Use Feature Explorer to confirm platform and release support for ECN.

Use the following table to review platform-specific behaviors for this feature.

Platform	Difference
PTX10000 Series	- You can implement low-threshold ECN (begin ECN marking as soon as the buffer starts filling up) by defining a buffer rate for a scheduler and a low percentage drop profile. - The buffer rate acts as the base rate for buffer size calculation. The buffer rate is the target rate of a VOQ, which is the intended egress queue rate during typical congestion. Set the `buffer-rate` at the `[edit class-of-service schedulers scheduler-name]` hierarchy level. - You can also define more granular (to the tenths of a percent) and smaller fill-level percentages for drop profiles. That is, you can set a fill-level as low as 0.7 percent. **NOTE**: PTX routers support only Static ECN. - If both PFC and ECN are enabled, when the occupancy of a PFC account is above XON, by default ECN-capable packets are marked as congestion experienced (CE). - On PTX10002-36QDD routers, virtual output queues (VOQs) with a buffer size equal to or less than 40 microseconds, at the configured buffer rate, are always on-chip. When you have a VOQ with an OCB and then configure an ECN profile for the queue, the queue remains on-chip after the system reaches 100% ECN marking probability at the defined queue occupancy. Keeping the queue on-chip eliminates the need to access slower external memory. This approach is ideal for low-latency, high-throughput applications.

(Continued)

Platform	Difference
QFX5000 Series	On QFX5K platforms, ECN functionality is tightly integrated with WRED thresholds. WRED thresholds are static, so ECN also works based on static calculations of buffer thresholds. However, actual shared buffer usage of queues is dynamic. Following are the formulas used for ECN marking threshold calculations at the time of ECN configuration. • `max buffer access eligibility for ECN enabled queue =` `(( shared pool size * hardware_alpha)/(1 + hardware_alpha)) +` `egress queue dedicated buffer` • `ECN marking start threshold = WRED start fill level percent *` `Max buffer access eligibility for ECN enabled queue` • `ECN 100% marking threshold = WRED end fill level percent * Max` `buffer access eligibility for ECN enabled queue` During congestion for ECN capable packets, ECN CE marking starts after `ECN marking start threshold` is reached. ECN capable packets are probabilistically ECN CE marked until `ECN 100% marking threshold` is reached. After this threshold, all ECN capable packets are ECN CE marked until the queue reaches `max buffer access eligibility for ECN enabled queue`. After this threshold, tail drops occur. In the above calculation of `max buffer access eligibility for ECN enabled queue`, the best case scenario of a single queue alone competing for shared buffer space is assumed. However, the actual shared buffer use for a congested queue can decrease dynamically based on the number of competing queues for the shared buffer at any point of time. Following is the formula to calculate the `actual dynamic max buffer usage per queue`. • `actual max buffer usage for ECN enabled queue = (shared pool` `size * hw_alpha) / (1 + (hw_alpha * number of competing` `queues)) + egress queue dedicated buffer + ingress Pg` `dedicated buffer by the traffic flow` Two parameters, `number of competing queues` and `ingress Pg dedicated buffer by the traffic flow`, used in `actual dynamic max buffer usage per queue` calculation cannot be considered while

(Continued)

Platform	Difference
	calculating the ECN threshold as both these parameters are dynamic in nature. This creates a possibility that actual max buffer usage for an ECN enabled queue can be below the calculated static ECN marking thresholds. Therefore, with certain shared buffer and WRED fill level configurations, there is a possibility of packet tail drops due to shared buffer exhaustion occurring even before ECN marking on ECN enabled lossy queues. For lossless queues, due to the above limitation, PFC can start from an ingress port before ECN marking, as the PFC XOFF threshold is dynamic, unlike the static ECN threshold. You can determine proper ECN marking thresholds by monitoring the peak buffer usage of congested queues and fine tuning the ECN/WRED thresholds accordingly.
QFX10000 Series	• On QFX10000 switches, when you enable a queue for ECN and apply a WRED drop profile to the queue, the WRED drop profile only sets the thresholds for marking ECN traffic as experiencing congestion (CE, 11). On ECN-enabled queues, the WRED drop profile does not set drop thresholds for non-ECT (00) traffic (traffic that is not ECN-capable). Instead, the switch uses the tail-drop algorithm on traffic is that is marked non-ECT on ECN-enabled queues during periods of congestion.
SRX Series	• Explicit congestion notification (ECN) enables end-to-end congestion notification between two endpoints on TCP/IP based networks. The two endpoints are an ECN-enabled sender and an ECN-enabled receiver. ECN must be enabled on both endpoints. However, in the case of an unsupported peer, an SRX Series Firewall that supports ECN bootstraps the incoming packets from the unsupported peer and marks the packets to signal network congestion when it occurs. • If the client is not ECN capable, then the SRX firewall negotiates ECN on behalf of client during the connection establishment. The SRX firewall sets the ECE and CWR bits in the TCP header of the SYN packet.

Example: Configuring Static and Dynamic ECN

IN THIS SECTION

This example shows how to enable explicit congestion notification (ECN) on an output queue.

Requirements

This example uses the following hardware and software components:

- One device that supports ECN.

> (i) **NOTE**: Use Feature Explorer to confirm platform and release support for static and dynamic ECN.

Overview

ECN enables end-to-end congestion notification between two endpoints on TCP/IP based networks. The two endpoints are an ECN-enabled sender and an ECN-enabled receiver. ECN must be enabled on both endpoints and on all of the intermediate devices between the endpoints for ECN to work properly. Any device in the transmission path that does not support ECN breaks the end-to-end ECN functionality

A weighted random early detection (WRED) packet drop profile must be applied to the output queues on which ECN is enabled. ECN uses the WRED drop profile thresholds to mark packets when the output queue experiences congestion.

ECN reduces packet loss by forwarding ECN-capable packets during periods of network congestion instead of dropping those packets. (TCP notifies the network about congestion by dropping packets.) During periods of congestion, ECN marks ECN-capable packets that egress from congested queues.

When the receiver receives an ECN packet that is marked as experiencing congestion, the receiver echoes the congestion state back to the sender. The sender then reduces its transmission rate to clear the congestion.

ECN is disabled by default. You can enable ECN on best-effort traffic. ECN should not be enabled on lossless traffic queues, which uses priority-based flow control (PFC) for congestion notification, and ECN should not be enabled on strict-high priority traffic queues.

There are two types of ECN available: Static ECN and Dynamic ECN. Static ECN requires you to manually set the threshold levels that trigger a notification event. Dynamic ECN automatically adjusts the thresholds based on real-time conditions like queue length and traffic patterns.

To enable ECN on an output queue, you not only need to enable ECN in the queue scheduler, you also need to:

- Configure a WRED packet drop profile.

- Configure a queue scheduler that includes the WRED drop profile and enables ECN. (This example shows only ECN and drop profile configuration; you can also configure bandwidth, priority, and buffer settings in a scheduler.)

- Map the queue scheduler to a forwarding class (output queue) in a scheduler map.

- If you are using enhanced transmission selection (ETS) hierarchical scheduling, add the forwarding class to a forwarding class set (priority group).

- If you are using ETS, associate the queue scheduler map with a traffic control profile (priority group scheduler for hierarchical scheduling).

- If you are using ETS, apply the traffic control profile and the forwarding class set to an interface. On that interface, the output queue uses the scheduler mapped to the forwarding class, as specified by the scheduler map attached to the traffic control profile. This enables ECN on the queue and applies the WRED drop profile to the queue.

  If you are using port scheduling, apply the scheduler map to an interface. On that interface, the output queue uses the scheduler mapped to the forwarding class in the scheduler map, which enables ECN on the queue and applies the WRED drop profile to the queue.

Table 34 on page 181 shows the configuration components for this example.

Table 34: Components of the ECN Configuration Example

Component	Settings
Drop profile (with two fill level/drop probability pairs)	Name: `be-dp` Drop start fill level: `30` percent Drop end fill level: `75` percent Drop probability at drop start (minimum drop rate): `0` percent Drop probability at drop end (maximum drop rate): `80` percent
Scheduler	Name: `be-sched` ECN: enabled Drop profile: `be-dp` Transmit rate: `25%` Buffer size: `25%` Priority: `low`
Scheduler map	Name: `be-map` Forwarding class: `best-effort` Scheduler: `be-sched` **NOTE**: By default, the `best-effort` forwarding class is mapped to output queue `0`.
Forwarding class set (ETS only)	Name: `be-pg` Forwarding class: `best-effort` (queue 0)
Traffic control profile (ETS only)	Name: `be-tcp` Scheduler map: `be-map`
Interface (ETS only)	Name: `xe-0/0/20` Forwarding class set: `be-pg` (Output) traffic control profile: `be-tcp`
Interface (port scheduling only)	Name: `xe-0/0/20` Scheduler map: `be-map`

> **ⓘ**   **NOTE**: Only devices that support ETS hierarchical scheduling support forwarding class set and traffic control profile configuration. Direct port scheduling does not use the hierarchical scheduling structure.

Configuration

IN THIS SECTION

CLI Quick Configuration

To quickly configure the drop profile, scheduler with ECN enabled, and to map the scheduler to an output queue on an interface, copy the following commands, paste them in a text file, remove line breaks, change variables and details to match your network configuration, and then copy and paste the commands into the CLI at the `[edit]` hierarchy level.

ETS Quick Configuration

```
[edit class-of-service]
set drop-profile be-dp interpolate fill-level 30 fill-level 75 drop-probability 0 drop-
probability 80
set schedulers be-sched explicit-congestion-notification
set schedulers be-sched drop-profile-map loss-priority low protocol any drop-profile be-dp
set schedulers be-sched transmit-rate percent 25
set schedulers be-sched buffer-size percent 25
set schedulers be-sched priority low
set scheduler-maps be-map forwarding-class best-effort scheduler be-sched
set forwarding-class-sets be-pg class best-effort
set traffic-control-profiles be-tcp scheduler-map be-map
set interfaces xe-0/0/20 forwarding-class-set be-pg output-traffic-control-profile be-tcp
```

Port Scheduling Quick Configuration

```
[edit class-of-service]
set drop-profile be-dp interpolate fill-level 30 fill-level 75 drop-probability 0 drop-
probability 80
set schedulers be-sched explicit-congestion-notification
set schedulers be-sched drop-profile-map loss-priority low protocol any drop-profile be-dp
set schedulers be-sched transmit-rate percent 25
set schedulers be-sched buffer-size percent 25
set schedulers be-sched priority low
set scheduler-maps be-map forwarding-class best-effort scheduler be-sched
set interfaces xe-0/0/20 scheduler-map be-map
```

Configuring Static ECN

Step-by-Step Procedure

To configure Static ECN:

1. Configure the WRED packet drop profile `be-dp`. This example uses a drop start point of `30` percent, a drop end point of `75` percent, a minimum drop rate of `0` percent, and a maximum drop rate of `80` percent:

   ```
   [edit class-of-service]
   user@switch# set drop-profile be-dp interpolate fill-level 30 fill-level 75 drop-probability
   0 drop-probability 80
   ```

2. Create the scheduler `be-sched` with ECN enabled and associate the drop profile `be-dp` with the scheduler:

   ```
   [edit class-of-service]
   user@switch# set schedulers be-sched explicit-congestion-notification
   user@switch# set schedulers be-sched drop-profile-map loss-priority low protocol any drop-
   profile be-dp
   user@switch# set be-sched transmit-rate percent 25
   user be-sched transmit-rate percent 25
   user@switch# set be-sched buffer-size percent 25
   user@switch# set be-sched buffer-size percent 25
   user@switch# set be-sched priority low
   ```

3. Map the scheduler `be-sched` to the `best-effort` forwarding class (output queue 0) using scheduler map `be-map`:

```
[edit class-of-service]
user@switch# set scheduler-maps be-map forwarding-class best-effort scheduler be-sched
```

4. If you are using ETS, add the forwarding class `best-effort` to the forwarding class set `be-pg`; if you are using direct port scheduling, skip this step:

```
[edit class-of-service]
user@switch# set forwarding-class-sets be-pg class best-effort
```

5. If you are using ETS, associate the scheduler map `be-map`with the traffic control profile `be-tcp`; if you are using direct port scheduling, skip this step:

```
[edit class-of-service]
user@switch# set traffic-control-profiles be-tcp scheduler-map be-map
```

6. If you are using ETS, associate the traffic control profile `be-tcp` and the forwarding class set `be-pg` with the interface on which you want to enable ECN on the best-effort queue:

```
[edit class-of-service]
user@switch# set interfaces xe-0/0/20 forwarding-class-set be-pg output-traffic-control-
profile be-tcp
```

If you are using direct port scheduling, associate the scheduler map `be-map` with the interface on which you want to enable ECN on the best-effort queue:

```
[edit class-of-service]
user@switch# set interfaces xe-0/0/20 scheduler-map be-map
```

Configuring Dynamic ECN

Step-by-Step Procedure

To configure Dynamic ECN:

1. Follow the steps to configure Static ECN.

2. Enable the Dynamic ECN profile:

```
[edit class-of-service]
user@switch# set decn-profile profile-name queue queue number floor floor value offset offset
value
```

Floor and offset values range from 0 to 524287.

Floor and offset values do not need to be configured. If no value is selected, the maximum value of 524287 will be selected by default.

3. Assign the Dynamic ECN profile to an interface:

```
[edit class-of-service]
user@switch# set interfaces interface decn-profile profile-name
```

Verification

IN THIS SECTION

- Verifying That ECN Is Enabled | **185**

Verifying That ECN Is Enabled

Purpose

Verify that ECN is enabled in the scheduler be-sched by showing the configuration for the scheduler map be-map.

Action

Display the scheduler map configuration using the operational mode command show class-of-service scheduler-map be-map:

```
user@switch> show class-of-service scheduler-map be-map
Scheduler map: be-map, Index: 12240
```

```
   Scheduler:be-sched, Forwarding class: best-effort, Index: 115
     Transmit rate: 25 percent, Rate Limit: none, Buffer size: 25 percent,
     Buffer Limit: none, Priority: low
     Excess Priority: unspecified, Explicit Congestion Notification: enable
     ECN Type: static/dynamic Offset offset-value Floor floor-value
     Drop profiles:
       Loss priority   Protocol    Index   Name
       Low             any         3312    be-dp
       Medium-high     any         1       <default-drop-profile>
       High            any         1       <default-drop-profile>
```

Meaning

The `show class-of-service scheduler-map` operational command shows the configuration of the scheduler associated with the scheduler map and the forwarding class mapped to that scheduler. The output shows that:

- The scheduler associated with the scheduler map is `be-sched`.

- The scheduler map applies to the forwarding class `best-effort` (output queue 0).

- The scheduler `be-sched` has a transmit rate of `25` percent, a queue buffer size of `25` percent, and a drop priority of `low`.

- Explicit congestion notification state is `enable`.

- The WRED drop profile used for low drop priority traffic is `be-dp`.

RELATED DOCUMENTATION

CoS Explicit Congestion Notification (ECN)

3
PART

CoS on Overlay Networks

CoS on MPLS Networks

IN THIS CHAPTER

Understanding Using CoS with MPLS Networks on EX Series Switches

IN THIS SECTION

You can use *class of service* (CoS) within MPLS networks to prioritize certain types of traffic during periods of congestion. See EX Series Switch Software Features Overview for a complete list of the Junos OS MPLS features that are supported on specific EX Series switches.

Juniper Networks EX Series Ethernet Switches support Differentiated Service Code Point (DSCP) or IP precedence and IEEE 802.1p CoS classifiers on the customer-edge interfaces of the ingress provider

edge (PE) switch. DSCP or IP precedence classifiers are used for Layer 3 packets. IEEE 802.1p is used for Layer 2 packets.

When a packet enters a customer-edge interface of the ingress PE switch, the switch associates the packet with a particular CoS servicing level before putting the packet onto the label-switched path (LSP). The switches within the LSP utilize the CoS value set at the ingress PE switch. The CoS value that was embedded in the classifier is translated and encoded in the MPLS header by means of the EXP or experimental bits. EX Series switches enable a default EXP classifier and a default EXP rewrite rule. For more information about EXP classifiers and EXP *rewrite rules*, see EXP Classifiers and EXP rewrite Rules.

This topic includes:

EXP Classifiers and EXP rewrite Rules

EX Series switches enable a default EXP classifier and a default EXP rewrite rule. You can configure a custom EXP classifier and a custom EXP rewrite rule if you prefer. However, the switch supports only one type of EXP classifier (default or custom) and only one EXP rewrite rule (default or custom).

You do not bind the EXP classifier or the EXP rewrite rule to individual interfaces. The switch automatically and implicitly applies the default or the custom EXP classifier and the default or the custom EXP rewrite rule to the appropriate MPLS-enabled interfaces. Because rewrite rules affect only egress interfaces, the switch applies the EXP rewrite rule only to those MPLS interfaces that are transmitting MPLS packets (not to the MPLS interfaces that are receiving the packets).

After traversing the MPLS tunnel, the traffic flows out from the egress provider edge (PE) switch. Before the traffic leaves the egress interface, the egress PE switch copies the EXP bits from the MPLS header to the most significant bits in the original IP packet--- that is, to the IP precedence bits.

Guidelines for Using CoS Classifiers on CCCs

When you are configuring CoS for MPLS over circuit cross-connect (CCC), there are some additional guidelines, as follows:

- You *must* explicitly bind a CoS classifier to the CCC interface on the ingress PE switch.

- You *must* use the same DSCP, IP precedence, or IEEE 802.1p classifier on CCC interfaces. However, if the CCC interfaces are on the same switch, you cannot configure both a DSCP and an IP precedence classifier on these interfaces. Thus, if you configure one CCC interface to use a DSCP classifier DSCP1, you cannot configure another CCC interface to use another DSCP classifier DSCP2. All the CCC interfaces on the switch must use the same DSCP (or IP precedence) classifier and the same IEEE 802.1p classifier.

- You *cannot* configure one CCC interface to use a DSCP classifier and another CCC interface to use an IP precedence classifier, because these classifier types overlap.

- You *can* configure one CCC interface to use a DSCP classifier and another CCC interface to use IEEE 802.1p classifier.

- You *can* configure one CCC interface to use both a DSCP and an IEEE 802.1p classifier. If you configure a CCC interface to use both these classifiers, the DSCP classifier is used for routing Layer 3 packets and the IEEE 802.1p classifier is used for routing Layer 2 packets.

- You *can* configure one CCC interface to use both an IP precedence and an IEEE 802.1p classifier. If you configure a CCC interface to use both these classifiers, the IP precedence classifier is used for routing Layer 3 packets and the IEEE 802.1p classifier is used for routing Layer 2 packets.

You can define multiple DSCP, IP precedence, and IEEE 802.1p classifiers for the non-CCC interfaces on a switch.

Using CoS Classifiers with IP over MPLS

When you are configuring CoS for IP over MPLS, the customer-edge interface uses the CoS configuration for the switch as the default. You do not have to bind a classifier to the customer-edge interface in this case. There are no restrictions on using multiple DSCP, IP precedence, and IEEE 802.1p classifiers on the same switch.

- You can modify the CoS classifier for a particular interface, but it is not required.

- You can configure a DSCP classifier, DSCP1 on the first interface, another DSCP classifier, DSCP2 on the second interface, and an IP precedence classifier on a third interface, and so forth.

Setting CoS Bits in an MPLS Header

When traffic enters an LSP tunnel, the CoS bits in the MPLS header are set in one of two ways:

- The number of the output queue into which the packet was buffered and the packet loss priority (PLP) bit are written into the MPLS header and are used as the packet's CoS value. This behavior is the default, and no configuration is required. The Junos OS Class of Service User Guide for Routing Devices explains the IP CoS values, and summarizes how the CoS bits are treated.

- You set a fixed CoS value on all packets entering the LSP tunnel. A fixed CoS value means that all packets entering the LSP receive the same class of service.

The CoS value can be a decimal number from 0 through 7. This number corresponds to a 3-bit binary number. The high-order 2 bits of the CoS value select which transmit queue to use on the outbound interface card.

The low-order bit of the CoS value is treated as the PLP bit and is used to select the RED drop profile to use on the output queue. If the low-order bit is 0, the non-PLP drop profile is used, and if the low-order bit is 1, the PLP drop profile is used. It is generally expected that random early detection (RED) will more

aggressively drop packets that have the PLP bit set. For more information about RED and drop profiles, see the Junos OS Class of Service User Guide for Routing Devices.

> **NOTE**: Configuring the PLP drop profile to drop packets more aggressively (for example, setting the CoS value from 6 to 7) decreases the likelihood of traffic getting through.

Table 35 on page 191 summarizes how MPLS CoS values correspond to the transmit queue and PLP bit. Note that in MPLS, the mapping between the CoS bit value and the output queue is hard-coded. You cannot configure the mapping for MPLS; you can configure it only for IPv4 traffic flows, as described in the Junos OS Class of Service User Guide for Routing Devices.

Table 35: MPLS CoS Values

MPLS CoS Value	Bits	Transmit Queue	PLP Bit
0	000	0	Not set
1	001	0	Set
2	010	1	Not set
3	011	1	Set
4	100	2	Not set
5	101	2	Set
6	110	3	Not set
7	111	3	Set

Because the CoS value is part of the MPLS header, the value is associated with the packets only while they travel through the LSP tunnel. The value is not copied back to the IP header when the packets exit from the LSP tunnel.

EXP Rewrite Rules

When traffic passes from the customer-edge interface to an MPLS interface, the DSCP, IP precedence, or IEEE 802.1p CoS classifier is translated into the EXP bits within the MPLS header. You cannot disable the default EXP rewrite rule, but you can configure your own custom EXP classifier and a custom EXP rewrite rule. You cannot bind the EXP classifier to individual MPLS interfaces; the switch applies it globally to all the MPLS-enabled interfaces on the switch.

Only one EXP rewrite rule (either default or custom) is supported on a switch. The switch applies it to all the egress interfaces on which MPLS is enabled.

Policer

Policing helps to ensure that the amount of traffic forwarded through an LSP never exceeds the requested bandwidth allocation. During periods of congestion (when the total rate of queuing packets exceeds the rate of transmission), any new packets being sent to an interface can be dropped because there is no place to store them. You can configure a policer on the ingress PE switch to prevent this:

- If you are using MPLS over CCC, you bind the policer to the LSP. You cannot bind a policer to a CCC interface.

- If you are using IP over MPLS, you bind the policer to the **inet-family** customer-edge interface. You cannot bind a policer to the LSP when you are using IP over MPLS.

Schedulers

The schedulers for using CoS with MPLS are the same as for the other CoS configurations on EX Series switches. Default schedulers are provided for best-effort and network-control forwarding classes. If you are using assured-forwarding, expedited-forwarding, or any custom forwarding class, we recommend that you configure a scheduler to support that forwarding class. See "Understanding CoS Schedulers" on page 125.

Example: Combining CoS with MPLS on EX Series Switches

IN THIS SECTION

- Requirements | **193**
- Overview and Topology | **193**

You can use class of service (CoS) within MPLS networks to prioritize certain types of traffic during periods of congestion. The CoS value is included within the MPLS label, which is passed through the network, enabling end-to-end CoS across the network.

MPLS services are often used to ensure better performance for low-latency applications such as VoIP and other business-critical functions. These applications place specific demands on a network for successful transmission. CoS gives you the ability to control the mix of bandwidth, delay, jitter, and packet loss while taking advantage of the MPLS labeling mechanism.

This example shows how to configure CoS on an MPLS network that is using a unidirectional circuit cross-connect (CCC) from the ingress provider edge (PE) switch to the egress PE switch. The configuration is for the customer-edge interface of the ingress PE switch. It describes adding the configuration of CoS components to the ingress PE switch, the egress PE switch, and the core provider switches of the existing MPLS network. Because of the unidirectional configuration, the DSCP classifier needs to be configured only on the ingress PE switch.

Requirements

This example uses the following hardware and software components:

- Junos OS Release 10.1 or later for EX Series switches

- Three EX Series switches

Before you configure CoS with MPLS, be sure you have:

Configured an MPLS network with two PE switches and one provider switch. See *Basic MPLS Configuration*. This example assumes that an MPLS network has been configured using a cross circuit-connect (CCC).

Overview and Topology

This example describes adding custom classifiers and custom rewrite rules to switches in an MPLS network that is using MPLS over CCC.

It is a unidirectional configuration. Therefore, you need to configure custom classifiers and custom rewrite rules as follows:

- On the ingress PE switch: custom DSCP classifier and custom EXP rewrite rule

- On the egress PE switch: custom EXP classifier

- On the provider switch: customer EXP classifier and custom EXP rewrite rule

> **NOTE**: You can also configure schedulers and shapers as needed. If you are using **assured-forwarding**, **expedited-forwarding**, or other custom forwarding classes, we recommend that you configure a scheduler to support that forwarding class. See "Defining CoS Schedulers and Scheduler Maps (CLI Procedure)" on page 129.

The example creates a custom DSCP classifier (**dscp1**) on the ingress PE switch and binds this classifier to the CCC interface. It includes configuration of a policer on the ingress PE switch. The policer is applied as a filter on the label-switched path (LSP) **lsp_to_pe2_ge1** to ensure that the amount of traffic forwarded through the LSP never exceeds the requested bandwidth allocation.

This example creates a custom EXP rewrite rule (**exp1**) on the ingress PE switch, specifying a loss-priority and code point to be used for the expedited-forwarding class as the packet travels through the LSP. The switch applies this custom rewrite rule on the core interfaces **ge-0/0/5.0** and **ge-0/0/6.0**, which are the egress interfaces for this switch.

Table 36 on page 194 shows the CoS configuration components added to the ingress PE switch.

Table 36: CoS Configuration Components on the Ingress PE Switch

Property	Settings	Description
Local PE switch hardware	EX Series switch	PE-1
Policing filter configured and applied to the LSP.	**policing filter mypolicer** **filter myfilter**	Name of the rate-limiting policer. Name of the filter, which refers to the policer
Custom DSCP classifier	**dscp1**	Specifies the name of the custom DSCP classifier

Table 36: CoS Configuration Components on the Ingress PE Switch *(Continued)*

Property	Settings	Description
Custom EXP rewrite rule	**e1**	Name of the custom EXP rewrite rule.
Customer-edge interface	**ge-0/0/1.0**	Interface that receives packets from devices outside the network. The custom DSCP classifier must be specified on this CCC interface.
Core interfaces	**ge-0/0/5.0** and **ge-0/0/6.0**	Interfaces that transmit MPLS packets to other switches within the MPLS network. The EXP rewrite rule is applied implicitly to these interfaces.

shows the CoS configuration components added to the egress PE switch in this example.

Table 37: CoS Configuration Components of the Egress PE Switch

Property	Settings	Description
Remote provider edge switch hardware	EX Series switch	PE-2
Custom EXP classifier	**exp1**	Name of custom EXP classifier
Customer-edge interface	**ge-0/0/1.0**	Interface that transmits packets from this network to devices outside the network. No CoS classifier is specified for this interface. A scheduler can be specified.

Table 37: CoS Configuration Components of the Egress PE Switch *(Continued)*

Property	Settings	Description
Core interfaces	**ge-0/0/7.0** and **ge-0/0/8.0**	Core interfaces on PE-2 that receive MPLS packets from the provider switch. The EXP classifier is enabled by default on the switch and applied implicitly to these interfaces.

Table 38 on page 196 shows the MPLS configuration components used for the provider switch in this example.

Table 38: CoS Configuration Components of the Provider Switch

Property	Settings	Description
Provider switch hardware	EX Series switch	Transit switch within the MPLS network configuration.
Custom EXP classifier	**exp1**	Name of the custom EXP classifier.
Custom EXP rewrite rule	**e1**	Name of the custom EXP rewrite rule.
Core interfaces receiving packets from other MPLS switches.	**ge-0/0/5.0** and **ge-0/0/6.0**	Interfaces that connect the provider switch to the ingress PE switch (PE-1). The EXP classifier is enabled by default on the switch and applied implicitly to these interfaces.
Core interfaces transmitting packets to other switches within the MPLS network.	**ge-0/0/7.0** and **ge-0/0/8.0**	Interfaces that transmit packets to the egress PE (PE-2). The EXP rewrite rule is applied implicitly on these interfaces. Schedulers can also be specified and will be applied to these interfaces.

Topology

Configuring the Local PE Switch

IN THIS SECTION

● Procedure | **197**

Procedure

CLI Quick Configuration

To quickly configure a custom DSCP classifier, custom EXP rewrite rule, and a policer on the local PE switch, copy the following commands and paste them into the switch terminal window of PE-1:

```
[edit]

set class-of-service classifiers dscpset class-of-service classifiers dscp dscp1 import default
set class-of-service classifiers dscp dscp1 forwarding-class expedited-forwarding loss-priority
low code-points 000111
set class-of-service rewrite-rules exp e1 forwarding-class expedited-forwarding loss-priority
low code-point 111
set class-of-service interfaces ge-0/0/1 unit 0 classifier dscp1
set firewall policer mypolicer if-exceeding bandwidth-limit 500m
set firewall policer mypolicer if-exceeding burst-size-limit 33553920
set firewall policer mypolicer then discard
set firewall family any filter myfilter term t1 then policer mypolicer
set protocols mpls label-switched-path lsp_to_pe2_ge1 to 127.1.1.3 policing filter myfilter
```

Step-by-Step Procedure

To configure a custom DSCP classifier, custom EXP rewrite rule, and a policer on the ingress PE switch:

1. Import the default DSCP classifier classes to the custom DSCP classifier that you are creating:

```
[edit class-of-service]
user@switch# set classifiers dscp dscp1 import default
```

2. Add the expedited-forwarding class to this custom DSCP classifier, specifying a loss priority and code point:

```
[edit class-of-service]
user@switch# set classifiers dscp dscp1 forwarding-class expedited-forwarding loss-priority
low code-points 000111
```

3. Specify the values for the custom EXP rewrite rule, **e1**:

```
[edit class-of-service]
user@switch# set rewrite-rules exp e1 forwarding-class expedited-forwarding loss-priority low
code-point 111
```

4. Bind the DSCP classifier to the CCC interface:

```
[edit class-of-service]
user@switch# set class-of-service interfaces ge-0/0/1 unit 0 classifier dscp1
```

5. Specify the number of bits per second permitted, on average, for the firewall policer, which will later be applied to the LSP:

```
[edit firewall]
set policer mypolicer if-exceeding bandwidth-limit 500m
```

6. Specify the maximum size permitted for bursts of data that exceed the given bandwidth limit for this policer:

```
[edit firewall policer]
set mypolicer if-exceeding burst-size-limit 33553920
```

7. Discard traffic that exceeds the rate limits for this policer:

```
[edit firewall policer]
set mypolicer then discard
```

8. To reference the policer, configure a filter term that includes the policer action:

```
[edit firewall]
user@switch# set family any filter myfilter term t1 then policer
mypolicer
```

9. Apply the filter to the LSP:

```
[edit protocols mpls]
set label-switched-path lsp_to_pe2_ge1 policing filter myfilter
```

Results

Display the results of the configuration:

```
[edit]
user@switch# show
class-of-service {
    classifiers {
        dscp dscp1 {
            import default;
            forwarding-class expedited-forwarding {
                loss-priority low code-points 000111;
            }
        }
    }
    interfaces {
        ge-0/0/1 {
            unit 0 {
                classifiers {
                    dscp dscp1;
                }
            }
        }
```

```
        }
    rewrite-rules {
        exp e1 {
            forwarding-class expedited-forwarding {
                loss-priority low code-point 111;
            }
        }
    }
}
firewall {
    family any {
        filter myfilter {
            term t1 {
                then policer mypolicer;
            }
        }
    }
    policer mypolicer {
        if-exceeding {
            bandwidth-limit 500m;
            burst-size-limit 33553920;
        }
        then discard;
    }
}
```

Configuring the Remote PE Switch

IN THIS SECTION

- Procedure | **201**

Procedure

CLI Quick Configuration

To quickly configure a custom EXP classifier on the remote PE switch, copy the following commands and paste them into the switch terminal window of PE-2:

```
[edit]
set class-of-service classifiers exp exp1 import default
set class-of-service classifiers exp exp1 forwarding-class expedited-forwarding loss-priority
low code-points 010
```

Step-by-Step Procedure

To configure a custom EXP classifier on the egress PE switch:

1. Import the default EXP classifier classes to the custom EXP classifier that you are creating:

   ```
   [edit class-of-service]
   user@switch# set classifiers exp exp1 import default
   ```

2. Add the expedited-forwarding class to this custom EXP classifier, specifying a loss priority and code point:

   ```
   [edit class-of-service]
   user@switch# set classifiers exp exp1 forwarding-class expedited-forwarding loss-priority low
   code-points 010
   ```

Results

Display the results of the configuration:

```
[edit]
 user@switch# show
class-of-service {
    classifiers {
        exp exp1 {
```

```
        import default;
        forwarding-class expedited-forwarding {
            loss-priority low code-points 010;
        }
    }
}
}
```

Configuring the Provider Switch

IN THIS SECTION

● Procedure | **202**

Procedure

CLI Quick Configuration

To quickly configure a custom EXP classifier and a custom EXP rewrite rule on the provider switch, copy the following commands and paste them into the switch terminal window of the provider switch:

```
[edit]
set class-of-service classifiers exp exp1 import default
set class-of-service classifiers exp exp1 forwarding-class expedited-forwarding loss-priority
low code-points 010
set class-of-service rewrite-rules exp e1 forwarding-class expedited-forwarding loss-priority
low code-point 111
```

Step-by-Step Procedure

To configure a custom EXP classifier and a custom EXP rewrite rule on the provider switch:

1. Import the default EXP classifier classes to the custom EXP classifier that you are creating:

```
[edit class-of-service]
user@switch# set classifiers exp exp1 import default
```

2. Add the expedited-forwarding class to this custom EXP classifier, specifying a loss priority and code point:

```
[edit class-of-service]
user@switch# set classifiers exp exp1 forwarding-class expedited-forwarding loss-priority low
code-points 010
```

3. Specify the values for the custom EXP rewrite rule, **e1**:

```
[edit class-of-service]
user@switch# set rewrite-rules exp e1 forwarding-class expedited-forwarding loss-priority low
code-point 111
```

Results

Display the results of the configuration:

```
[edit]
 user@switch# show
class-of-service {
    classifiers {
        exp exp1 {
            import default;
            forwarding-class expedited-forwarding {
                loss-priority low code-points 010;
            }
        }
    }
    rewrite-rules {
        exp e1 {
            forwarding-class expedited-forwarding {
                loss-priority low code-point 111;
            }
        }
    }
}
```

Verification

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

Verifying That the Policer Firewall Filter Is Operational

Purpose

Verify the operational state of the policer that is configured on the ingress PE switch.

Action

```
user@switch> show firewall

Filter: myfilter
Policers:
Name                                       Packets
mypolicer-t1                                     0
```

Meaning

This output shows that the firewall filter **mypolicer** has been created.

Verifying That the CoS Classifiers Are Going to the Right Queue

Purpose

Verify that the CoS classifiers are going to the right queue.

Action

```
user@switch> show class-of-service forwarding-table classifier

Classifier table index: 7, # entries: 64, Table type: DSCP
Entry #   Code point   Forwarding-class #   PLP
   0       000000            0            0
   1       000001            0            0
   2       000010            0            0
   3       000011            0            0
   4       000100            0            0
   5       000101            0            0
   6       000110            0            0
   7       000111            0            0
   8       001000            0            0
   9       001001            0            0
  10       001010            0            0
  11       001011            0            0
  12       001100            0            0
  13       001101            0            0
  14       001110            0            0
  15       001111            0            0
  16       010000            0            0
  17       010001            0            0
  18       010010            0            0
  19       010011            0            0
  20       010100            0            0
  21       010101            0            0
  22       010110            0            0
  23       010111            0            0
  24       011000            0            0
  25       011001            0            0
  26       011010            0            0
  27       011011            0            0
  28       011100            0            0
  29       011101            0            0
  30       011110            0            0
  31       011111            0            0
  32       100000            0            0
  33       100001            0            0
  34       100010            0            0
  35       100011            0            0
```

```
  36        100100        0        0
  37        100101        0        0
  38        100110        0        0
  39        100111        0        0
  40        101000        0        0
  41        101001        0        0
  42        101010        0        0
  43        101011        0        0
  44        101100        0        0
  45        101101        0        0
  46        101110        0        0
  47        101111        0        0
  48        110000        3        0
  49        110001        3        0
  50        110010        3        0
  51        110011        3        0
  52        110100        3        0
  53        110101        3        0
  54        110110        3        0
  55        110111        3        0
  56        111000        3        0
  57        111001        3        0
  58        111010        3        0
  59        111011        3        0
  60        111100        3        0
  61        111101        3        0
  62        111110        3        0
  63        111111        3        0


Classifier table index: 11, # entries: 8, Table type: IEEE 802.1
Entry #   Code point   Forwarding-class #   PLP
   0          000          0        0
   1          001          0        0
   2          010          0        0
   3          011          0        0
   4          100          0        0
   5          101          0        0
   6          110          3        0
   7          111          3        0


Classifier table index: 12, # entries: 8, Table type: IPv4 precedence
Entry #   Code point   Forwarding-class #   PLP
   0          000          0        0
```

```
    1          001         0         0
    2          010         0         0
    3          011         0         0
    4          100         0         0
    5          101         0         0
    6          110         3         0
    7          111         3         0


Classifier table index: 16, # entries: 8, Table type: Untrust
Entry #   Code point   Forwarding-class #   PLP
    0          000         0         0
    1          001         0         0
    2          010         0         0
    3          011         0         0
    4          100         0         0
    5          101         0         0
    6          110         0         0
    7          111         0         0


Classifier table index: 9346, # entries: 64, Table type: DSCP
Entry #   Code point   Forwarding-class #   PLP
    0         000000       0         0
    1         000001       0         0
    2         000010       0         0
    3         000011       0         0
    4         000100       0         0
    5         000101       0         0
    6         000110       0         0
    7         000111       1         0
    8         001000       0         0
    9         001001       0         0
   10         001010       0         0
   11         001011       0         0
   12         001100       0         0
   13         001101       0         0
   14         001110       0         0
   15         001111       0         0
   16         010000       0         0
   17         010001       0         0
   18         010010       0         0
   19         010011       0         0
   20         010100       0         0
   21         010101       0         0
```

22	010110	0	0
23	010111	0	0
24	011000	0	0
25	011001	0	0
26	011010	0	0
27	011011	0	0
28	011100	0	0
29	011101	0	0
30	011110	0	0
31	011111	0	0
32	100000	0	0
33	100001	0	0
34	100010	0	0
35	100011	0	0
36	100100	0	0
37	100101	0	0
38	100110	0	0
39	100111	0	0
40	101000	0	0
41	101001	0	0
42	101010	0	0
43	101011	0	0
44	101100	0	0
45	101101	0	0
46	101110	0	0
47	101111	0	0
48	110000	3	0
49	110001	3	0
50	110010	3	0
51	110011	3	0
52	110100	3	0
53	110101	3	0
54	110110	3	0
55	110111	3	0
56	111000	3	0
57	111001	3	0
58	111010	3	0
59	111011	3	0
60	111100	3	0
61	111101	3	0
62	111110	3	0

```
    63        111111       3       0
```

Meaning

This output shows that a new DSCP classifier has been created, index **9346**, on the ingress PE switch (PE-1).

Verifying the CoS Forwarding Table Mapping

Purpose

For each logical interface, display either the table index of the classifier for a given code point type or the queue number (if it is a fixed classification) in the forwarding table.

Action

```
user@switch> show class-of-service forwarding-table classifier mapping

                      Table Index/
Interface      Index     Q num      Table type
ge-0/0/1.0        92      9346       DSCP
```

Meaning

The results show that the new DSCP classifier, index number **9346**, is bound to interface **ge-0/0/1.0**.

Verifying the Rewrite Rules

Purpose

Display mapping of the queue number and loss priority to code point value for each rewrite rule as it exists in the forwarding table.

Action

```
user@switch>show class-of-service forwarding-table rewrite-rule

Rewrite table index: 31, # entries: 4, Table type: DSCP
```

```
FC#     Low bits  State      High bits  State
0       000000  Enabled      000000  Enabled
1       101110  Enabled      101110  Enabled
2       001010  Enabled      001100  Enabled
3       110000  Enabled      111000  Enabled


Rewrite table index: 34, # entries: 4, Table type: IEEE 802.1
FC#     Low bits  State      High bits  State
0          000  Enabled         001  Enabled
1          010  Enabled         011  Enabled
2          100  Enabled         101  Enabled
3          110  Enabled         111  Enabled


Rewrite table index: 35, # entries: 4, Table type: IPv4 precedence
FC#     Low bits  State      High bits  State
0          000  Enabled         000  Enabled
1          101  Enabled         101  Enabled
2          001  Enabled         001  Enabled
3          110  Enabled         111  Enabled


Rewrite table index: 9281, # entries: 1, Table type: EXP
FC#     Low bits  State      High bits  State
1          111  Enabled         000  Disabled
```

Meaning

This output shows that a new EXP classifier with the index number **9281** has been created.

Configuring CoS on an MPLS Provider Edge Switch Using IP Over MPLS

IN THIS SECTION

- Configuring CoS | **211**
- Configuring an LSP Policer | **211**

You can use class of service (CoS) within MPLS networks to prioritize certain types of traffic during periods of congestion. This topic describes configuring CoS components on a provider edge (PE) switch that is using IP Over MPLS.

This task describes how to create a custom DSCP classifier and a custom EXP rewrite rule on the ingress PE switch. It includes configuring a policer firewall filter and applying it to the customer-edge interface of the ingress PE switch. The policer firewall filter ensures that the amount of traffic forwarded through the MPLS tunnel never exceeds the requested bandwidth allocation.

Before you begin, configure the basic components for an MPLS network:

- Configure two PE switches. See *Basic MPLS Configuration*.

- Configure one or more provider switches.

Configuring CoS

To configure CoS on a provider edge switch:

1. Import the default DSCP classifier classes to the custom DSCP classifier that you are creating:

```
[edit class-of-service]
user@switch# set classifiers dscp classifier-name import default
```

2. Add a forwarding class to this custom DSCP classifier and specify a loss priority and code point:

```
[edit class-of-service]
user@switch# set classifiers dscp classifier-name forwarding-class forwarding-class loss-
priority loss-priority code-points code-point
```

3. Specify the values for the custom EXP rewrite rule, e1:

```
[edit class-of-service]
user@switch# set rewrite-rules exp e1 forwarding-class forwarding-class loss-priority loss-
priority code-points code-point
```

Configuring an LSP Policer

To configure an LSP policer:

1. Specify the number of bits per second permitted, on average, for the firewall policer, which will later be applied to the customer-edge-interface:

```
[edit firewall]
user@switch# set policer mypolicer if-exceeding bandwidth-limit 500m
```

2. Specify the maximum size permitted for bursts of data that exceed the given bandwidth limit for this policer:

```
[edit firewall policer]
user@switch# set mypolicer if-exceeding burst-size-limit 33553920
```

3. Discard traffic that exceeds the rate limits for this policer:

```
[edit firewall policer]
user@switch# set mypolicer then discard
```

4. To reference the policer, configure a filter term that includes the policer action:

```
[edit firewall]
user@switch# set family inet filter myfilter term t1 then policer mypolicer
```

5. Apply the filter to the customer-edge interface:

```
[edit interfaces]
user@switch# set ge-2/0/3 unit 0 family inet address 192.168.121.1/16 policing filter myfilter
```

> **NOTE**: You can also configure schedulers and shapers as needed. See "Defining CoS Schedulers and Scheduler Maps (CLI Procedure)" on page 129.

Configuring CoS on an MPLS Provider Edge Switch Using Circuit Cross-Connect

IN THIS SECTION

- Configuring CoS | **213**
- Configuring an LSP Policer | **214**

You can use class of service (CoS) within MPLS networks to prioritize certain types of traffic during periods of congestion. This topic describes configuring CoS components on a provider edge (PE) switch that is using MPLS over circuit-cross connect (CCC).

> (i) **NOTE**: If you are using MPLS over CCC, you can use only one DSCP or IP precedence classifier and only one IEEE 802.1p classifier on the CCC interfaces.

This procedure is for creating a custom DSCP classifier and a custom EXP rewrite rule on the ingress PE. It also includes enabling a policer on the label-switched path (LSP) of the ingress PE to ensure that the amount of traffic forwarded through the LSP never exceeds the requested bandwidth allocation.

This topic includes:

Configuring CoS

To configure CoS on a provider edge switch:

1. Import the default DSCP classifier classes to the custom DSCP classifier that you are creating:

   ```
   [edit class-of-service]
   user@switch# set classifiers dscp classifier-nameimport default
   ```

2. Add the expedited-forwarding class to this custom DSCP classifier, specifying a loss priority and code point:

   ```
   [edit class-of-service]
   user@switch# set classifiers dscp classifier-name forwarding-class forwarding-class loss-
   priority loss-priority code-points code-point
   ```

3. Specify the values for the custom EXP rewrite rule, e1:

```
                            [edit class-of-service]
user@switch# set rewrite-rules exp e1 forwarding-class forwarding-class loss-priority loss-
priority code-point code-point
```

4. Bind the DSCP classifier to the CCC interface:

```
[edit ]
user@switch# set class-of-service interfaces interface unit unit classifier classifier-
name
```

Configuring an LSP Policer

To configure an LSP policer:

1. Specify the number of bits per second permitted, on average, for the policer, which will later be applied to the LSP:

```
[edit firewall]
set policer mypolicer if-exceeding bandwidth-limit 500m
```

2. Specify the maximum size permitted for bursts of data that exceed the given bandwidth limit for this policer:

```
[edit firewall policer]
set mypolicer if-exceeding burst-size-limit 33553920
```

3. Discard traffic that exceeds the rate limits for this policer:

```
[edit firewall policer]
set mypolicer then discard
```

4. To reference the policer, configure a filter term that includes the policer action:

```
                            [edit firewall]
user@switch# set family any filter myfilter term t1 then policer
mypolicer
```

5. Apply the filter to the LSP:

```
[edit protocols mpls]
set label-switched-path lsp_to_pe2_ge1 policing filter myfilter
```

> **NOTE**: You can also configure schedulers and shapers as needed. See "Defining CoS Schedulers and Scheduler Maps (CLI Procedure)" on page 129.

Configuring CoS on Provider Switches of an MPLS Network

You can add class-of-service (CoS) components to your MPLS networks on EX Series switches to achieve end-to-end Differentiated Services to match your specific business requirements. The configuration of CoS components on the provider switches is the same regardless of whether the provider edge (PE) switches are using MPLS over CCC or IP over MPLS.

This task shows how to configure a custom EXP classifier and custom EXP rewrite rule on the provider switch.

1. Import the default EXP classifier classes to the custom EXP classifier that you are creating:

```
[edit class-of-service]
user@switch# set classifiers exp exp1 import default
```

2. Add the expedited-forwarding class to this custom EXP classifier, specifying a loss priority and code point:

```
[edit class-of-service]
user@switch# set classifiers exp exp1 forwarding-class expedited-forwarding loss-priority low
code-points 010
```

3. Specify the values for the custom EXP rewrite rule, **e1**:

```
[edit class-of-service]
user@switch# set rewrite-rules exp e1 forwarding-class expedited-forwarding loss-priority low
code-point 111
```

NOTE: You can also configure schedulers and shapers as needed. See "Defining CoS Schedulers and Scheduler Maps (CLI Procedure)" on page 129.

CoS on EVPN VXLANs

IN THIS CHAPTER

CoS Support on EVPN VXLANs

IN THIS SECTION

You can configure class of service (CoS) features on VXLAN interfaces. VXLAN traffic from different tenants traverses network boundaries over the same physical underlay network. To ensure fairness in the treatment of traffic for all tenants in the VXLAN, and to prioritize higher priority traffic, apply CoS features to the VXLAN interfaces.

Understanding CoS on VXLAN Interfaces

This section describes how classification and rewrite rules are applied to packets in a VXLAN instance. shows a simple VXLAN with two leaf nodes and one spine node.

Figure 7: Classifiers and Rewrite Rules on VXLANs
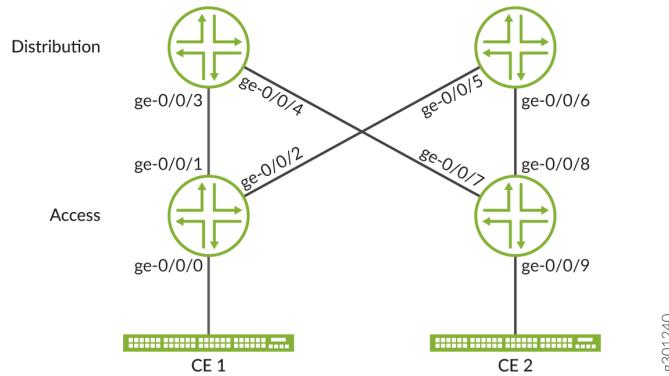
Refer to Figure 7 on page 218 to understand the packet flow with DSCP/ToS fields in a VXLAN:

1. CE 1 sends a packet with Layer3 DSCP/ToS bit programmed to the Leaf 1 node.

2. Leaf 1 receives the original packet and appends the VXLAN header on top of the original packet. The outer VXLAN Layer3 header uses the original packet DSCP/Tos bit. You can create classifiers based on the original packet DSCP/802.1p bit. The ingress interface on the ingress leaf supports DSCP and 802.1p classifiers.

3. If rewrite is configured on Leaf 1, the inner header will have the DSCP/802.1p bit set by CE 1 and the outer header will have the rewrite bit. Only DSCP rewrite rules are supported, except on QFX10000 switches where 802.1p rewrite is also supported if the underlay is tagged.

4. The Spine node receives the VXLAN packet and can use ingress classification using these DSCP bits and forward the packet to the egress interface with the appropriate forwarding class.

5. The Spine egress interface can rewrite these bits using rewrite rules. These Spine rewrite rules only affects the outer Layer3 DSCP field. The inner/original packet still holds the DSCP/802.1p bit that was set by CE 1.

6. Leaf 2 receives the packet, processes the tunnel termination, and remove the outer VXLAN header.

7. Leaf 2 classification and rewrite functionality works on the inner header.

8. The original packet arrives on CE 2.

> **NOTE**: On the leaf nodes, if the packet is multicast, you can use `multi-destination` classification to create appropriate multicast classification and rewrite rules.

Configuring CoS on VXLAN Interfaces

This section shows sample configurations of classifiers and rewrite rules for the leaf and spine nodes in VXLAN using Figure 7 on page 218 as a reference. You can create schedulers as normal for the classifiers on each node.

Sample configuration of classifiers and rewrite rules on Leaf 1.

1. Create a classifier based on the *original* DSCP/ToS bits:

```
[edit class-of-service classifiers]
user@leaf1#set dscp dscp_cf forwarding-class best-effort loss-priority low code-points 100000
user@leaf1#set dscp dscp_cf forwarding-class network-control loss-priority high code-points
110000
user@leaf1#set dscp dscp_cf forwarding-class expedited-forwarding loss-priority low code-
points 011010
user@leaf1#set dscp dscp_cf forwarding-class assured-forwarding loss-priority high code-
points 001010
```

2. Apply the classier to the ingress interface:

```
[edit class-of-service interfaces]
user@leaf1#set ge-0/0/0 unit 0 classifiers dscp dscp_cf
```

3. Create a rewrite rule for the *outer* VXLAN DSCP/ToS bits:

```
[edit class-of-service rewrite-rules]
user@leaf1#set dscp dscp_rw forwarding-class best-effort loss-priority low code-points af22
user@leaf1#set dscp dscp_rw forwarding-class network-control loss-priority high code-points
af31
user@leaf1#set dscp dscp_rw forwarding-class expedited-forwarding loss-priority low code-
points af13
user@leaf1#set dscp dscp_rw forwarding-class assured-forwarding loss-priority high code-
points cs3
```

4. Apply the rewrite rule to the egress Leaf 1 interfaces:

```
[edit class-of-service interfaces]
user@leaf1#set ge-0/0/1 unit 0 rewrite-rules dscp dscp_rw
user@leaf1#set ge-0/0/2 unit 0 rewrite-rules dscp dscp_rw
```

Sample configuration of classifiers and rewrite rules on the Spine.

1. Create a classifier based on the outer VXLAN DSCP/ToS bits:

```
[edit class-of-service classifiers]
user@spine#set dscp dscp_cf forwarding-class best-effort loss-priority low code-points af22
user@spine#set dscp dscp_cf forwarding-class network-control loss-priority high code-points af31
user@spine#set dscp dscp_cf forwarding-class expedited-forwarding loss-priority low code-points af13
user@spine#set dscp dscp_cf forwarding-class assured-forwarding loss-priority high code-points cs3
```

2. Apply the classier to the ingress Spine interfaces:

```
[edit class-of-service interfaces]
user@spine#set ge-0/0/3 unit 0 classifiers dscp dscp_cf
user@spine#set ge-0/0/5 unit 0 classifiers dscp dscp_cf
```

3. Create a rewrite rule for the outer VXLAN DSCP/ToS bits:

```
[edit class-of-service rewrite-rules]
user@spine#set dscp dscp_rw forwarding-class best-effort loss-priority low code-points af22
user@spine#set dscp dscp_rw forwarding-class network-control loss-priority high code-points af31
user@spine#set dscp dscp_rw forwarding-class expedited-forwarding loss-priority low code-points af13
user@spine#set dscp dscp_rw forwarding-class assured-forwarding loss-priority high code-points cs3
```

4. Apply the rewrite rule to the egress Spine interfaces:

```
[edit class-of-service interfaces]
user@spine#set ge-0/0/4 unit 0 rewrite-rules dscp dscp_rw
user@spine#set ge-0/0/6 unit 0 rewrite-rules dscp dscp_rw
```

Sample configuration of classifiers and rewrite rules on Leaf 2.

1. Create a classifier based on the *original* DSCP/ToS bits, as the VXLAN header is removed at tunnel termination *before* forwarding classes are applied:

```
[edit class-of-service classifiers]
user@leaf2#set dscp dscp_cf forwarding-class best-effort loss-priority low code-points 100000
user@leaf2#set dscp dscp_cf forwarding-class network-control loss-priority high code-points
110000
user@leaf2#set dscp dscp_cf forwarding-class expedited-forwarding loss-priority low code-
points 011010
user@leaf2#set dscp dscp_cf forwarding-class assured-forwarding loss-priority high code-
points 001010
```

2. Apply the classier to the ingress Leaf 2 interfaces:

```
[edit class-of-service interfaces]
user@leaf2#set ge-0/0/7 unit 0 classifiers dscp dscp_cf
user@leaf2#set ge-0/0/8 unit 0 classifiers dscp dscp_cf
```

3. Create a rewrite rule for the *original* DSCP/ToS bits:

```
[edit class-of-service rewrite-rules]
user@leaf2#set dscp dscp_rw forwarding-class best-effort loss-priority low code-points 100000
user@leaf2#set dscp dscp_rw forwarding-class network-control loss-priority high code-points
110000
user@leaf2#set dscp dscp_rw forwarding-class expedited-forwarding loss-priority low code-
points 011010
user@leaf2#set dscp dscp_rw forwarding-class assured-forwarding loss-priority high code-
points 001010
```

4. Apply the rewrite rule to the egress Leaf 2 interface:

```
[edit class-of-service interfaces]
user@leaf2#set ge-0/0/9 unit 0 rewrite-rules dscp dscp_rw
```

To check the CoS configuration on one of the interfaces:

```
user@node#show class-of-service interface interface-name
```

To check the queue statistics on one of the interfaces:

```
user@node#show interfaces queue interface-name
```

Implementing CoS on VXLAN Interfaces (Junos OS Evolved)

CoS for EVPN VXLAN traffic is supported using a combination of classifiers, schedulers, and rewrite rules. This section describes how these components are implemented across different nodes on devices running Junos OS Evolved to apply CoS on the EVPN VXLAN traffic.

- **Classification at User Network Interface (UNI)/Ingress PE** — Traffic classification based on IEEE 802.1p and Differentiated Services code point (DSCP) are supported on the ingress PE where the EVPN VXLAN tunnel is initiated. BA and MF classifiers can be applied to Enterprise style (EP) or Service Provider (SP) style access interfaces.

- **Classification at Network Node Interface (NNI)/Egress PE** — Traffic classification based on IEEE 802.1p and Differentiated Services code point (DSCP) are supported on the egress PE where the EVPN VXLAN tunnel is terminated. BA classifiers can be applied to the underlying logical interface or unit. MF classifiers are not supported in tunnel terminations.

- **Rewrite at NNI** — After the encapsulation of the VXLAN tunnel, the rewrites on the outer/tunnel header are configured using the rewrite rules on the underlying logical interface or unit. Based on the configured rewrite rules, the VXLAN traffic is classified in the Spine/Network.
  DSCP rewrite on the outer/tunnel header of VXLAN packets is supported on the NNI interface.

  Rewrite rules are supported in the following EVPN VXLAN scenarios:

  - Intra-VNI L2 gateway — Rewrite rules are applied to both unicast and broadcast, unknown unicast, and multicast (BUM) traffic.

  - Inter-VNI L3 gateway — Centrally-routed bridging (CRB) and edge-routed bridging (ERB).

  - EVPN Type 5 routes.

- **Rewrite at UNI** — After the termination of the VXLAN tunnel, the rewrites on the inner headers are configured using rewrite rules on the Enterprise style (EP) or Service Provider (SP) style access interfaces. Based on the configured rewrite rules, the de-encapsulated packets are classified in the CE side network. The following rewrite rules are supported on the UNI interface for the de-encapsulated packets:

  - DSCP rewrites on the inner IPv4/IPv6 header

  - IEEE 802.1p rewrites on the inner Ethernet header (if tagged)

  Rewrite rules are supported in the following EVPN VXLAN scenarios:

- Intra-VNI L2 gateway — Rewrite rules are applied to both unicast and broadcast, unknown unicast, and multicast (BUM) traffic.

- Inter-VNI L3 gateway — Centrally-routed bridging (CRB) and edge-routed bridging (ERB).

- EVPN Type 5 routes.

- **Scheduling** — Traffic prioritization and bandwidth reservation are achieved by using schedulers. The schedulers are associated with a forwarding class set via classifiers.

Platform-Specific CoS on EVPN-VxLANs Behavior

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

Use the following table to review platform-specific behaviors for CoS on EVPN-VxLANS:

Platform	Difference
EX4100, EX4300MP, and EX4400 switches	EX4100, EX4300MP, and EX4400 switches support CoS features such as classification and rewrite on IRB interfaces that carry EVPN-VxLAN traffic.

(Continued)

Platform	Difference
PTX Series routers	The following limitations apply to PTX routers: • DSCP rewrite rules are not supported on Integrated Routing and Bridging (IRB) (L3 gateway scenarios). • IEEE 802.1p rewrite rules are not supported on the NNI interface. • Explicit congestion notification (ECN) rewrites are not supported on either UNI or NNI interfaces. • Priority-based flow control (PFC) is not supported. • No support for CoS classification and rewrite mechanism for IRB underlay. **NOTE**: In EVPN-VXLAN networks with an IPv6 underlay, some PTX Series routers support CoS classification and explicit congestion notification (ECN) copy operations, but not PFC, DSCP copy, or IEEE 802.1p rewrite. See Feature Explorer for platform and release support.
PTX10002-36QDD routers	PTX10002-36QDD routers that originate or terminate EVPN-VXLAN tunnels and have ECN enabled automatically copy the ECN bits from the inner header to the outer header. The router copies the ECN bits from the outer header to the inner header if the inner header has the ECT bit set. If the router experiences congestion, it sets the CE bits if the ECT bit is enabled. These routers support ECN copy operations in EVPN-VXLAN networks with an IPv4 underlay or an IPv6 underlay. PTX10002-36QDD routers that originate EVPN-VXLAN tunnels copy the DSCP bits from the inner to the outer header during encapsulation. If DSCP rewrite is configured, DSCP rewrite takes precedence over DSCP copy.

(Continued)

Platform	Difference

QFX5130 and QFX5700 switches	The following limitations apply to the QFX5130 and QFX5700 platforms:
	
	• HQoS is not supported due to hardware limitations.
	
	• Classifier, rewrite and scheduler on IRB interface is not supported.
	
	• DOT1P rewrite and classifier on the NNI port is not supported.
	
	• DOT1P and DSCP rewrite on the UNI port is not supported.
	
	• DSCP rewrite on the NNI port is supported with the following conditions:
	
	   • DSCP rewrite on the NNI port is supported by default and does not work if `vxlan-tos-copy-filter` at `[edit forwarding-options]` is enabled.
	
	   • Inner ECN bits are copied to the outer VXLAN header regardless of whether `vxlan-tos-copy-filter` at `[edit forwarding-options]` is enabled.
	
	   • Do not enable `vxlan-tos-copy-filter` and `vxlan-disable-copy-tos-encap` at `[edit forwarding-options]`at the same time as this causes in-deterministic behavior.
	
	• PFC configuration will cause momentary traffic drops of up to 10ms.
	
	• DSCP IPV6 classifiers and rewrites are not supported. Use DSCP classifier and rewrite instead.
	
	• TOS copy feature does not work for Type-5 EVPN VXLANs.

(Continued)

Platform	Difference
QFX10000 Series switches	The following limitation applies to QFX10000 platforms: • Because IRB interfaces do not support dscp rewrite rules, you can apply rewrite rules on underlying L2 interfaces. 802.1p/dscp values in a VXLAN tunneled packet are written using underlying L2 interface rules.

4
PART

Configuration Statements and Operational Commands

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