Engineering
Simplicity

Junos® OS

Class of Service User Guide (Security Devices)

Published
2025-12-12

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

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

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

Junos® OS Class of Service User Guide (Security Devices)
Copyright © 2025 Juniper Networks, Inc. All rights reserved.

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

YEAR 2000 NOTICE

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

END USER LICENSE AGREEMENT

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

Table of Contents

About This Guide | xi

1 Overview

Introduction to Class of Service | 2

Understanding Class of Service | 2

Benefits of CoS | 4

CoS Across the Network | 5

Junos OS CoS Components | 6

CoS Components Packet Flow | 8

CoS Device Configuration Overview | 10

Understanding CoS Default Settings | 11

2 Configuring Class of Service Components

Assigning Service Levels with Classifiers | 14

Classification Overview | 14

Understanding Packet Loss Priorities | 18

Default Behavior Aggregate Classification | 19

Sample Behavior Aggregate Classification | 21

Example: Configure and Apply Custom BA Classifiers | 23

- Requirements | 23
- Overview | 23
- Configuration | 24
- Verification | 27

Apply MPLS EXP Classifiers to Routing Instances | 31

- Configure and Apply Custom MPLS EXP Classifiers to Routing Instances | 32
- Apply Global Classifiers and Wildcard Routing Instances | 33
- Apply Global MPLS EXP Classifiers to Routing Instances | 34
- Apply Classifiers by Using Wildcard Routing Instances | 36

- Verify the Classifiers Associated with Routing Instances | 37

Controlling Network Access with Traffic Policing | 39

Simple Filters and Policers Overview | 39

Two-Rate Three-Color Policer Overview | 41

Example: Configuring a Two-Rate Three-Color Policer | 42

- Requirements | 42
- Overview | 42
- Configuration | 43
- Verification | 48

Logical Interface (Aggregate) Policer Overview | 50

Two-Color Policer Configuration Overview | 50

Example: Configuring a Two-Color Logical Interface (Aggregate) Policer | 55

- Requirements | 55
- Overview | 56
- Configuration | 56
- Verification | 62

Guidelines for Configuring Simple Filters | 64

Example: Configure and Apply a Firewall Filter for a Multifield Classifier | 68

- Requirements | 68
- Overview | 68
- Configuration | 70
- Verification | 74

Controlling Output Queues with Forwarding Classes | 77

Forwarding Classes Overview | 77

Example: Configuring Forwarding Classes | 80

Example: Assigning Forwarding Classes to Output Queues | 86

- Requirements | 86
- Overview | 86
- Configuration | 87
- Verification | 89

Example: Classify all Traffic from a Remote Device with Fixed Interface-Based Classification | 89

- Requirements | 90
- Overview | 90
- Configuration | 92
- Verification | 95

Understanding the SPC High-Priority Queue | 97

Example: Configuring the SPC High-Priority Queue | 98

- Requirements | 98
- Overview | 98
- Configuration | 99
- Verification | 101

Understanding Queuing and Marking of Host Outbound Traffic | 102

Default Routing Engine Protocol Queue Assignments | 104

Altering Outgoing Packets Headers with Rewrite Rules | 108

Rewrite Rules Overview | 108

Rewriting Frame Relay Headers | 109

- Assigning the Default Frame Relay Rewrite Rule to an Interface | 109
- Defining a Custom Frame Relay Rewrite Rule | 110

Example: Configure and Apply Rewrite Rules on a Security Device | 110

- Requirements | 111
- Overview | 111
- Configuration | 112
- Verification | 115

Defining Output Queue Properties with Schedulers | 117

Schedulers Overview | 117

Default Scheduler Settings | 123

Transmission Scheduling Overview | 124

Excess Bandwidth Sharing and Minimum Logical Interface Shaping | 126

Excess Bandwidth Sharing Proportional Rates | 127

Calculated Weights Mapped to Hardware Weights | 128

Weight Allocation with Only Shaping Rates or Unshaped Logical Interfaces | **129**

Shared Bandwidth Among Logical Interfaces | **131**

Example: Configuring Class-of-Service Schedulers on a Security Device | **133**

- Requirements | **133**
- Overview | **133**
- Configuration | **134**
- Verification | **138**

Scheduler Buffer Size Overview | **139**

Configuring Large Delay Buffers in CoS | **143**

Example: Configuring and Applying Scheduler Maps | **149**

- Requirements | **149**
- Overview | **149**
- Configuration | **150**
- Verification | **152**

Applying Scheduler Maps and Shaping Rate to Logical Interfaces and VLANs | **153**

Example: Applying Scheduling and Shaping to VLANs | **158**

- Requirements | **159**
- Overview | **159**
- Configuration | **159**

Removing Delays with Strict-Priority Queues | 169

Strict-Priority Queue Overview | **169**

Understanding Strict-Priority Queues | **170**

Example: Configuring Priority Scheduling | **171**

- Requirements | **171**
- Overview | **171**
- Configuration | **171**
- Verification | **174**

Example: Configuring Strict-Priority Queuing | **174**

- Requirements | **175**
- Overview | **175**
- Configuration | **175**

  - Verification | 187

Example: Configuring CoS Non-Strict Priority Scheduling | 188
  - Requirements | 189
  - Overview | 189
  - Configuration | 190
  - Verification | 193

Controlling Congestion with Drop Profiles | 194

RED Drop Profiles Overview | 194

RED Drop Profiles and Congestion Control | 196

Configuring RED Drop Profiles | 198

Example: Configuring RED Drop Profiles | 201
  - Requirements | 201
  - Overview | 201
  - Configuration | 202
  - Verification | 204

Example: Configuring Segmented and Interpolated Style Profiles | 205
  - Requirements | 205
  - Overview | 205
  - Configuration | 205
  - Verification | 209

Controlling Congestion with Explicit Congestion Notification | 210

CoS Explicit Congestion Notification (ECN) | 210
  - Overview | 210
  - How ECN Works | 211
  - WRED Drop Profile Control of ECN Thresholds | 216
  - Dynamic ECN | 219
  - ECN Packets per Queue | 220
    - Overview | 220
    - Configuration | 221
  - On-Chip Buffer with ECN on PTX10002-36QDD Routers | 222
  - Platform-Specific Behavior | 223

Limiting Traffic Using Virtual Channels | 227

Virtual Channels Overview | **227**

Understanding Virtual Channels | **228**

Example: Configuring Virtual Channels | **230**

- Requirements | **230**
- Overview | **230**
- Configuration | **230**
- Verification | **236**

Enabling Queuing for Tunnel Interfaces | 237

CoS Queuing for Tunnels Overview | **237**

Understanding the ToS Value of a Tunnel Packet | **242**

Example: Configuring CoS Queuing for GRE or IP-IP Tunnels | **243**

- Requirements | **243**
- Overview | **244**
- Configuration | **245**
- Verification | **247**

Copy Outer IP Header DSCP and ECN to Inner IP Header | **250**

Understanding CoS Support on st0 Interfaces | **251**

Naming Components with Code-Point Aliases | 254

Code-Point Aliases Overview | **254**

Default CoS Values and Aliases | **255**

Example: Defining Code-Point Aliases for Bits on a Security Device | **258**

- Requirements | **258**
- Overview | **259**
- Configuration | **259**
- Verification | **260**

3

Configuring Class of Service Scheduler Hierarchy

Controlling Traffic by Configuring Scheduler Hierarchy | 262

Understanding Hierarchical Schedulers | **262**

Understanding Internal Scheduler Nodes | **266**

Example: Configuring a Four-Level Scheduler Hierarchy | **267**

  - Requirements | 267
  - Overview | 268
  - Configuration | 269
  - Verification | 284

Example: Controlling Remaining Traffic | 285

  - Requirements | 285
  - Overview | 286
  - Configuration | 288
  - Verification | 292

4 Configuring Class of Service for IPv6

Configuring Class of Service for IPv6 Traffic | 295

CoS Functions for IPv6 Traffic Overview | 295

Understanding CoS with DSCP IPv6 BA Classifier | 297

Example: Configuring CoS with DSCP IPv6 BA Classifiers | 298

  - Requirements | 298
  - Overview | 298
  - Configuration | 298
  - Verification | 302

Understanding DSCP IPv6 Rewrite Rules | 302

Example: Configuring CoS with DSCP IPv6 Rewrite Rules | 303

  - Requirements | 304
  - Overview | 304
  - Configuration | 304
  - Verification | 307

5 Configuring Class of Service for I/O Cards

Configuring Class of Service for I/O Cards | 310

PIR-Only and CIR Mode Overview | 310

Understanding Priority Propagation | 312

Understanding IOC Hardware Properties | 315

Understanding IOC Map Queues | 317

WRED on the IOC Overview | **318**

MDRR on the IOC Overview | **324**

CoS Support on the SRX5000 Module Port Concentrator Overview | **326**

Example: Configuring CoS on SRX5000 Firewalls with an MPC | **327**

- Requirements | **328**
- Overview | **328**
- Configuration | **330**
- Verification | **338**

6

Configuration Statements and Operational Commands

Junos CLI Reference Overview | 342

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 security devices.

1
PART

Overview

- Introduction to Class of Service | 2

CHAPTER 1

Introduction to Class of Service

IN THIS CHAPTER

- Understanding Class of Service | 2
- Benefits of CoS | 4
- CoS Across the Network | 5
- Junos OS CoS Components | 6
- CoS Components Packet Flow | 8
- CoS Device Configuration Overview | 10
- Understanding CoS Default Settings | 11

Understanding Class of Service

IN THIS SECTION

- Platform-Specific Behavior | 3

When a network experiences congestion and delay, some packets must be dropped. Junos OS *class of service* (CoS) allows you to divide traffic into classes and offer various levels of throughput and packet loss when congestion occurs. This allows packet loss to happen according to the rules you configure.

For interfaces that carry IPv4, IPv6, or MPLS traffic, you can configure the Junos OS CoS features to provide multiple classes of service for different applications. On the device, you can configure multiple forwarding classes for transmitting packets, define which packets are placed into each output queue, schedule the transmission service level for each queue, and manage congestion using a random early detection (RED) algorithm.

Traffic shaping is the allocation of the appropriate amount of network bandwidth to every user and application on an interface. The appropriate amount of bandwidth is defined as cost-effective carrying capacity at a guaranteed CoS. You can use a Juniper Networks device to control traffic rate by applying classifiers and shapers.

The CoS features provide a set of mechanisms that you can use to provide differentiated services when best-effort delivery is insufficient.

Using Junos OS CoS features, you can assign service levels with different delay, *jitter* (delay variation), and packet loss characteristics to particular applications served by specific traffic flows. CoS is especially useful for networks supporting time-sensitive video and audio applications.

CoS features include traffic classifying, policing, queuing, scheduling, shaping and marker rewriting. You can configure all these features on the physical interfaces. So, the speeds of physical interfaces are of very much importance for CoS.

If an interface speed configured is none of these speeds then the speed considered for CoS features is 1-Gbps.

Overall performance of network traffic is usually measured by aspects such as the bandwidth, delay, and error rate. If there is congestion in the network then packets are dropped. CoS helps divide the traffic during the time of congestion. So, with the different physical interface speed rates supported to configure CoS the CoS performance is improved.

NOTE: Policing, scheduling, and shaping CoS services are not supported for pre-encryption and post-encryption packets going into and coming out of an IPsec VPN tunnel.

Junos OS supports the following RFCs for traffic classification and policing:

- RFC 2474, *Definition of the Differentiated Services Field in the IPv4 and IPv6*
- RFC 2475, *An Architecture for Differentiated Services*
- RFC 2597, *Assured Forwarding PHB Group*
- RFC 2598, *An Expedited Forwarding PHB*
- RFC 2697, *A Single Rate Three Color Marker*
- RFC 2698, A *Two Rate Three Color Marker*

Platform-Specific Behavior

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

Table 1: Platform-Specific Behavior

Platform	Difference
vSRX Virtual Firewall	• On vSRX Virtual Firewall and vSRX Virtual Firewall 3.0 instances, different physical interface speed rates of 1-Gbps, 10-Gbps, 40-Gbps, and 100-Gbps are supported to configure CoS features. VMXNET3 or VIRTIO interface speed is 10Gbps, SR-IOV interface speed depends on the ethernet card.

RELATED DOCUMENTATION

Junos OS CoS Components | **6**
CoS Components Packet Flow | **8**
Understanding CoS Default Settings | **11**
CoS Device Configuration Overview | **10**

Benefits of CoS

IP routers normally forward packets independently, without controlling throughput or delay. This type of packet forwarding, known as *best-effort service*, is as good as your network equipment and links allow. Best-effort service is sufficient for many traditional IP data delivery applications, such as e-mail or Web browsing. However, newer IP applications such as real-time video and audio (or voice) require lower delay, *jitter*, and packet loss than simple best-effort networks can provide.

CoS features allow a Juniper Networks device to improve its processing of critical packets while maintaining best-effort traffic flows, even during periods of congestion. Network throughput is determined by a combination of available bandwidth and delay. CoS dedicates a guaranteed minimum bandwidth to a particular service class by reducing forwarding queue delays. (The other two elements of overall network delay, serial transmission delays determined by link speeds and propagation delays determined by media type, are not affected by CoS settings.)

Normally, packets are queued for output in their order of arrival, regardless of service class. Queuing delays increase with network congestion and often result in lost packets when queue buffers overflow. CoS packet classification assigns packets to forwarding queues by service class.

Because CoS must be implemented consistently end-to-end through the network, the CoS features on the Juniper Networks device are based on IETF Differentiated Services (DiffServ) standards to interoperate with other vendors' CoS implementations.

RELATED DOCUMENTATION

Understanding Class of Service | 2

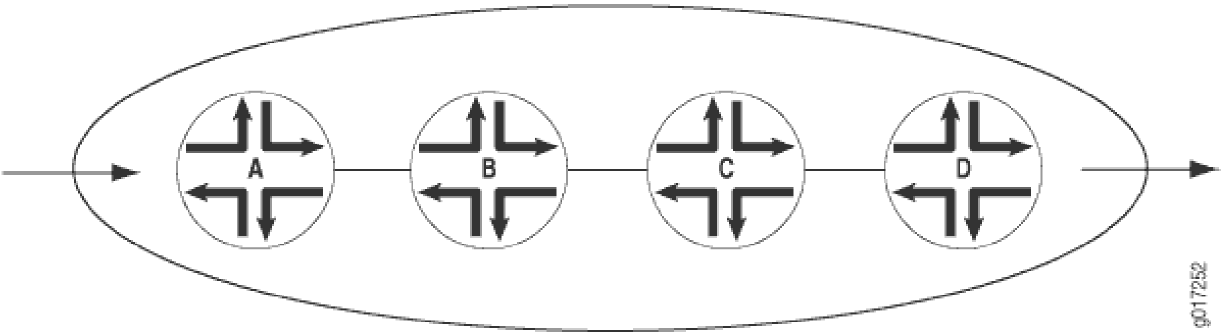
CoS Across the Network

CoS works by examining traffic entering at the edge of your network. The edge devices classify traffic into defined service groups, which allow for 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 to ensure that higher-paying customers receive better service. As the traffic leaves the network at the far edge, you can reclassify the traffic.

To support CoS, you must configure each device in the network. Generally, each device examines the packets that enter it to determine their CoS settings. These settings then dictate which packets are first transmitted to the next downstream device. In addition, the devices at the edges of your network might be required to alter the CoS settings of the packets transmitting to the neighboring network.

Figure 1 on page 5 shows an example of CoS operating across an Internet Service Provider (ISP) network.

Figure 1: CoS Across the Network



In the ISP network shown in Figure 1 on page 5, Device A is receiving traffic from your network. As each packet enters, Device A examines the packet's current CoS settings and classifies the traffic into one of the groupings defined by the ISP. This definition allows Device A to prioritize its resources for servicing the traffic streams it is receiving. In addition, Device A might alter the CoS settings (forwarding class and loss priority) of the packets to better match the ISP's traffic groups. When Device B receives the packets, it examines the CoS settings, determines the appropriate traffic group, and processes the packet according to those settings. Device B then transmits the packets to Device C, which performs the same actions. Device D also examines the packets and determines the appropriate group. Because it sits at the far end of the network, the ISP might decide once again to alter the CoS settings of the packets before Device D transmits them to the neighboring network.

RELATED DOCUMENTATION

Understanding Class of Service | 2

Junos OS CoS Components

Junos OS supports CoS components on Juniper Networks devices as indicated in Table 2 on page 6.

Table 2: Supported Junos OS CoS Components

Junos OS CoS Component	Description	For More Information
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.	"Code-Point Aliases Overview" on page 254
Classifiers	Packet classification refers to the examination of an incoming packet. This function associates the packet with a particular CoS servicing level. Two general types of classifiers are supported—behavior aggregate (BA) classifiers and multifield (MF) classifiers. When both BA and MF classifications are performed on a packet, the MF classification has higher precedence.	"Classification Overview" on page 14
Forwarding classes	Forwarding classes allow you to group packets for transmission. Based on forwarding classes, you assign packets to output queues. The forwarding class plus the loss priority define the per-hop behavior (PHB in DiffServ) of a packet. Juniper Networks routers and services gateways support eight queues (0 through 7).	"Forwarding Classes Overview" on page 77

Table 2: Supported Junos OS CoS Components *(Continued)*

Junos OS CoS Component	Description	For More Information
Loss priorities	Loss priorities allow you to set the priority of dropping a packet. You can use the loss priority setting to identify packets that have experienced congestion.	"Understanding Packet Loss Priorities" on page 18
Forwarding policy options	CoS-based forwarding (CBF) enables you to control next-hop selection based on a packet's class of service and, in particular, the value of the IP packet's precedence bits. For example, you can specify a particular interface or next hop to carry high-priority traffic while all best-effort traffic takes some other path.	*Example: Classifying All Traffic from a Remote Device by Configuring Fixed Interface-Based Classification*
Transmission queues	After a packet is sent to the outgoing interface on a device, it is queued for transmission on the physical media. The amount of time a packet is queued on the device is determined by the availability of the outgoing physical media as well as the amount of traffic using the interface. Juniper Networks routers and services gateways support queues 0 through 7.	"Transmission Scheduling Overview" on page 124
Schedulers	An individual device interface has multiple queues assigned to store packets temporarily before transmission. To determine the order to service the queues, the device uses a round-robin scheduling method based on priority and the queue's weighted round-robin (WRR) credits. Junos OS schedulers allow you to define the priority, bandwidth, delay buffer size, rate control status, and RED drop profiles to be applied to a particular queue for packet transmission.	"Schedulers Overview" on page 117
Virtual channels	On Juniper Networks routers and services gateways, you can configure virtual channels to limit traffic sent from a corporate headquarters to branch offices. Virtual channels might be required when the headquarters site has an expected aggregate bandwidth higher than that of the individual branch offices. The router at the headquarters site must limit the traffic sent to each branch office router to avoid oversubscribing their links.	"Virtual Channels Overview" on page 227

Table 2: Supported Junos OS CoS Components *(Continued)*

Junos OS CoS Component	Description	For More Information
Policers for traffic classes	Policers allow you to limit traffic of a certain class to a specified bandwidth and burst size. Packets exceeding the policer limits can be discarded, or can be assigned to a different forwarding class, a different loss priority, or both. You define policers with firewall filters that can be associated with input or output interfaces.	"Simple Filters and Policers Overview" on page 39
Rewrite rules	A rewrite rule modifies the appropriate CoS bits in an outgoing packet. Modification of CoS bits allows the next downstream device to classify the packet into the appropriate service group. Rewriting or marking outbound packets is useful when the device is at the border of a network and must alter the CoS values to meet the policies of the targeted peer.	"Rewrite Rules Overview" on page 108

RELATED DOCUMENTATION

Understanding Class of Service | 2
CoS Components Packet Flow | 8
Understanding CoS Default Settings | 11
CoS Device Configuration Overview | 10

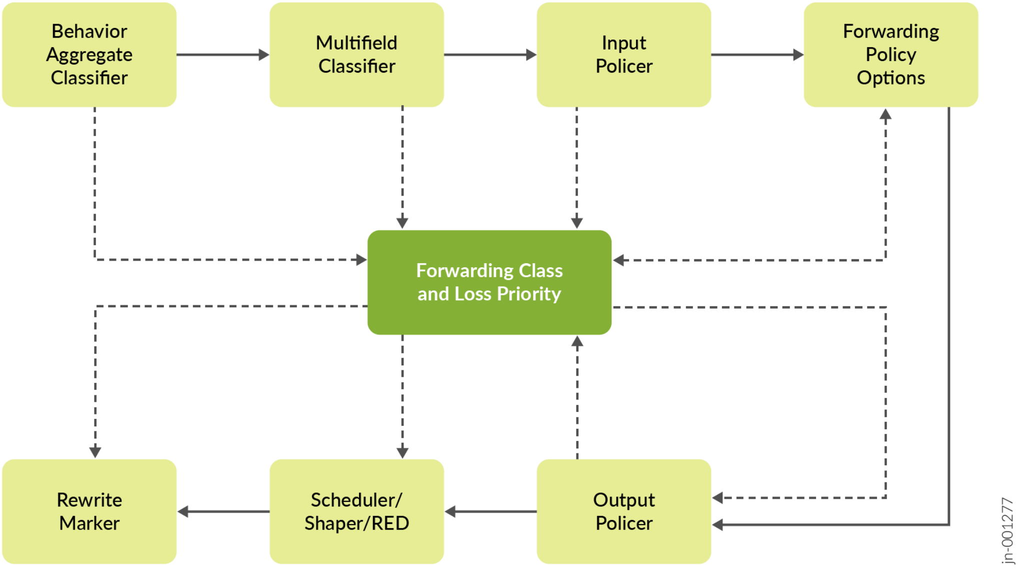
CoS Components Packet Flow

IN THIS SECTION

- CoS Process on Incoming Packets | 9
- CoS Process on Outgoing Packets | 10

On Juniper Networks devices, you configure CoS functions using different components. These components are configured individually or in a combination to define particular CoS services. Figure 2 on page 9 displays the relationship of different CoS components to each other and illustrates the sequence in which they interact.

Figure 2: Packet Flow Through Juniper Networks Device


Behavior Aggregate Classifier
Multifield Classifier
Input Policer
Forwarding Policy Options
Forwarding Class and Loss Priority
Rewrite Marker
Scheduler/ Shaper/RED
Output Policer
jn-001277

Each box in Figure 2 on page 9 represents a CoS component. The solid lines show the direction of packet flow in a device. The upper row indicates an incoming packet, and the lower row an outgoing packet. The dotted lines show the inputs and outputs of particular CoS components. For example, the forwarding class and loss priority are outputs of behavior aggregate classifiers and multifield classifiers and inputs for rewrite markers and schedulers.

Typically, only a combination of some components shown in Figure 2 on page 9 (not all) is used to define a CoS service offering. For example, if a packet's class is determined by a behavior aggregate classifier, it is associated with a forwarding class and loss priority and does not need further classification by the multifield classifier.

This section contains the following topics:

CoS Process on Incoming Packets

Classifiers and policers perform the following operations on incoming packets:

1. A classifier examines an incoming packet and assigns a forwarding class and loss priority to it.

2. Based on the forwarding class, the packet is assigned to an outbound transmission queue.

3. Input policers meter traffic to see if traffic flow exceeds its service level. Policers might discard, change the forwarding class and loss priority, or set the PLP bit of a packet. A packet for which the PLP bit is set has an increased probability of being dropped during congestion.

CoS Process on Outgoing Packets

The scheduler map and rewrite rules perform the following operations on outgoing packets:

1. Scheduler maps are applied to interfaces and associate the outgoing packets with a scheduler and a forwarding class.
2. The scheduler defines how the packet is treated in the output transmission queue based on the configured transmit rate, buffer size, priority, and drop profile.
    - The buffer size defines the period for which the packet is stored during congestion.
    - The scheduling priority and transmit rate determine the order in which the packet is transmitted.
    - The drop profile defines how aggressively to drop packets that are using a particular scheduler.
3. Output policers meter traffic and might change the forwarding class and loss priority of a packet if a traffic flow exceeds its service level.
4. The rewrite rule writes information to the packet (for example, EXP or DSCP bits) according to the forwarding class and loss priority of the packet.

RELATED DOCUMENTATION

Understanding Class of Service | **2**

Junos OS CoS Components | **6**

Understanding CoS Default Settings | **11**

CoS Device Configuration Overview | **10**

CoS Device Configuration Overview

Before you begin configuring a Juniper Networks device for CoS, complete the following tasks:

- Determine whether the device needs to support different traffic streams, such as voice or video. If so, CoS helps to make sure this traffic receives more than basic best-effort packet delivery service.

- Determine whether the device is directly attached to any applications that send CoS-classified packets. If no sources are enabled for CoS, you must configure and apply rewrite rules on the interfaces facing the sources.
- Determine whether the device must support assured forwarding (AF) classes. Assured forwarding usually requires random early detection (RED) drop profiles to be configured and applied.
- Determine whether the device must support expedited forwarding (EF) classes with a policer. Policers require you to apply a burst size and bandwidth limit to the traffic flow, and set a consequence for packets that exceed these limits—usually a high loss priority, so that packets exceeding the policer limits are discarded first.

RELATED DOCUMENTATION

CLI Explorer
Understanding Class of Service | 2
CoS Components Packet Flow | 8
Understanding CoS Default Settings | 11

Understanding CoS Default Settings

The *Class of Service* menu in J-Web allows you to configure most of the Junos OS CoS components for the IPv4 and MPLS traffic on a Juniper Networks device. You can configure forwarding classes for transmitting packets, define which packets are placed into each output queue, schedule the transmission service level for each queue, and manage congestion using a random early detection (RED) algorithm. After defining the CoS components, you must assign classifiers to the required physical and logical interfaces.

Even when you do not configure any CoS settings on your routing platform, the software performs some CoS functions to ensure that user traffic and protocol packets are forwarded with minimum delay when the network is experiencing congestion. Some default mappings are automatically applied to each *logical interface* that you configure. Other default mappings, such as explicit default classifiers and *rewrite rules*, are in operation only if you explicitly associate them with an interface.

You can display default CoS settings by running the **show class-of-service** *operational mode command*.

You configure CoS when you need to override the default packet forwarding behavior of a Juniper Networks device—especially in the three areas identified in Table 3 on page 12.

Table 3: Reasons to Configure Class of Service (CoS)

Default Behavior to Override with CoS	CoS Configuration Area
Packet classification—By default, the Juniper Networks device does not use behavior aggregate (BA) classifiers to classify packets. Packet classification applies to incoming traffic.	Classifiers
Scheduling queues—By default, the Juniper Networks device has only two queues enabled. Scheduling queues apply to outgoing traffic.	Schedulers
Packet headers—By default, the Juniper Networks device does not rewrite CoS bits in packet headers. Rewriting packet headers applies to outgoing traffic.	Rewrite rules

RELATED DOCUMENTATION

Understanding Class of Service | **2**
CoS Components Packet Flow | **8**
CoS Device Configuration Overview | **10**

2
PART

Configuring Class of Service Components

- Assigning Service Levels with Classifiers | 14
- Controlling Network Access with Traffic Policing | 39
- Controlling Output Queues with Forwarding Classes | 77
- Altering Outgoing Packets Headers with Rewrite Rules | 108
- Defining Output Queue Properties with Schedulers | 117
- Removing Delays with Strict-Priority Queues | 169
- Controlling Congestion with Drop Profiles | 194
- Controlling Congestion with Explicit Congestion Notification | 210
- Limiting Traffic Using Virtual Channels | 227
- Enabling Queuing for Tunnel Interfaces | 237
- Naming Components with Code-Point Aliases | 254

CHAPTER 2

Assigning Service Levels with Classifiers

IN THIS CHAPTER

- Classification Overview | **14**
- Understanding Packet Loss Priorities | **18**
- Default Behavior Aggregate Classification | **19**
- Sample Behavior Aggregate Classification | **21**
- Example: Configure and Apply Custom BA Classifiers | **23**
- Apply MPLS EXP Classifiers to Routing Instances | **31**

Classification Overview

IN THIS SECTION

- Behavior Aggregate Classifiers | **15**
- Multifield Classifiers | **16**
- Default IP Precedence Classifier | **17**

Packet classification refers to the examination of an incoming packet, which associates the packet with a particular class-of-service (CoS) servicing level. Junos operating system (OS) supports these classifiers:

- Behavior aggregate (BA) classifiers
- Multifield (MF) classifiers
- Default IP precedence classifiers

NOTE: The total number of classifiers supported on a Services Processing Unit (SPU) is 79. Three classifiers are installed on the SPU as default classifiers in the Layer 3 mode, independent of any CoS configuration, which leaves 76 classifiers that can be configured using the CoS CLI commands. The default classifiers number can vary in future releases or in different modes.

Verify the number of default classifiers installed on the SPU to determine how many classifiers can be configured using the CoS CLI commands.

When both BA and MF classifications are performed on a packet, the MF classification has higher precedence.

In Junos OS, classifiers associate incoming packets with a forwarding class (FC) and packet loss priority (PLP), and, based on the associated FC, assign packets to output queues. A packet's FC and PLP specify the behavior of a hop, within the system, to process the packet. The per-hop behavior (PHB) comprises packet forwarding, policing, scheduling, shaping, and marking. For example, a hop can put a packet in one of the priority queues according to its FC and then manage the queues by checking the packet's PLP. Junos OS supports up to eight FCs and four PLPs.

Behavior Aggregate Classifiers

A BA classifier operates on a packet as it enters the device. Using BA classifiers, the device aggregates different types of traffic into a single FC so that all the types of traffic will receive the same forwarding treatment. The CoS value in the packet header is the single field that determines the CoS settings applied to the packet. BA classifiers allow you to set a packet's FC and PLP based on the Differentiated Services (DiffServ) code point (DSCP) value, DSCP IPv4 value, DSCP IPv6 value, IP precedence value, MPLS EXP bits, or IEEE 802.1p value. The default classifier is based on the IP precedence value. For more information, see "Default IP Precedence Classifier" on page 17.

Junos OS performs BA classification for a packet by examining its Layer 2, Layer 3, and related CoS parameters, as shown in Table 4 on page 15.

Table 4: BA Classification

Layer	CoS Parameter
Layer 2	IEEE 802.1p value: User Priority

Table 4: BA Classification *(Continued)*

Layer	CoS Parameter
Layer 3	IPv4 precedence IPv4 Differentiated Services code point (DSCP) value IPv6 DSCP value

NOTE: If you apply an IEEE 802.1 classifier to a logical interface, this classifier takes precedence over any other classifier type. Classifiers for IEEE 802.1, IP (DSCP or IP precedence), and MPLS (EXP) can coexist on a logical interface.

Multifield Classifiers

An MF classifier is a second means of classifying traffic flows. Unlike the BA classifier, an MF classifier can examine multiple fields in the packet—for example, the source and destination address of the packet, or the source and destination port numbers of the packet. With MF classifiers, you set the FC and PLP based on *firewall filter* rules.

NOTE: For a specified interface, you can configure both an MF classifier and a BA classifier without conflicts. Because the classifiers are always applied in sequential order (the BA classifier followed by the MF classifier) any BA classification result is overridden by an MF classifier if they conflict.

Junos OS performs MF traffic classification by directly scrutinizing multiple fields of a packet to classify a packet. This avoids having to rely on the output of the previous BA traffic classification. Junos OS can simultaneously check a packet's data for Layers 2, 3, 4, and 7, as shown in Table 5 on page 16.

Table 5: MF Classification

Layer	CoS Parameter
Layer 2	IEEE 802.1Q: VLAN ID IEEE 802.1p: User priority

Table 5: MF Classification *(Continued)*

Layer	CoS Parameter
Layer 3	IP precedence value DSCP or DSCP IPv6 value Source IP address Destination IP address Protocol ICMP: Code and type
Layer 4	TCP/UDP: Source port TCP/UDP: Destination port TCP: Flags AH/ESP: SPI
Layer 7	Not supported.

Using Junos OS, you configure an MF classifier with a firewall filter and its associated match conditions. This enables you to use any filter match criterion to locate packets that require classification.

Default IP Precedence Classifier

With Junos OS, all *logical interface* are automatically assigned a default IP precedence classifier when the logical interface is configured. This default traffic classifier maps IP precedence values to an FC and a PLP as shown in Table 6 on page 17. These mapping results are in effect for an ingress packet until the packet is further processed by another classification method.

Table 6: Default IP Precedence Classifier

IP Precedence CoS Values	Forwarding Class	Packet Loss Priority
000	best-effort	low
001	best-effort	high

Table 6: Default IP Precedence Classifier *(Continued)*

IP Precedence CoS Values	Forwarding Class	Packet Loss Priority
010	best-effort	low
011	best-effort	high
100	best-effort	low
101	best-effort	high
110	network-control	low
111	network-control	high

RELATED DOCUMENTATION

Default Behavior Aggregate Classification | 19
Sample Behavior Aggregate Classification | 21
Example: Configure and Apply Custom BA Classifiers | 23

Understanding Packet Loss Priorities

Packet loss priorities (PLPs) allow you to set the priority for dropping packets. You can use the PLP setting to identify packets that have experienced congestion. Typically, you mark packets exceeding some service level with a high loss priority—that is, a greater likelihood of being dropped. You set PLP by configuring a classifier or a policer. The PLP is used later in the work flow to select one of the drop profiles used by random early detection (RED).

You can configure the PLP bit as part of a congestion control strategy. The PLP bit can be configured on an interface or in a filter. A packet for which the PLP bit is set has an increased probability of being dropped during congestion.

RELATED DOCUMENTATION

Classification Overview | 14

Default Behavior Aggregate Classification | 19

Sample Behavior Aggregate Classification | 21

Example: Configure and Apply Custom BA Classifiers | 23

Default Behavior Aggregate Classification

Table 7 on page 19 shows the forwarding class (FC) and packet loss priority (PLP) that are assigned by default to each well-known Differentiated Services (DiffServ) code point (DSCP). Although several DSCPs map to the expedited-forwarding (ef) and assured-forwarding (af) classes, by default no resources are assigned to these forwarding classes. All af classes other than af1*x* are mapped to best-effort, because RFC 2597, *Assured Forwarding PHB Group*, prohibits a node from aggregating classes. Assignment to the best-effort FC implies that the node does not support that class. You can modify the default settings through configuration.

Table 7: Default Behavior Aggregate Classification

DSCP and DSCP IPv6 Alias	Forwarding Class	Packet Loss Priority
ef	expedited-forwarding	low
af11	assured-forwarding	low
af12	assured-forwarding	high
af13	assured-forwarding	high
af21	best-effort	low
af22	best-effort	low
af23	best-effort	low
af31	best-effort	low

Table 7: Default Behavior Aggregate Classification ***(Continued)***

DSCP and DSCP IPv6 Alias	Forwarding Class	Packet Loss Priority
af32	best-effort	low
af33	best-effort	low
af41	best-effort	low
af42	best-effort	low
af43	best-effort	low
be	best-effort	low
cs1	best-effort	low
cs2	best-effort	low
cs3	best-effort	low
cs4	best-effort	low
cs5	best-effort	low
nc1/cs6	network-control	low
nc2/cs7	network-control	low
other	best-effort	low

RELATED DOCUMENTATION

Classification Overview | 14

Sample Behavior Aggregate Classification | 21

Example: Configure and Apply Custom BA Classifiers | 23

Understanding Packet Loss Priorities | 18

Sample Behavior Aggregate Classification

Table 8 on page 21 shows the device forwarding classes (FCs) associated with each well-known Differentiated Services (DiffServ) code point (DSCP) and the resources assigned to the output queues for a sample DiffServ CoS implementation. This example assigns expedited forwarding to queue 1 and a subset of the assured FCs (af*x*) to queue 2, and distributes resources among all four forwarding classes. Other DiffServ-based implementations are possible.

Table 8: Sample Behavior Aggregate Classification Forwarding Classes and Queues

DSCP and DSCP IPv6 Alias	DSCP and DSCP IPv6 Bits	Forwarding Class	Packet Loss Priority	Queue
ef	101110	expedited-forwarding	low	1
af11	001010	assured-forwarding	low	2
af12	001100	assured-forwarding	high	2
af13	001110	assured-forwarding	high	2
af21	010010	best-effort	low	0
af22	010100	best-effort	low	0
af23	010110	best-effort	low	0
af31	011010	best-effort	low	0

Table 8: Sample Behavior Aggregate Classification Forwarding Classes and Queues ***(Continued)***

DSCP and DSCP IPv6 Alias	DSCP and DSCP IPv6 Bits	Forwarding Class	Packet Loss Priority	Queue
af32	011100	best-effort	low	0
af33	011110	best-effort	low	0
af41	100010	best-effort	low	0
af42	100100	best-effort	low	0
af43	100110	best-effort	low	0
be	000000	best-effort	low	0
cs1	001000	best-effort	low	0
cs2	010000	best-effort	low	0
cs3	011000	best-effort	low	0
cs4	100000	best-effort	low	0
cs5	101000	best-effort	low	0
nc1/cs6	110000	network-control	low	3
nc2/cs7	111000	network-control	low	3
other	–	best-effort	low	0

RELATED DOCUMENTATION

Classification Overview | 14
Default Behavior Aggregate Classification | 19
Example: Configure and Apply Custom BA Classifiers | 23
Understanding Packet Loss Priorities | 18

Example: Configure and Apply Custom BA Classifiers

IN THIS SECTION

- Requirements | 23
- Overview | 23
- Configuration | 24
- Verification | 27

This example shows how to configure BA classifiers for a device to determine forwarding treatment of packets.

Requirements

Before you begin, determine the default forwarding class and PLP for each well-known DSCP for which you want to configure for a BA classifier. See "Default Behavior Aggregate Classification" on page 19.

Overview

You configure BA classifiers to classify packets that contain valid DSCPs to appropriate queues. Once configured, you must apply the BA classifier to the correct interfaces. You can override the default IP precedence classifier by defining a classifier and applying it to a logical interface. To define new classifiers for all code point types, include the `classifiers` statement at the `[edit class-of-service]` hierarchy level.

In this example, you set the DSCP BA classifier to `ba-classifier` as the default DSCP map. You set:

- A best-effort forwarding class as `be-class`
- An expedited forwarding class as `ef-class`

- An assured forwarding class as `af-class`
- A network control forwarding class as `nc-class`

Finally, you apply the BA classifier to an interface called ge-0/0/0.

Table 9 on page 24 shows how the behavior aggregate classifier assigns loss priorities to incoming packets in the four forwarding classes.

Table 9: Sample ba-classifier Loss Priority Assignments

ba-classifier Forwarding Class	For CoS Traffic Type	ba-classifier Assignments
`be-class`	Best-effort traffic	High-priority code point: `000001`
`ef-class`	Expedited forwarding traffic	High-priority code point: `101111`
`af-class`	Assured forwarding traffic	High-priority code point: `001100`
`nc-class`	Network control traffic	High-priority code point: `110001`

Configuration

IN THIS SECTION

- Procedure | 24

Procedure

CLI Quick Configuration

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

```
set class-of-service classifiers dscp ba-classifier import default
set class-of-service classifiers dscp ba-classifier forwarding-class be-class loss-priority high
```

```
code-points 000001
set class-of-service classifiers dscp ba-classifier forwarding-class ef-class loss-priority high
code-points 101111
set class-of-service classifiers dscp ba-classifier forwarding-class af-class loss-priority high
code-points 001100
set class-of-service classifiers dscp ba-classifier forwarding-class nc-class loss-priority high
code-points 110001
set class-of-service interfaces ge-0/0/0 unit 0 classifiers dscp ba-classifier
```

Step-by-Step Procedure

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.

To configure behavior aggregate classifiers for a device:

1. Configure the class of service.

    ```
    [edit]
    user@host# edit class-of-service
    ```

2. Configure behavior aggregate classifiers for DiffServ CoS.

    ```
    [edit class-of-service]
    user@host# edit classifiers dscp ba-classifier
    user@host# set import default
    ```

3. Configure a best-effort forwarding class classifier.

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

4. Configure an expedited forwarding class classifier.

    ```
    [edit class-of-service classifiers dscp ba-classifier]
    user@host# set forwarding-class ef-class loss-priority high code-points 101111
    ```

5. Configure an assured forwarding class classifier.

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

6. Configure a network control forwarding class classifier.

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

7. Apply the behavior aggregate classifier to an interface.

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

NOTE: You can use interface wildcards for `interface-name` and `logical-unit-number`.

Results

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

```
[edit]
user@host# show class-of-service
classifiers {
    dscp ba-classifier {
        import default;
        forwarding-class be-class {
            loss-priority high code-points 000001;
        }
        forwarding-class ef-class {
            loss-priority high code-points 101111;
        }
        forwarding-class af-class {
            loss-priority high code-points 001100;
        }
        forwarding-class nc-class {
            loss-priority high code-points 110001;
        }
    }
    interfaces {
        ge-0/0/0 {
            unit 0 {
                classifiers {
                    dscp ba-classifier;
                }
            }
        }
```

If you are done configuring the device, enter `commit` from configuration mode.

Verification

IN THIS SECTION

- Verify the DSCP Classifier | 27
- Verify That the Classifier Is Applied to the Interfaces | 30

Confirm that the configuration is working properly.

Verify the DSCP Classifier

Purpose

Make sure that the DSCP classifier is configured as expected.

Action

Run the `show class-of-service classifiers name ba-classifier` command.

```
user@host> show class-of-service classifiers name ba-classifier


Classifier: ba-classifier, Code point type: dscp, Index: 10755
```

```
Code point          Forwarding class                    Loss priority
000000              best-effort                         low
000001              be-class                            high
000010              best-effort                         low
000011              best-effort                         low
000100              best-effort                         low
000101              best-effort                         low
000110              best-effort                         low
000111              best-effort                         low
001000              best-effort                         low
001001              best-effort                         low
001010              assured-forwarding                  low
001011              best-effort                         low
001100              af-class                            high
001101              best-effort                         low
001110              assured-forwarding                  high
001111              best-effort                         low
010000              best-effort                         low
010001              best-effort                         low
010010              best-effort                         low
010011              best-effort                         low
010100              best-effort                         low
010101              best-effort                         low
010110              best-effort                         low
010111              best-effort                         low
011000              best-effort                         low
011001              best-effort                         low
011010              best-effort                         low
011011              best-effort                         low
011100              best-effort                         low
011101              best-effort                         low
011110              best-effort                         low
011111              best-effort                         low
100000              best-effort                         low
100001              best-effort                         low
100010              best-effort                         low
100011              best-effort                         low
100100              best-effort                         low
100101              best-effort                         low
100110              best-effort                         low
100111              best-effort                         low
101000              best-effort                         low
101001              best-effort                         low
```

```
  101010           best-effort                      low
  101011           best-effort                      low
  101100           best-effort                      low
  101101           best-effort                      low
  101110           expedited-forwarding             low
  101111           ef-class                         high
  110000           network-control                  low
  110001           nc-class                         high
  110010           best-effort                      low
  110011           best-effort                      low
  110100           best-effort                      low
  110101           best-effort                      low
  110110           best-effort                      low
  110111           best-effort                      low
  111000           network-control                  low
  111001           best-effort                      low
  111010           best-effort                      low
  111011           best-effort                      low
  111100           best-effort                      low
  111101           best-effort                      low
  111110           best-effort                      low
  111111           best-effort                      low
```

Meaning

Notice that the default classifier is incorporated into the customer classifier. If you were to remove the `import default` statement from the custom classifier, the custom classifier would look like this:

```
user@host> show class-of-service classifier name ba-classifier
Classifier: ba-classifier, Code point type: dscp, Index: 10755
  Code point         Forwarding class                 Loss priority
  000001             be-class                         high
  001100             af-class                         high
  101111             ef-class                         high
  110001             nc-class                         high
```

Verify That the Classifier Is Applied to the Interfaces

Purpose

Make sure that the classifier is applied to the correct interfaces.

Action

Run the `show class-of-service interface` command.

```
user@host> show class-of-service interface ge-0/0/0



Physical interface: ge-0/0/0, Index: 144
Queues supported: 8, Queues in use: 4
  Scheduler map: <default>, Index: 2
  Congestion-notification: Disabled

  Logical interface: ge-0/0/0.0, Index: 333
Object                  Name                  Type                    Index
Classifier              ba-classifier         dscp                    10755
```

Meaning

The interface is configured as expected.

RELATED DOCUMENTATION

- Interfaces User Guide for Security Devices
- Classification Overview | 14
- Sample Behavior Aggregate Classification | 21
- Understanding Packet Loss Priorities | 18

Apply MPLS EXP Classifiers to Routing Instances

IN THIS SECTION

- Configure and Apply Custom MPLS EXP Classifiers to Routing Instances | 32
- Apply Global Classifiers and Wildcard Routing Instances | 33
- Apply Global MPLS EXP Classifiers to Routing Instances | 34
- Apply Classifiers by Using Wildcard Routing Instances | 36
- Verify the Classifiers Associated with Routing Instances | 37

This topic shows how to apply MPLS EXP classifiers to routing instances.

When you enable VRF table labels and you do not explicitly apply a classifier configuration to the routing instance, Junos applies the default MPLS EXP classifier to the routing instance. For detailed information about VRF table labels, see the Junos OS VPNs Library for Routing Devices.

Table 10 on page 31 show the default MPLS EXP classification table.

Table 10: Default MPLS EXP Classifier

MPLS EXP Bits	Forwarding Class	Loss Priority
`000`	`best-effort`	`low`
`001`	`best-effort`	`high`
`010`	`expedited-forwarding`	`low`
`011`	`expedited-forwarding`	`high`
`100`	`assured-forwarding`	`low`
`101`	`assured-forwarding`	`high`

Table 10: Default MPLS EXP Classifier *(Continued)*

MPLS EXP Bits	Forwarding Class	Loss Priority
`110`	`network-control`	`low`
`111`	`network-control`	`high`

NOTE: At times, you might need to maintain the original classifier—for example with bridge domains, where you neither want to configure a custom classifier for the routing instance nor accept the default classifier. You can maintain the original MPLS EXP classifier. To do so, apply the `no-default` option for the routing instance. For example:

```
[edit class-of-service]
routing-instances routing-instance-name {
    classifiers {
        no-default;
    }
}
```

Configure and Apply Custom MPLS EXP Classifiers to Routing Instances

For routing instances with VRF table labels enabled, you can override the default MPLS EXP classifier and apply a custom classifier to a routing instance.

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

To apply a custom classifier to a routing instance:

1. Filter traffic based on the IP header.

```
[edit]
user@host# edit routing-instances routing-instance-name
user@host# set vrf-table-label
```

2. Configure the custom MPLS EXP classifier.

```
[edit]
user@host# edit class-of-service
user@host# set classifiers exp classifier-name import classifier-name forwarding-class class-
name loss-priority level code-points (aliases | bit-patterns)
user@host# set forwarding-classes queue queue-number class-name priority (high | low)
```

3. Apply the custom MPLS EXP classifier to the routing instance.

```
[edit class-of-service routing-instances routing-instance-name classifiers]
user@host# set exp classifier-name;
```

4. Commit and confirm your configuration.

```
[edit]
user@host# show class-of-service routing-instances
```

Apply Global Classifiers and Wildcard Routing Instances

To apply a classifier to all routing instances:

- Specify that the MPLS EXP classifier is for all routing instances.

```
[edit class-of-service ]
user@host# set routing-instances all classifiers exp classifier-name
```

For routing instances you associate with specific classifiers, Junos ignores the global configuration.

To use a wildcard to apply a classifier to all routing instances:

- Include an asterisk (*) in the name of the routing instance.

```
[edit]]
user@host# edit class-of-service routing-instances routing-instance-name*
user@host# set classifiers exp classifier-name
```

The wildcard configuration follows the longest match. If you include a specific configuration, it has precedence over the wildcard configuration.

NOTE: Junos supports the wildcard * and the `all` keyword at the `[edit class-of-service routing-instances]` hierarchy level but not at the `[edit routing-instances]` hierarchy level.

If you configure a routing instance at the `[edit routing-instances]` hierarchy level with, for example, the name `vpn*`, Junos treats `vpn*` as a valid and distinct routing instance name. If you then try to apply a classifier to the `vpn*` routing instance at the `[edit class-of-service routing-instances]` hierarchy level, Junos treats the `vpn*` routing instance name as a wildcard. Thus all routing instances that start with `vpn` and do not have a specific classifier applied receive the classifier associated with `vpn*`.

This same behavior applies with the `all` keyword.

Note that the * wildcard *must* be appended to an instance name at these configuration levels. The * wildcard is not a stand-alone substitute for the `all` keyword.

Apply Global MPLS EXP Classifiers to Routing Instances

This example shows how to apply a global classifier to all routing instances and then override the global classifier for a specific routing instance. For this example, we have three routing instances: `vpn1`, `vpn2`, and `vpn3`, each with VRF table label enabled. The classifier `exp-classifier-global` is applied to `vpn1` and `vpn2` (that is, all but `vpn3`, which is listed separately). The classifier `exp-classifier-3` is applied to `vpn3`.

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

To configure a global classifier for all routing instances and override the global classifier for a specific routing instance:

1. Enable the VRF table label for all three routing instances.

```
[edit routing-instances]
user@host#  set vpn1 vrf-table-label
user@host#  set vpn2 vrf-table-label
user@host#  set vpn3 vrf-table-label
```

2. Apply the EXP classifier `exp-classifier-global` to all routing instances.

```
[edit class-of-service routing-instances]
user@host# set all classifiers exp exp-classifier-global
```

3. Apply the EXP classifier `exp-classifier-3` to only the routing-instance `vpn3`.

```
[edit class-of-service routing-instances]
user@host# set vpn3 classifiers exp exp-classifier-3
```

4. Confirm your configuration.

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

```
vpn1 {
    vrf-table-label;
}
vpn2 {
    vrf-table-label;
}
vpn3 {
    vrf-table-label;
}
```

```
[edit class-of-service routing-instances]
user@host# show
```

```
all {
    classifiers {
        exp exp-classifier-global;
    }
}
vpn3 {
    classifiers {
        exp exp-classifier-3;
    }
}
```

Apply Classifiers by Using Wildcard Routing Instances

Configure a wildcard routing instance and override the wildcard with a specific routing instance. For this example, we have three routing instances: `vpn-red`, `vpn-yellow`, and `vpn-green`, each with VRF table label enabled. The classifier `exp-class-wildcard` is applied to `vpn-yellow` and `vpn-green`. The classifier `exp-class-red` is applied to `vpn-red`.

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

To configure a wildcard routing instance and override the wildcard with a specific routing instance:

1. Enable the VRF table label for all three routing instances.

```
[edit routing-instances]
user@host#  set vpn-red vrf-table-label
user@host#  set vpn-yellow vrf-table-label
user@host#  set vpn-green vrf-table-label
```

2. Apply the EXP classifier `exp-class-wildcard` to all routing instances by using a wildcard.

```
[edit class-of-service routing-instances]
user@host# set vpn* classifiers exp exp-class-wildcard
```

3. Apply the EXP classifier `exp-class-red` to only the routing-instance `vpn-red`.

```
[edit class-of-service routing-instances]
user@host# set vpn-red classifiers exp exp-class-red
```

4. Commit and confirm your configuration.

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

```
vpn-red {
    vrf-table-label;
}
vpn-yellow {
    vrf-table-label;
}
vpn-green {
    vrf-table-label;
}
```

```
[edit class-of-service routing-instances]
user@host# show
```

```
vpn* {
    classifiers {
        exp exp-class-wildcard;
    }
}
vpn-red {
    classifiers {
        exp exp-class-red;
    }
}
```

Verify the Classifiers Associated with Routing Instances

IN THIS SECTION

- Purpose | 37
- Action | 38
- Platform-Specific Behavior | 38

Purpose

Display the MPLS EXP classifiers associated with two routing instances:

Action

To verify the MPLS EXP classifiers associated with two routing instances, enter the following Junos CLI operational mode command:

```
user@host> show class-of-service routing-instances
  Routing Instance : vpn1
    Object           Name                    Type                      Index
    Classifier       exp-default             exp                           8

  Routing Instance : vpn2
    Object           Name                    Type                      Index
    Classifier       class2                  exp                       57507
```

Platform-Specific Behavior

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

Table 11: Platform-Specific Behavior

Platform	Difference
MX Series devices	• On MX Series devices only, you can maintain the original MPLS EXP classifier.

RELATED DOCUMENTATION

Configuring Behavior Aggregate Classifiers

Default MPLS EXP Classifier

Applying MPLS EXP Classifiers for Explicit-Null Labels

CHAPTER 3

Controlling Network Access with Traffic Policing

IN THIS CHAPTER

- Simple Filters and Policers Overview | 39
- Two-Rate Three-Color Policer Overview | 41
- Example: Configuring a Two-Rate Three-Color Policer | 42
- Logical Interface (Aggregate) Policer Overview | 50
- Two-Color Policer Configuration Overview | 50
- Example: Configuring a Two-Color Logical Interface (Aggregate) Policer | 55
- Guidelines for Configuring Simple Filters | 64
- Example: Configure and Apply a Firewall Filter for a Multifield Classifier | 68

Simple Filters and Policers Overview

IN THIS SECTION

- Platform-Specific Simple Filter Behavior | 40

You can configure simple filters and policers to handle oversubscribed traffic in SRX Series firewalls. In Junos OS, policers can be configured as part of the *firewall filter* hierarchy. (Platform support depends on the Junos OS release in your installation.)

The simple filter functionality consists of the following:

- Classifying packets according to configured policies
- Taking appropriate actions based on the results of classification

In Junos OS, ingress traffic policers can limit the rate of incoming traffic. The main reasons to use traffic policing are:

- To enforce traffic rates to conform to the service-level agreement (SLA)
- To protect next hops, such as protecting the central point and the SPU from being overwhelmed by excess traffic like DOS attacks

Using the results of packet classification and traffic metering, a policer can take one of the following actions for a packet: forward a conforming (green) packet or drop a nonconforming (yellow) packet. Policers always discard a nonconforming red packet. Traffic metering supports the algorithm of the two-rate tricolor marker (TCM). (See RFC 2698, A *Two Rate Three Color Marker*.)

Platform-Specific Simple Filter Behavior

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

Table 12: Platform-Specific Simple Filter Behavior

Platform	Difference
SRX5000 Series	• For SRX5000 Series firewalls with FIOC, use of ingress traffic policers can prevent the central point and the SPU from being overwhelmed by traffic, for example in a DDoS attack.
SRX5600 and SRX5800	• For SRX5600 and SRX5800 firewalls, you can apply the simple filter or policing actions only to logical interfaces residing in an SRX5000 line Flex IOC (FIOC), because only an SRX5000 line FIOC supports the simple filter and policing features on the SRX5600 and SRX5800 firewalls.

RELATED DOCUMENTATION

Guidelines for Configuring Simple Filters | 64

Example: Configuring a Two-Rate Three-Color Policer | 42

Example: Configure and Apply a Firewall Filter for a Multifield Classifier | 68

Two-Rate Three-Color Policer Overview

A two-rate three-color policer defines two bandwidth limits (one for guaranteed traffic and one for peak traffic) and two burst sizes (one for each of the bandwidth limits). A two-rate three-color policer is most useful when a service is structured according to arrival rates and not necessarily packet length.

Two-rate three-color policing meters a traffic stream based on the following configured traffic criteria:

- Committed information rate (CIR)—Bandwidth limit for guaranteed traffic.
- Committed burst size (CBS)—Maximum packet size permitted for bursts of data that exceed the CIR.
- Peak information rate (PIR)—Bandwidth limit for peak traffic.
- Peak burst size (PBS)—Maximum packet size permitted for bursts of data that exceed the PIR.

Two-rate tricolor marking (two-rate TCM) classifies traffic as belonging to one of three color categories and performs congestion-control actions on the packets based on the color marking:

- Green—Traffic that conforms to the bandwidth limit and burst size for guaranteed traffic (CIR and CBS). For a green traffic flow, two-rate TCM marks the packets with an implicit loss priority of `low` and transmits the packets.
- Yellow—Traffic that exceeds the bandwidth limit or burst size for guaranteed traffic (CIR or CBS) but not the bandwidth limit and burst size for peak traffic (PIR and PBS). For a yellow traffic flow, two-rate TCM marks packets with an implicit loss priority of `medium-high` and transmits the packets.
- Red—Traffic that exceeds the bandwidth limit and burst size for peak traffic (PIR and PBS). For a red traffic flow, two-rate TCM marks packets with an implicit loss priority of `high` and, optionally, discards the packets.

If congestion occurs downstream, the packets with higher loss priority are more likely to be discarded.

NOTE: For both single-rate and two-rate three-color policers, the only *configurable* action is to discard packets in a red traffic flow.

For a tricolor marking policer referenced by a *firewall filter* term, the `discard` policing action is supported on the following routing platforms:

- EX Series switches
- M7i and M10i routers with the Enhanced CFEB (CFEB-E)
- M120 and M320 routers with Enhanced-III FPCs
- MX Series routers with Trio MPCs

To apply a tricolor marking policer on these routing platforms, it is not necessary to include the `logical-interface-policer` statement.

RELATED DOCUMENTATION

> *Example: Configuring a Two-Rate Three-Color Policer*

Example: Configuring a Two-Rate Three-Color Policer

IN THIS SECTION

- Requirements | **42**
- Overview | **42**
- Configuration | **43**
- Verification | **48**

This example shows how to configure a two-rate three-color policer.

Requirements

Support for two-rate three-color policers varies according to the device.

No special configuration beyond device initialization is required before configuring this example.

Overview

IN THIS SECTION

- Topology | **43**

A two-rate three-color policer meters a traffic flow against a bandwidth limit and burst-size limit for guaranteed traffic, plus a bandwidth limit and burst-size limit for peak traffic. Traffic that conforms to the limits for guaranteed traffic is categorized as green, and nonconforming traffic falls into one of two categories:

- Nonconforming traffic that does not exceed peak traffic limits is categorized as yellow.
- Nonconforming traffic that exceeds peak traffic limits is categorized as red.

Each category is associated with an action. For green traffic, packets are implicitly set with a loss-priority value of `low` and then transmitted. For yellow traffic, packets are implicitly set with a loss-priority value of `medium-high` and then transmitted. For red traffic, packets are implicitly set with a loss-priority value of `high` and then transmitted. If the policer configuration includes the optional `action` statement (`action loss-priority high then discard`), then packets in a red flow are discarded instead.

You can apply a three-color policer to Layer 3 traffic as a firewall filter policer only. You reference the policer from a stateless firewall filter term, and then you apply the filter to the input or output of a logical interface at the protocol level.

Topology

In this example, you apply a color-aware, two-rate three-color policer to the input IPv4 traffic at logical interface `fe-0/1/1.0`. The IPv4 firewall filter term that references the policer does not apply any packet-filtering. The filter is used only to apply the three-color policer to the interface.

You configure the policer to rate-limit traffic to a bandwidth limit of 40 Mbps and a burst-size limit of 100 KB for green traffic, and you configure the policer to also allow a peak bandwidth limit of 60 Mbps and a peak burst-size limit of 200 KB for yellow traffic. Only nonconforming traffic that exceeds the peak traffic limits is categorized as red. In this example, you configure the three-color policer action `loss-priority high then discard`, which overrides the implicit marking of red traffic to a `high` loss priority.

Configuration

IN THIS SECTION

- CLI Quick Configuration | 44
- Configuring a Two-Rate Three-Color Policer | 44
- Configuring an IPv4 Stateless Firewall Filter That References the Policer | 46
- Applying the Filter to a Logical Interface at the Protocol Family Level | 47

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

To configure this example, perform the following tasks:

CLI Quick Configuration

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

```
set firewall three-color-policer trTCM1-ca two-rate color-aware
set firewall three-color-policer trTCM1-ca two-rate committed-information-rate 40m
set firewall three-color-policer trTCM1-ca two-rate committed-burst-size 100k
set firewall three-color-policer trTCM1-ca two-rate peak-information-rate 60m
set firewall three-color-policer trTCM1-ca two-rate peak-burst-size 200k
set firewall three-color-policer trTCM1-ca action loss-priority high then discard
set firewall family inet filter filter-trtcm1ca-all term 1 then three-color-policer two-rate
trTCM1-ca
set interfaces ge-2/0/5 unit 0 family inet address 10.10.10.1/30
set interfaces ge-2/0/5 unit 0 family inet filter input filter-trtcm1ca-all
set class-of-service interfaces ge-2/0/5 forwarding-class af
```

Configuring a Two-Rate Three-Color Policer

Step-by-Step Procedure

To configure a two-rate three-color policer:

1. Enable configuration of a three-color policer.

   ```
   [edit]
   user@host# set firewall three-color-policer trTCM1-ca
   ```

2. Configure the color mode of the two-rate three-color policer.

   ```
   [edit firewall three-color-policer trTCM1-ca]
   user@host# set two-rate color-aware
   ```

3. Configure the two-rate guaranteed traffic limits.

   ```
   [edit firewall three-color-policer trTCM1-ca]
   user@host# set two-rate committed-information-rate 40m
   user@host# set two-rate committed-burst-size 100k
   ```

   Traffic that does not exceed both of these limits is categorized as green. Packets in a green flow are implicitly set to `low` loss priority and then transmitted.

4. Configure the two-rate peak traffic limits.

   ```
   [edit firewall three-color-policer trTCM1-ca]
   user@host# set two-rate peak-information-rate 60m
   user@host# set two-rate peak-burst-size 200k
   ```

   Nonconforming traffic that does not exceed both of these limits is categorized as yellow. Packets in a yellow flow are implicitly set to `medium-high` loss priority and then transmitted. Nonconforming traffic that exceeds both of these limits is categorized as red. Packets in a red flow are implicitly set to `high` loss priority.

5. (Optional) Configure the policer action for red traffic.

   ```
   [edit firewall three-color-policer trTCM1-ca]
   user@host# set action loss-priority high then discard
   ```

   For three-color policers, the only configurable action is to discard red packets. Red packets are packets that have been assigned high loss priority because they exceeded the peak information rate (PIR) and the peak burst size (PBS).

Results

Confirm the configuration of the policer by entering the `show firewall` configuration mode command. If the command output does not display the intended configuration, repeat the instructions in this procedure to correct the configuration.

```
[edit]
user@host# show firewall
three-color-policer trTCM1-ca {
    action {
        loss-priority high then discard;
```

```
    }
    two-rate {
        color-aware;
        committed-information-rate 40m;
        committed-burst-size 100k;
        peak-information-rate 60m;
        peak-burst-size 200k;
    }
}
```

Configuring an IPv4 Stateless Firewall Filter That References the Policer

Step-by-Step Procedure

To configure an IPv4 stateless firewall filter that references the policer:

1. Enable configuration of an IPv4 standard stateless firewall filter.

   ```
   [edit]
   user@host# set firewall family inet filter filter-trtcm1ca-all
   ```

2. Specify the filter term that references the policer.

   ```
   [edit firewall family inet filter filter-trtcm1ca-all]
   user@host# set term 1 then three-color-policer two-rate trTCM1-ca
   ```

   Note that the term does not specify any match conditions. The firewall filter passes all packets to the policer.

Results

Confirm the configuration of the firewall filter by entering the `show firewall` configuration mode command. If the command output does not display the intended configuration, repeat the instructions in this procedure to correct the configuration.

```
[edit]
user@host# show firewall
family inet {
    filter filter-trtcm1ca-all {
        term 1 {
```

```
            then {
                three-color-policer {
                    two-rate trTCM1-ca;
                }
            }
        }
    }
}
three-color-policer trTCM1-ca {
    action {
        loss-priority high then discard;
    }
    two-rate {
        color-aware;
        committed-information-rate 40m;
        committed-burst-size 100k;
        peak-information-rate 60m;
        peak-burst-size 200k;
    }
}
```

Applying the Filter to a Logical Interface at the Protocol Family Level

Step-by-Step Procedure

To apply the filter to the logical interface at the protocol family level:

1. Enable configuration of an IPv4 firewall filter.

```
[edit]
user@host# edit interfaces ge-2/0/5 unit 0 family inet
```

2. Apply the policer to the logical interface at the protocol family level.

```
[edit interfaces ge-2/0/5 unit 0 family inet]
user@host# set address 10.10.10.1/30
user@host# set filter input filter-trtcm1ca-all
```

3. (MX Series routers and EX Series switches only) (Optional) For input policers, you can configure a fixed classifier. A fixed classifier reclassifies all incoming packets, regardless of any preexisting classification.

   NOTE: Platform support depends on the Junos OS release in your implementation.

   ```
   [edit]
   user@host# set class-of-service interfaces ge-2/0/5 forwarding-class af
   ```

   The classifier name can be a configured classifier or one of the default classifiers.

Results

Confirm the configuration of the interface by entering the `show interfaces` configuration mode command. If the command output does not display the intended configuration, repeat the instructions in this procedure to correct the configuration.

```
[edit]
user@host# show interfaces
ge-2/0/5 {
    unit 0 {
        family inet {
            address 10.10.10.1/30;
            filter {
                input filter-trtcm1ca-all;
            }
        }
    }
}
```

If you are done configuring the device, enter `commit` from configuration mode.

Verification

IN THIS SECTION

- Displaying the Firewall Filters Applied to the Logical Interface | 49

Confirm that the configuration is working properly.

Displaying the Firewall Filters Applied to the Logical Interface

Purpose

Verify that the firewall filter is applied to IPv4 input traffic at the logical interface.

Action

Use the `show interfaces` operational mode command for the logical interface `ge-2/0/5.0`, and specify `detail` mode. The **Protocol inet** section of the command output displays IPv4 information for the logical interface. Within that section, the **Input Filters** field displays the name of IPv4 firewall filters associated with the logical interface.

```
user@host> show interfaces ge-2/0/5.0 detail
  Logical interface ge-2/0/5.0 (Index 105) (SNMP ifIndex 556) (Generation 170)
    Flags: Device-Down SNMP-Traps 0x4004000 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 inet, MTU: 1500, Generation: 242, Route table: 0
      Flags: Sendbcast-pkt-to-re
      Input Filters: filter-trtcm1ca-all
      Addresses, Flags: Dest-route-down Is-Preferred Is-Primary
        Destination: 10.20.130/24, Local: 10.20.130.1, Broadcast: 10.20.130.255,
        Generation: 171
    Protocol multiservice, MTU: Unlimited, Generation: 243, Route table: 0
      Policer: Input: __default_arp_policer__
```

RELATED DOCUMENTATION

Two-Rate Three-Color Policer Overview

Logical Interface (Aggregate) Policer Overview

A *logical interface policer*—also called an *aggregate policer*—is a two-color or three-color policer that defines traffic rate limiting. You can apply a policer to input or output traffic for multiple protocol families on the same logical interface without needing to create multiple instances of the policer.

To configure a single-rate two-color logical interface policer, include the `logical-interface-policer` statement at the `[edit firewall policer policer-name]` hierarchy level.

You apply a logical interface policer to Layer 3 traffic directly to the interface configuration at the protocol family level (to rate-limit traffic of a specific protocol family). You cannot reference a logical interface policer from a stateless firewall filter term and then apply the filter to a logical interface.

You can apply a logical interface policer to unicast traffic only. .

To display a logical interface policer on a particular interface, issue the `show interfaces policers` *operational mode command.*

RELATED DOCUMENTATION

Two-Color Policer Configuration Overview | **50**

Example: Configuring a Two-Color Logical Interface (Aggregate) Policer | **55**

logical-interface-policer

Two-Color Policer Configuration Overview

IN THIS SECTION

- Platform-Specific Policer Behavior | **55**

You can configure and apply single-rate two-color policers to Layer 3 traffic.

Table 13 on page 51 describes the hierarchy levels at which you can configure and apply two-color policers.

Table 13: Two-Color Policer Configuration and Application Overview

Policer Configuration	Layer 3 Application	Key Points
Single-Rate Two-Color Policer *Defines traffic rate limiting that you can apply to Layer 3 protocol-specific traffic at a logical interface. Can be applied as an interface policer or as a firewall filter policer.*		

Table 13: Two-Color Policer Configuration and Application Overview *(Continued)*

Policer Configuration	Layer 3 Application	Key Points
Basic policer configuration: (code below)	Method A—Apply as an interface policer at the protocol family level: (code below) Method B—Apply as a firewall filter policer at the protocol family level: (code below)	Policer configuration: • Use `bandwidth-limit` *`bps`* to specify an absolute value. Firewall filter configuration (*) • If applying to multiple interfaces, include the `interface-specific` statement to create unique policers and counters for each interface. Interface policer verification: • Use the `show interfaces (detail | extensive)` operational mode command. • Use the `show policer` operational mode command. Firewall filter policer verification: • Use the `show interfaces (detail | extensive)` operational mode command. • Use the `show firewall filter` *`filter-name`* operational mode command.

Basic policer configuration:

```
[edit firewall]
policer policer-name {
    if-exceeding {
        bandwidth-limit bps;
        burst-size-limit bytes;
    }
    then {
        discard;
        forwarding-class class-name;
        loss-priority supported-value;
    }
}
```

Method A—Apply as an interface policer at the protocol family level:

```
[edit interfaces]
interface-name {
    unit unit-number {
        family family-name {
            policer {
                input policer-name;
                output policer-name;
            }
        }
    }
}
```

Method B—Apply as a firewall filter policer at the protocol family level:

```
[edit firewall]
family family-name {
    filter filter-name {
        interface-specific; # (*)
        from {
            ... match-conditions ...
        }
        then {
            policer policer-name;
        }
    }
}
```

```
[edit interfaces]
interface-name {
    unit unit-number {
        family family-name {
            filter {
                input filter-name;
                output filter-name;
            }
            ... protocol-configuration ...
        }
```

Table 13: Two-Color Policer Configuration and Application Overview *(Continued)*

Policer Configuration	Layer 3 Application	Key Points
	`}` `}`	
Logical Interface (Aggregate) Policer *Defines traffic rate limiting that you can apply to multiple protocol families on the same logical interface without creating multiple instances of the policer. Can be applied directly to a logical interface configuration only.*		

Table 13: Two-Color Policer Configuration and Application Overview *(Continued)*

Policer Configuration	Layer 3 Application	Key Points
Logical interface policer configuration:	Method A—Apply as an interface policer only: Method B—Apply as a firewall filter policer at the protocol family level:	Policer configuration: • Include the `logical-interface-policer` statement. Two options for interface policer application: • To rate-limit all traffic types, regardless of the protocol family, apply the logical interface policer at the logical unit level. • To rate-limit traffic of a specific protocol family, apply the logical interface policer at the protocol family level. Interface policer verification: • Use the `show interfaces (detail | extensive)` operational mode command. • Use the `show policer` operational mode command.

```
[edit firewall]
policer policer-name {
    logical-interface-policer;
    if-exceeding {
        bandwidth-limit bps;
        burst-size-limit bytes;
    }
    then {
        discard;
        forwarding-class class-name;
        loss-priority supported-value;
    }
}
```

```
[edit interfaces]
interface-name {
    unit unit-number {
        policer { # All protocols
            input policer-name;
            output policer-name;
        }
        family family-name {
            policer { # One protocol
                input policer-name;
                output policer-name;
            }
        }
    }
}
```

```
[edit firewall]
family family-name {
    filter filter-name {
        interface-specific;
        term term-name{
            from {
                ... match-conditions ...
            }
        }
        then {
            policer policer-name;
        }
    }
}
```

Platform-Specific Policer Behavior

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

Table 14: Platform-Specific Policer Behavior

Platform	Difference
vSRX Virtual Firewall, SRX1500, SRX1600, SRX2300, SRX4100, SRX4120, SRX4200, and SRX4600 firewalls	We recommend limiting the traffic rate in a given policer to 80 Mbps or less.

RELATED DOCUMENTATION

logical-interface-policer

Example: Configuring a Two-Color Logical Interface (Aggregate) Policer | 55

Example: Configuring a Two-Color Logical Interface (Aggregate) Policer

IN THIS SECTION

- Requirements | 55
- Overview | 56
- Configuration | 56
- Verification | 62

You can configure a single-rate two-color policer as a logical interface policer and apply it to incoming IPv4 traffic on a logical interface. This example shows how to do to so.

Requirements

Before you begin, make sure that the logical interface to which you apply the two-color logical interface policer is hosted on at least a Gigabit Ethernet interface (`ge-`).

Overview

IN THIS SECTION

- Topology | 56

In this example, you configure the single-rate two-color policer `policer_IFL` as a logical interface policer and apply it to incoming IPv4 traffic at logical interface `ge-1/3/1.0`.

Topology

If the input IPv4 traffic on the physical interface `ge-1/3/1` exceeds the bandwidth limit equal to 90 percent of the media rate with a 300 KB burst-size limit, then the logical interface policer `policer_IFL` rate-limits the input IPv4 traffic on the logical interface `ge-1/3/1.0`. Configure the policer to mark nonconforming traffic by setting packet loss priority (PLP) levels to `high` and classifying packets as `best-effort`.

As the incoming IPv4 traffic rate on the physical interface slows and conforms to the configured limits, Junos OS stops marking the incoming IPv4 packets at the logical interface.

Configuration

IN THIS SECTION

- CLI Quick Configuration | 57
- Configuring the Logical Interfaces | 57
- Configuring the Single-Rate Two-Color Policer as a Logical Interface Policer | 59
- Applying the Logical Interface Policer to Input IPv4 Traffic at a Logical Interface | 61

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

To configure this example, perform the following tasks:

CLI Quick Configuration

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

```
set interfaces ge-1/3/1 vlan-tagging
set interfaces ge-1/3/1 unit 0 vlan-id 100
set interfaces ge-1/3/1 unit 0 family inet address 10.10.10.1/30
set interfaces ge-1/3/1 unit 1 vlan-id 101
set interfaces ge-1/3/1 unit 1 family inet address 20.20.20.1/30 arp 20.20.20.2 mac
00:00:11:22:33:44
set firewall policer policer_IFL logical-interface-policer
set firewall policer policer_IFL if-exceeding bandwidth-percent 90
set firewall policer policer_IFL if-exceeding burst-size-limit 300k
set firewall policer policer_IFL then loss-priority high
set firewall policer policer_IFL then forwarding-class best-effort
set interfaces ge-1/3/1 unit 0 family inet policer input policer_IFL
```

Configuring the Logical Interfaces

Step-by-Step Procedure

To configure the logical interfaces:

1. Enable configuration of the interface.

```
[edit]
user@host# edit interfaces ge-1/3/1
```

2. Configure single tagging.

```
[edit interfaces ge-1/3/1]
user@host# set vlan-tagging
```

3. Configure logical interface `ge-1/3/1.0`.

```
[edit interfaces ge-1/3/1]
user@host# set unit 0 vlan-id 100
user@host# set unit 0 family inet address 10.10.10.1/30
```

4. Configure logical interface `ge-1/3/1.0`.

```
[edit interfaces ge-1/3/1]
user@host# set unit 1 vlan-id 101
user@host# set unit 1 family inet address 20.20.20.1/30 arp 20.20.20.2 mac 00:00:11:22:33:44
```

Results

Confirm the configuration of the logical interfaces by entering the `show interfaces` configuration mode command. If the command output does not display the intended configuration, repeat the instructions in this procedure to correct the configuration.

```
[edit]
user@host# show interfaces
ge-1/3/1 {
    vlan-tagging;
    unit 0 {
        vlan-id 100;
        family inet {
            address 10.10.10.1/30;
        }
    }
    unit 1 {
        vlan-id 101;
        family inet {
            address 20.20.20.1/30 {
                arp 20.20.20.2 mac 00:00:11:22:33:44;
            }
        }
    }
}
```

Configuring the Single-Rate Two-Color Policer as a Logical Interface Policer

Step-by-Step Procedure

To configure a single-rate two-color policer as a logical interface policer:

1. Enable configuration of a single-rate two-color policer.

   ```
   [edit]
   user@host# edit firewall policer policer_IFL
   ```

2. Specify that the policer is a logical interface (aggregate) policer.

   ```
   [edit firewall policer policer_IFL]
   user@host# set logical-interface-policer
   ```

   A logical interface policer rate-limits traffic based on a percentage of the media rate of the physical interface underlying the logical interface to which the policer is applied. The policer is applied directly to the interface rather than referenced by a firewall filter.

3. Specify the policer traffic limits.

   - Specify the bandwidth limit.

     - To specify the bandwidth limit as an absolute rate, from 8,000 bits per second through 50,000,000,000 bits per second, include the `bandwidth-limit` *`bps`* statement.

     - To specify the bandwidth limit as a percentage of the physical port speed on the interface, include the `bandwidth-percent` *`percent`* statement.

     In this example, the CLI commands and output are based on a bandwidth limit specified as a percentage rather than as an absolute rate.

     ```
     [edit firewall policer policer_IFL]
     user@host# set if-exceeding bandwidth-percent 90
     ```

   - Specify the burst-size limit, from 1,500 bytes through 100,000,000,000 bytes, which is the maximum packet size to be permitted for bursts of data that exceed the specified bandwidth limit.

     ```
     [edit firewall policer policer_IFL]
     user@host# set if-exceeding burst-size-limit 300k
     ```

4. Specify the policer actions to be taken on traffic that exceeds the configured rate limits.

   - To discard the packet, include the `discard` statement.
   - To set the loss-priority value of the packet, include the `loss-priority (low | medium-low | medium-high | high)` statement.
   - To classify the packet to a forwarding class, include the `forwarding-class (`*`forwarding-class`*` | assured-forwarding | best-effort | expedited-forwarding | network-control)` statement.

   In this example, the CLI commands and output are based on both setting the packet loss priority level and classifying the packet.

   ```
   [edit firewall policer policer_IFL]
   user@host# set then loss-priority high
   user@host# set then forwarding-class best-effort
   ```

Results

Confirm the configuration of the policer by entering the `show firewall` configuration mode command. If the command output does not display the intended configuration, repeat the instructions in this procedure to correct the configuration.

```
[edit]
user@host# show firewall
policer policer_IFL {
    logical-interface-policer;
    if-exceeding {
        bandwidth-percent 90;
        burst-size-limit 300k;
    }
    then {
        loss-priority high;
        forwarding-class best-effort;
    }
}
```

Applying the Logical Interface Policer to Input IPv4 Traffic at a Logical Interface

Step-by-Step Procedure

To apply the two-color logical interface policer to input IPv4 traffic a logical interface:

1. Enable configuration of the logical interface.

```
[edit]
user@host# edit interfaces ge-1/3/1 unit 0
```

2. Apply the policer to all traffic types or to a specific traffic type on the logical interface.

   - To apply the policer to all traffic types, regardless of the protocol family, include the `policer (input | output)` *`policer-name`* statement at the `[edit interfaces` *`interface-name`* `unit` *`number`*`]` hierarchy level.

   - To apply the policer to traffic of a specific protocol family, include the `policer (input | output)` *`policer-name`* statement at the `[edit interfaces` *`interface-name`* `unit` *`unit-number`* `family` *`family-name`*`]` hierarchy level.

   To apply the logical interface policer to incoming packets, use the `policer input` *`policer-name`* statement. To apply the logical interface policer to outgoing packets, use the `policer output` *`policer-name`* statement.

   In this example, the CLI commands and output are based on rate-limiting the IPv4 input traffic at logical interface `ge-1/3/1.0`.

```
[edit interfaces ge-1/3/1 unit 0]
user@host# set family inet policer input policer_IFL
```

Results

Confirm the configuration of the interface by entering the `show interfaces` configuration mode command. If the command output does not display the intended configuration, repeat the instructions in this procedure to correct the configuration.

```
[edit]
user@host# show interfaces
ge-1/3/1 {
    vlan-tagging;
```

```
        unit 0 {
            vlan-id 100;
            family inet {
                policer input policer_IFL;
                address 10.10.10.1/30;
            }
        }
        unit 1 {
            vlan-id 101;
            family inet {
                address 20.20.20.1/30 {
                    arp 20.20.20.2 mac 00:00:11:22:33:44;
                }
            }
        }
    }
```

If you are done configuring the device, enter `commit` from configuration mode.

Verification

IN THIS SECTION

- Displaying Traffic Statistics and Policers for the Logical Interface | 62
- Displaying Statistics for the Policer | 63

Confirm that the configuration is working properly.

Displaying Traffic Statistics and Policers for the Logical Interface

Purpose

Verify the traffic flow through the logical interface and that the policer is evaluating packets received on the logical interface.

Action

Use the `show interfaces` operational mode command for logical interface `ge-1/3/1.0`, and include the `detail` or `extensive` option. The command output section for **Traffic statistics** lists the number of bytes and packets received and transmitted on the logical interface. The **Protocol inet** subsection contains a **Policer** field that would list the policer `policer_IFL` as an input or output logical interface policer as follows:

- **Input: policer_IFL-ge-1/3/1.0-log_int-i**
- **Output: policer_IFL-ge-1/3/1.0-log_int-o**

The **log_int-i** suffix denotes a logical interface policer applied to input traffic, while the **log_int-o** suffix denotes a logical interface policer applied to output traffic. In this example, the logical interface policer is applied to input traffic only.

Displaying Statistics for the Policer

Purpose

Verify the number of packets evaluated by the policer.

Action

Use the `show policer` operational mode command and, optionally, specify the name of the policer. The command output displays the number of packets evaluated by each configured policer (or the specified policer), in each direction. For the policer `policer_IFL`, the input and output policer names are displayed as follows:

- **policer_IFL-ge-1/3/1.0-log_int-i**
- **policer_IFL-ge-1/3/1.0-log_int-o**

The **log_int-i** suffix denotes a logical interface policer applied to input traffic, while the **log_int-o** suffix denotes a logical interface policer applied to output traffic. In this example, the logical interface policer is applied to input traffic only.

RELATED DOCUMENTATION

logical-interface-policer

Two-Color Policer Configuration Overview | 50

Guidelines for Configuring Simple Filters

IN THIS SECTION

- Statement Hierarchy for Configuring Simple Filters | **64**
- Simple Filter Protocol Families | **65**
- Simple Filter Names | **65**
- Simple Filter Terms | **65**
- Simple Filter Match Conditions | **65**
- Simple Filter Terminating Actions | **67**
- Simple Filter Nonterminating Actions | **67**

This topic covers the following information:

Statement Hierarchy for Configuring Simple Filters

To configure a simple filter, include the `simple-filter` *`simple-filter-name`* statement at the `[edit firewall family inet]` hierarchy level.

```
[edit]
firewall {
    family inet {
        simple-filter  simple-filter-name {
            term term-name {
                from {
                    match-conditions;
                }
                then {
                    actions;
                }
            }
        }
    }
}
```

Individual statements supported under the `simple-filter` *`simple-filter-name`* statement are described separately in this topic and are illustrated in the example of configuring and applying a simple filter.

Simple Filter Protocol Families

You can configure simple filters to filter IPv4 traffic (`family inet`) only. No other protocol family is supported for simple filters.

> **NOTE**: You can apply simple filters to the family inet only, and only in the input direction.

Simple Filter Names

Under the `family inet` statement, you can include `simple-filter` *`simple-filter-name`* statements to create and name simple filters. The filter name can contain letters, numbers, and hyphens (-) and be up to 64 characters long. To include spaces in the name, enclose the entire name in quotation marks (“ ”).

Simple Filter Terms

Under the `simple-filter` *`simple-filter-name`* statement, you can include `term` *`term-name`* statements to create and name filter terms.

- You must configure at least one term in a *firewall filter*.
- You must specify a unique name for each term within a firewall filter. The term name can contain letters, numbers, and hyphens (-) and can be up to 64 characters long. To include spaces in the name, enclose the entire name in quotation marks (“ ”).
- The order in which you specify terms within a firewall filter configuration is important. Firewall filter terms are evaluated in the order in which they are configured. By default, new terms are always added to the end of the existing filter. You can use the `insert` configuration mode command to reorder the terms of a firewall filter.

Simple filters do *not* support the `next term` action.

Simple Filter Match Conditions

Simple filter terms support only a subset of the IPv4 match conditions that are supported for standard stateless firewall filters.

Unlike standard stateless firewall filters, the following restrictions apply to simple filters:

- Simple filters support only one `source-address` and one `destination-address` prefix for each filter term. If you configure multiple prefixes, only the last one is used.

- Simple filters do *not* support multiple source addresses and destination addresses in a single term. If you configure multiple addresses, only the last one is used.
- Simple filters do *not* support negated match conditions, such as the `protocol-except` match condition or the `exception` keyword.
- Simple filters support a range of values for `source-port` and `destination-port` match conditions only. For example, you can configure `source-port 400-500` or `destination-port 600-700`.
- Simple filters do *not* support noncontiguous mask values.

Table 15 on page 66 lists the simple filter match conditions.

Table 15: Simple Filter Match Conditions

Match Condition	Description
`destination-address` *`destination-address`*	Match IP destination address.
`destination-port` *`number`*	TCP or UDP destination port field. If you configure this match condition, we recommend that you also configure the `protocol` match statement to determine which protocol is being used on the port. In place of the numeric value, you can specify one of the following text aliases (the port numbers are also listed): `afs` (1483), `bgp` (179), `biff` (512), `bootpc` (68), `bootps` (67), `cmd` (514), `cvspserver` (2401), `dhcp` (67), `domain` (53), `eklogin` (2105), `ekshell` (2106), `exec` (512), `finger` (79), `ftp` (21), `ftp-data` (20), `http` (80), `https` (443), `ident` (113), `imap` (143), `kerberos-sec` (88), `klogin` (543), `kpasswd` (761), `krb-prop` (754), `krbupdate` (760), `kshell` (544), `ldap` (389), `login` (513), `mobileip-agent` (434), `mobilip-mn` (435), `msdp` (639), `netbios-dgm` (138), `netbios-ns` (137), `netbios-ssn` (139), `nfsd` (2049), `nntp` (119), `ntalk` (518), `ntp` (123), `pop3` (110), `pptp` (1723), `printer` (515), `radacct` (1813), `radius` (1812), `rip` (520), `rkinit` (2108), `smtp` (25), `snmp` (161), `snmptrap` (162), `snpp` (444), `socks` (1080), `ssh` (22), `sunrpc` (111), `syslog` (514), `tacacs-ds` (65), `talk` (517), `telnet` (23), `tftp` (69), `timed` (525), `who` (513), or `xdmcp` (177).
`forwarding-class` *`class`*	Match the forwarding class of the packet. Specify `assured-forwarding`, `best-effort`, `expedited-forwarding`, or `network-control`.

Table 15: Simple Filter Match Conditions *(Continued)*

Match Condition	Description
`protocol` *`number`*	IP protocol field. In place of the numeric value, you can specify one of the following text aliases (the field values are also listed): `ah` (51), `dstopts` (60), `egp` (8), `esp` (50), `fragment` (44), `gre` (47), `hop-by-hop` (0), `icmp` (1), `icmp6` (58), `icmpv6` (58), `igmp` (2), `ipip` (4), `ipv6` (41), `ospf` (89), `pim` (103), `rsvp` (46), `sctp` (132), `tcp` (6), `udp` (17), or `vrrp` (112).
`source-address` *`ip-source-address`*	Match the IP source address.
`source-port` *`number`*	Match the UDP or TCP source port field. If you configure this match condition, we recommend that you also configure the `protocol` match statement to determine which protocol is being used on the port. In place of the numeric field, you can specify one of the text aliases listed for `destination-port`.

Simple Filter Terminating Actions

Simple filters do *not* support explicitly configurable terminating actions, such as `accept`, `reject`, and `discard`. Terms configured in a simple filter always accept packets.

Simple filters do *not* support the `next` action.

Simple Filter Nonterminating Actions

Simple filters support only the following nonterminating actions:

- `forwarding-class (`*`forwarding-class`*` | assured-forwarding |best-effort | expedited-forwarding | network-control)`
- `loss-priority (high | low | medium-high | medium-low)`

Simple filters do not support actions that perform other functions on a packet (such as incrementing a counter, logging information about the packet header, sampling the packet data, or sending information to a remote host using the system log functionality).

RELATED DOCUMENTATION

Simple Filters and Policers Overview | 39

Example: Configure and Apply a Firewall Filter for a Multifield Classifier

IN THIS SECTION

- Requirements | 68
- Overview | 68
- Configuration | 70
- Verification | 74

This example shows how to configure a firewall filter to classify traffic using a multifield classifier. The classifier detects packets of interest to CoS as the packets arrive on an interface. Use multifield classifiers when a simple BA classifier is insufficient to classify a packet, when peering routers do not have CoS bits marked, or when the peering router's marking is untrusted.

Requirements

To verify this procedure, this example uses a traffic generator. The traffic generator can be hardware-based, or it can be software running on a server or host machine.

The functionality in this procedure is widely supported on devices that run Junos. The example shown here was tested and verified on MX Series routers running Junos OS Release 10.4.

Overview

IN THIS SECTION

- Topology | 69

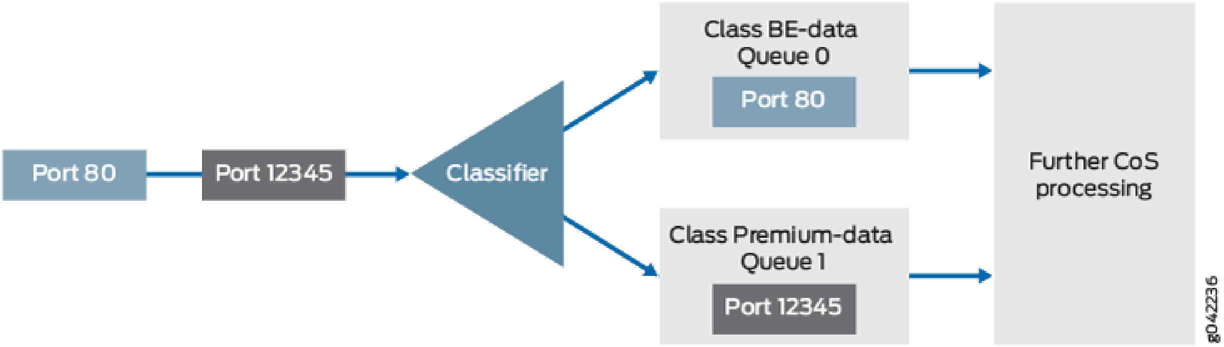
A classifier is a software operation that inspects a packet as it enters the router or switch. The packet header contents are examined, and this examination determines how the packet is treated when the network becomes too busy to handle all of the packets and you want your devices to drop packets intelligently, instead of dropping packets indiscriminately. One common way to detect packets of interest is by source port number. The TCP port numbers 80 and 12345 are used in this example, but many other matching criteria for packet detection are available to multifield classifiers, using firewall filter match conditions. The configuration in this example specifies that TCP packets with source port 80 are classified into the BE-data forwarding class and queue number 0. TCP packets with source port 12345 are classified into the Premium-data forwarding class and queue number 1.

You typically use multifield classifiers at the network edge as packets enter an autonomous system (AS).

In this example, you configure the firewall filter mf-classifier and specify some custom forwarding classes on Device R1. In specifying the custom forwarding classes, you also associate each class with a queue.

The classifier operation is shown in Figure 3 on page 69.

Figure 3: Multifield Classifier Based on TCP Source Ports

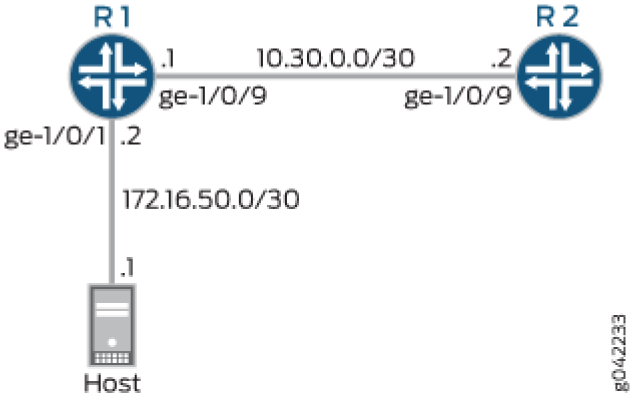


You apply the multifield classifier's firewall filter as an input filter on each customer-facing or host-facing interface that needs the filter. The incoming interface is ge-1/0/1 on Device R1. The classification and queue assignment is verified on the outgoing interface. The outgoing interface is Device R1's ge-1/0/9 interface.

Topology

Figure 4 on page 70 shows the sample network.

Figure 4: Multifield Classifier Scenario



"CLI Quick Configuration" on page 70 shows the configuration for all of the Juniper Networks devices in Figure 4 on page 70.

"Step-by-Step Procedure" on page 71 describes the steps on Device R1.

Classifiers are described in more detail in the following Juniper Networks Learning Byte video.

Video: Class of Service Basics, Part 2: Classification Learning Byte

Configuration

IN THIS SECTION

- Procedure | 70

Procedure

CLI Quick Configuration

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

Device R1

```
set interfaces ge-1/0/1 description to-host
set interfaces ge-1/0/1 unit 0 family inet filter input mf-classifier
set interfaces ge-1/0/1 unit 0 family inet address 172.16.50.2/30
set interfaces ge-1/0/9 description to-R2
set interfaces ge-1/0/9 unit 0 family inet address 10.30.0.1/30
set class-of-service forwarding-classes class BE-data queue-num 0
set class-of-service forwarding-classes class Premium-data queue-num 1
set class-of-service forwarding-classes class Voice queue-num 2
set class-of-service forwarding-classes class NC queue-num 3
set firewall family inet filter mf-classifier term BE-data from protocol tcp
set firewall family inet filter mf-classifier term BE-data from port 80
set firewall family inet filter mf-classifier term BE-data then forwarding-class BE-data
set firewall family inet filter mf-classifier term Premium-data from protocol tcp
set firewall family inet filter mf-classifier term Premium-data from port 12345
set firewall family inet filter mf-classifier term Premium-data then forwarding-class Premium-
data
set firewall family inet filter mf-classifier term accept-all-else then accept
```

Device R2

```
set interfaces ge-1/0/9 description to-R1
set interfaces ge-1/0/9 unit 0 family inet address 10.30.0.2/30
```

Step-by-Step Procedure

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

To configure Device R1:

1. Configure the device interfaces.

```
[edit interfaces]
user@R1# set ge-1/0/1 description to-host
user@R1# set ge-1/0/1 unit 0 family inet address 172.16.50.2/30
user@R1# set ge-1/0/9 description to-R2
user@R1# set ge-1/0/9 unit 0 family inet address 10.30.0.1/30
```

2. Configure the custom forwarding classes and associated queue numbers.

```
[edit class-of-service forwarding-classes]
user@R1# set BE-data queue-num 0
user@R1# set Premium-data queue-num 1
user@R1# set Voice queue-num 2
user@R1# set NC queue-num 3
```

3. Configure the firewall filter term that places TCP traffic with a source port of 80 (HTTP traffic) into the BE-data forwarding class, associated with queue 0.

```
[edit firewall family inet filter mf-classifier]
user@R1# set term BE-data from protocol tcp
user@R1# set term BE-data from port 80
user@R1# set term BE-data then forwarding-class BE-data
```

4. Configure the firewall filter term that places TCP traffic with a source port of 12345 into the Premium-data forwarding class, associated with queue 1.

```
[edit firewall family inet filter mf-classifier]
user@R1# set term Premium-data from protocol tcp
user@R1# set term Premium-data from port 12345
user@R1# set term Premium-data then forwarding-class Premium-data
```

5. At the end of your firewall filter, configure a default term that accepts all other traffic.

   Otherwise, all traffic that arrives on the interface and is not explicitly accepted by the firewall filter is discarded.

```
[edit firewall family inet filter mf-classifier]
user@R1# set term accept-all-else then accept
```

6. Apply the firewall filter to the ge-1/0/1 interface as an input filter.

```
[edit interfaces]
user@R1# set ge-1/0/1 unit 0 family inet filter input mf-classifier
```

Results

From configuration mode, confirm your configuration by entering the `show interfaces`, `show class-of-service`, `show firewall` commands. If the output does not display the intended configuration, repeat the instructions in this example to correct the configuration.

```
user@R1# show interfaces
ge-1/0/1 {
    description to-host;
    unit 0 {
        family inet {
            filter {
                input mf-classifier;
            }
            address 172.16.50.2/30;
        }
    }
}
ge-1/0/9 {
    description to-R2;
    unit 0 {
        family inet {
            address 10.30.0.1/30;
        }
    }
}
```

```
user@R1# show class-of-service
forwarding-classes {
    class BE-data queue-num 0;
    class Premium-data queue-num 1;
    class Voice queue-num 2;
    class NC queue-num 3;
}
```

```
user@R1# show firewall
family inet {
    filter mf-classifier {
        term BE-data {
            from {
                protocol tcp;
                port 80;
            }
            then forwarding-class BE-data;
        }
        term Premium-data {
            from {
                protocol tcp;
                port 12345;
            }
            then forwarding-class Premium-data;
        }
        term accept-all-else {
            then accept;
        }
    }
}
```

If you are done configuring the device, enter `commit` from configuration mode.

Verification

IN THIS SECTION

- Checking the CoS Settings | **74**
- Sending TCP Traffic into the Network and Monitoring the Queue Placement | **75**

Confirm that the configuration is working properly.

Checking the CoS Settings

Purpose

Confirm that the forwarding classes are configured correctly.

Action

From Device R1, run the `show class-of-service forwardng-classes` command.

```
user@R1> show class-of-service forwarding-class

Forwarding class                      ID      Queue  Restricted queue  Fabric priority
Policing priority   SPU priority
  BE-data                              0       0          0              low
normal            low
  Premium-data                         1       1          1              low
normal            low
  Voice                                2       2          2              low
normal            low
  NC                                   3       3          3              low
normal            low
```

Meaning

The output shows the configured custom classifier settings.

Sending TCP Traffic into the Network and Monitoring the Queue Placement

Purpose

Make sure that the traffic of interest is sent out the expected queue.

Action

1. Clear the interface statistics on Device R1’s outgoing interface.

   ```
   user@R1> clear interfaces statistics ge-1/0/9
   ```

2. Use a traffic generator to send 50 TCP port 80 packets to Device R2 or to some other downstream device.
3. On Device R1, check the queue counters.

Notice that you check the queue counters on the downstream output interface, not on the incoming interface.

```
user@R1> show interfaces extensive ge-1/0/9 | find "Queue counters"

  Queue counters:       Queued packets  Transmitted packets      Dropped packets
    0                               50                   50                    0
    1                                0                   57                    0
    2                                0                    0                    0
    3                                0                    0                    0
```

4. Use a traffic generator to send 50 TCP port 12345 packets to Device R2 or to some other downstream device.

```
[root@host]#  hping 172.16.60.1  -c 50 -s 12345  -k
```

5. On Device R1, check the queue counters.

```
user@R1> show interfaces extensive ge-1/0/9 | find "Queue counters"

  Queue counters:       Queued packets  Transmitted packets      Dropped packets
    0                               50                   50                    0
    1                               50                   57                    0
    2                                0                    0                    0
    3                                0                    0                    0
```

Meaning

The output shows that the packets are classified correctly. When port 80 is used in the TCP packets, queue 0 is incremented. When port 12345 is used, queue 1 is incremented.

RELATED DOCUMENTATION

Example: Configuring a Two-Rate Three-Color Policer

CHAPTER 4

Controlling Output Queues with Forwarding Classes

IN THIS CHAPTER

- Forwarding Classes Overview | 77
- Example: Configuring Forwarding Classes | 80
- Example: Assigning Forwarding Classes to Output Queues | 86
- Example: Classify all Traffic from a Remote Device with Fixed Interface-Based Classification | 89
- Understanding the SPC High-Priority Queue | 97
- Example: Configuring the SPC High-Priority Queue | 98
- Understanding Queuing and Marking of Host Outbound Traffic | 102
- Default Routing Engine Protocol Queue Assignments | 104

Forwarding Classes Overview

IN THIS SECTION

- Forwarding Class Queue Assignments | 78
- Forwarding Policy Options | 80

Forwarding classes (FCs) allow you to group packets for transmission and to assign packets to output queues. The forwarding class and the loss priority define the per-hop behavior (PHB in DiffServ) of a packet.

Juniper Networks devices support eight queues (0 through 7). For a classifier to assign an output queue (default queues 0 through 3) to each packet, it must associate the packet with one of the following forwarding classes:

Forwarding Class	Description
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 three drop probabilities (low, medium, and high).
Best effort (BE)	Provides no service profile. For the BE forwarding class, loss priority is typically not carried in a class-of-service (CoS) value, and random early detection (RED) drop profiles are more aggressive.
Network Control (NC)	This class is typically high priority because it supports protocol control.

In addition to behavior aggregate (BA) and multifield (MF) classification, the forwarding class (FC) of a packet can be directly determined by the *logical interface* that receives the packet. The packet FC can be configured using CLI commands, and if configured, this FC overrides the FC from any BA classification that was previously configured on the logical interface.

The following CLI command can assign an FC directly to packets received at a logical interface:

```
[edit class-of-service interfaces interface-name unit logical-unit-number]
forwarding-class class-name;
```

This section contains the following topics:

Forwarding Class Queue Assignments

Juniper Networks devices have eight queues built into the hardware. By default, four queues are assigned to four FCs. Table 16 on page 79 shows the four default FCs and queues that Juniper Networks classifiers assign to packets, based on the class-of-service (CoS) values in the arriving packet headers.

Queues 4 through 7 have no default assignments to FCs and are not mapped. To use queues 4 through 7, you must create custom FC names and map them to the queues.

By default, all incoming packets, except the IP control packets, are assigned to the FC associated with queue 0. All IP control packets are assigned to the FC associated with queue 3.

Table 16: Default Forwarding Class Queue Assignments

Forwarding Queue	Forwarding Class	Forwarding Class Description
Queue 0	best-effort (BE)	The Juniper Networks device does not apply any special CoS handling to packets with 000000 in the DiffServ field, a backward compatibility feature. These packets are usually dropped under congested network conditions.
Queue 1	expedited-forwarding (EF)	The Juniper Networks device delivers assured bandwidth, low loss, low delay, and low delay variation (jitter) end-to-end for packets in this service class. Devices accept excess traffic in this class, but in contrast to assured forwarding, out-of-profile expedited-forwarding packets can be forwarded out of sequence or dropped.
Queue 2	assured-forwarding (AF)	The Juniper Networks device 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 device accepts excess traffic, but applies a random early detection (RED) drop profile to determine whether the excess packets are dropped and not forwarded. Three drop probabilities (low, medium, and high) are defined for this service class.
Queue 3	network-control (NC)	The Juniper Networks device delivers packets in this service class with a low priority. (These packets are not delay sensitive.) Typically, these packets represent routing protocol hello or keepalive messages. Because loss of these packets jeopardizes proper network operation, delay is preferable to discard.

Forwarding Policy Options

CoS-based forwarding (CBF) enables you to control next-hop selection based on a packet’s CoS and, in particular, the value of the IP packet's precedence bits. For example, you can specify a particular interface or next hop to carry high-priority traffic while all the best-effort traffic takes some other path. CBF allows path selection based on FC. When a routing protocol discovers equal-cost paths, it can either pick a path at random or load-balance the packets across the paths through either hash selection or round-robin selection.

A forwarding policy also allows you to create CoS classification overrides. You can override the incoming CoS classification and assign the packets to an FC based on the packets’ input interfaces, input precedence bits, or destination addresses. When you override the classification of incoming packets, any mappings you configured for associated precedence bits or incoming interfaces to output transmission queues are ignored.

Example: Configuring Forwarding Classes

By default on all platforms, four output queues are mapped to four FCs as shown in "Forwarding Classes Overview" on page 77. On Juniper Networks devices, you can configure up to eight FCs and eight queues.

To configure up to eight FCs, include the **queue** statement at the **[edit class-of-service forwarding-classes]** hierarchy level:

```
[edit class-of-service forwarding-classes]
queue queue-number class-name;
```

The output queue number can be from 0 through 7, and you must map the forwarding classes one-to-one with the output queues. The default scheduler transmission rate and buffer size percentages for queues 0 through 7 are 95, 0, 0, 5, 0, 0, 0, and 0 percent, respectively.

For example, to configure a one-to-one mapping between eight FCs and eight queues, you would use the following configuration:

```
[edit class-of-service]
forwarding-classes {
    queue 0 be;
    queue 1 ef;
    queue 2 af;
    queue 3 nc;
    queue 4 ef1;
    queue 5 ef2;
    queue 6 af1;
    queue 7 nc1;
}
```

Defining Eight Classifiers

```
[edit class-of-service]
classifiers {
    dscp dscp-table {
        forwarding-class ef {
            loss-priority low code-points [101000, 101001];
            loss-priority high code-points [101010, 101011];
        }
        forwarding-class af {
            loss-priority low code-points [010000, 010001];
            loss-priority high code-points [010010, 010011];
        }
        forwarding-class be {
            loss-priority low code-points [000000];
        }
        forwarding-class nc {
            loss-priority low code-points [111000];
        }
        forwarding-class ef1 {
            loss-priority low code-points [101100, 101101];
            loss-priority high code-points [101110];
        }
        forwarding-class af1 {
            loss-priority high code-points [101110];
        }
        forwarding-class ef2 {
            loss-priority low code-points [101111];
        }
        forwarding-class nc1 {
            loss-priority low code-points [111001];
        }
    }
}
```

Adding Eight Schedulers to a Scheduler Map

Configure a custom scheduler map that applies globally to all interfaces, except those that are restricted to four queues:

```
[edit class-of-service]
scheduler-maps {
    sched {
        forwarding-class be scheduler Q0;
        forwarding-class ef scheduler Q1;
        forwarding-class af scheduler Q2;
        forwarding-class nc scheduler Q3;
        forwarding-class ef1 scheduler Q4;
        forwarding-class ef2 scheduler Q5;
        forwarding-class af1 scheduler Q6;
        forwarding-class nc1 scheduler Q7;
    }
}
schedulers {
    Q0 {
        transmit-rate percent 25;
        buffer-size percent 25;
        priority low;
        drop-profile-map loss-priority any protocol both drop-default;
    }
    Q1 {
        buffer-size temporal 2000;
        priority strict-high;
        drop-profile-map loss-priority any protocol both drop-ef;
    }
    Q2 {
        transmit-rate percent 35;
        buffer-size percent 35;
        priority low;
        drop-profile-map loss-priority any protocol both drop-default;
    }
    Q3 {
        transmit-rate percent 5;
        buffer-size percent 5;
        drop-profile-map loss-priority any protocol both drop-default;
    }
    Q4 {
        transmit-rate percent 5;
        priority high;
        drop-profile-map loss-priority any protocol both drop-ef;
    }
    Q5 {
        transmit-rate percent 10;
        priority high;
        drop-profile-map loss-priority any protocol both drop-ef;
    }
    Q6 {
        transmit-rate remainder;
        priority low;
        drop-profile-map loss-priority any protocol both drop-default;
    }
    Q7 {
        transmit-rate percent 5;
        priority high;
        drop-profile-map loss-priority any protocol both drop-default;
    }
}
```

Configuring an IP Precedence Classifier and Rewrite Tables

```
[edit class-of-service]
classifiers {
    inet-precedence inet-classifier {
        forwarding-class be {
            loss-priority low code-points 000;
        }
        forwarding-class af11 {
            loss-priority high code-points 001;
        }
        forwarding-class ef {
            loss-priority low code-points 010;
        }
        forwarding-class nc1 {
            loss-priority high code-points 011;
        }
        forwarding-class be1 {
            loss-priority low code-points 100;
        }
        forwarding-class af12 {
            loss-priority high code-points 101;
        }
        forwarding-class ef1 {
            loss-priority low code-points 110;
        }
        forwarding-class nc2 {
            loss-priority high code-points 111;
        }
    }
}
exp exp-rw-table {
    forwarding-class be {
        loss-priority low code-point 000;
    }
    forwarding-class af11 {
        loss-priority high code-point 001;
    }
    forwarding-class ef {
        loss-priority low code-point 010;
    }
    forwarding-class nc1 {
        loss-priority high code-point 111;
    }
    forwarding-class be1 {
        loss-priority low code-point 100;
    }
    forwarding-class af12 {
        loss-priority high code-point 101;
    }
    forwarding-class ef1 {
        loss-priority low code-point 110;
    }
    forwarding-class nc2 {
        loss-priority low code-point 111;
    }
}
inet-precedence inet-rw-table {
    forwarding-class be {
        loss-priority low code-point 000;
    }
    forwarding-class af11 {
        loss-priority high code-point 001;
    }
    forwarding-class ef {
        loss-priority low code-point 010;
    }
    forwarding-class nc1 {
        loss-priority low code-point 111;
    }
    forwarding-class be1 {
        loss-priority low code-point 100;
    }
    forwarding-class af12 {
        loss-priority high code-point 101;
    }
    forwarding-class ef1 {
        loss-priority low code-point 110;
    }
    forwarding-class nc2 {
        loss-priority low code-point 111;
    }
}
```

Configuring an IDP Policy with a Forwarding Class

Configure an IDP policy with a forwarding class as an action to rewrite DSCP values of IP packets:

```
[edit class-of-service]
security idp idp-policy policy_name rulebase-ips rule rule_name {
    then {
        action {
            class-of-service {
                    forwarding-class forwarding-class-name;
                    dscp-code-point value;
                }
            }
        }
    }
```

RELATED DOCUMENTATION

- Forwarding Classes Overview | 77
- Example: Assigning Forwarding Classes to Output Queues | 86
- *Example: Classifying All Traffic from a Remote Device by Configuring Fixed Interface-Based Classification*

Example: Assigning Forwarding Classes to Output Queues

IN THIS SECTION

- Requirements | 86
- Overview | 86
- Configuration | 87
- Verification | 89

This example shows how to assign forwarding classes to output queues.

Requirements

Before you begin, determine the MF classifier. See "Example: Configure and Apply a Firewall Filter for a Multifield Classifier" on page 68.

Overview

In this example, you configure class of service and assign best-effort traffic to queue 0 as be-class, expedited forwarding traffic to queue 1 as ef-class, assured forwarding traffic to queue 2 as af-class, and network control traffic to queue 3 as nc-class.

You must assign the forwarding classes established by the MF classifier to output queues. Table 17 on page 86 shows how this example assigns output queues.

Table 17: Sample Output Queue Assignments for mf-classifier Forwarding Queues

mf-classifier Forwarding Class	For Traffic Type	Output Queue
be-class	Best-effort traffic	Queue 0
ef-class	Expedited forwarding traffic	Queue 1
af-class	Assured forwarding traffic	Queue 2
nc-class	Network control traffic	Queue 3

Configuration

IN THIS SECTION

- Procedure | 87

Procedure

CLI Quick Configuration

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

```
set class-of-service forwarding-classes queue 0 be-class
set class-of-service forwarding-classes queue 1 ef-class
set class-of-service forwarding-classes queue  2 af-class
set class-of-service forwarding-classes queue 3 nc-class
```

Step-by-Step Procedure

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*.

To assign forwarding classes to output queues:

1. Configure class of service.

```
[edit]
user@host# edit class-of-service forwarding-classes
```

2. Assign best-effort traffic to queue 0.

```
[edit class-of-service forwarding-classes]
user@host# set queue 0 be-class
```

3. Assign expedited forwarding traffic to queue 1.

```
[edit class-of-service forwarding-classes]
user@host# set queue 1 ef-class
```

4. Assign assured forwarding traffic to queue 2.

```
[edit class-of-service forwarding-classes]
user@host# set queue 2 af-class
```

5. Assign network control traffic to queue 3.

```
[edit class-of-service forwarding-classes]
user@host# set queue 3 nc-class
```

Results

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

```
[edit]
user@host# show class-of-service
forwarding-classes {
    queue 0 be-class;
    queue 1 ef-class;
    queue 2 af-class;
    queue 3 nc-class;
}
```

If you are done configuring the device, enter `commit` from configuration mode.

NOTE: You cannot commit a configuration that assigns the same forwarding class to two different queues.

Verification

IN THIS SECTION

- Verifying Forwarding Classes Are Assigned to Output Queues | 89

Verifying Forwarding Classes Are Assigned to Output Queues

Purpose

Verify that the forwarding classes are properly assigned to output queues.

Action

From configuration mode, enter the `show class-of-service` command.

RELATED DOCUMENTATION

Forwarding Classes Overview | 77

Example: Classifying All Traffic from a Remote Device by Configuring Fixed Interface-Based Classification

Example: Configuring Forwarding Classes | 80

Example: Classify all Traffic from a Remote Device with Fixed Interface-Based Classification

IN THIS SECTION

- Requirements | 90
- Overview | 90
- Configuration | 92
- Verification | 95

This example shows the configuration of fixed classification based on the incoming interface. Fixed classification can be based on the physical interface or a logical interface.

Requirements

To verify this procedure, this example uses a traffic generator. The traffic generator can be hardware-based or it can be software running on a server or host machine.

The functionality in this procedure is widely supported on devices that run Junos. The example shown here was tested and verified on SRX Series Firewalls running a supported Junos OS release. The SRX Series Firewalls are configured to run as routers.

TIP: If you are performing tests on SRX Series Firewalls, you might need to configure the devices to run as unsecured routers in your test environment. You do not typically do this in a production environment.

Overview

IN THIS SECTION

- Topology | 91

A fixed interface classifier is the simplest way to classify all packets from a specific interface to a forwarding class. You typically use this approach on edge routers to classify all traffic from a remote device to a certain forwarding class and queue. A fixed interface classifier simply looks at the ingress interface on which the packet arrives and assigns all traffic received on that interface to a certain service class.

The fixed interface classifier cannot set the locally-meaningful packet-loss-priority, which is used by rewrite rules and drop profiles. The implicit packet-loss-priority is low for all fixed interface classifiers.

A fixed interface classifier is inadequate for scenarios in which interfaces receive traffic that belongs to multiple service classes. However, interface-based classification can be useful when it is combined with other classification processes. Filtering based on the inbound interface can improve the granularity of classification, for example, when combined with filtering based on code point markings. Combining the processes for interface and code point marking classification allows a single code point marking to have different meanings, depending on the interface on which the packet is received. If you want to combine a fixed interface classifier with a code point classifier, this is in effect a multifield classifier.

More Granular Alternative to Fixed Interface Classifier

In Junos, you can combine interface-based classification and code-point classification by using a multifield classifier, as follows:

```
[edit firewall family inet filter MF_CLASSIFIER term 1]
from {
    dscp ef;
    interface ge-0/0/0.0;
}
then forwarding-class Voice;
```

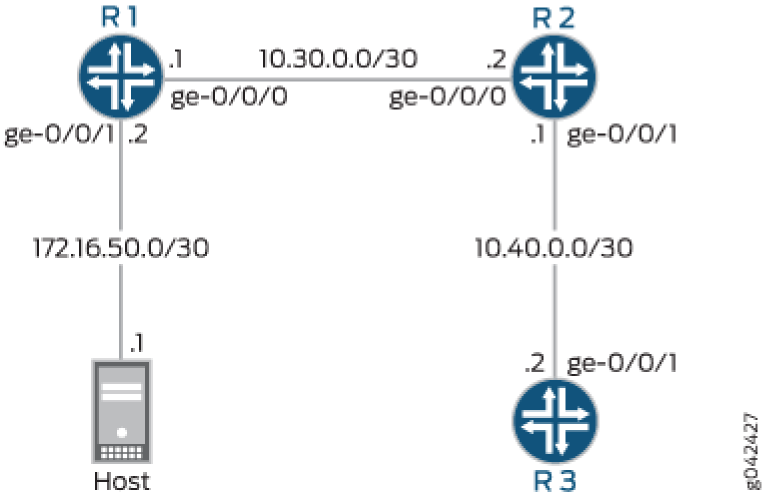
The following Juniper Networks Learning Byte video describes classifiers in more detail.

Video: Class of Service Basics, Part 2: Classification Learning Byte

Topology

Figure 5 on page 91 shows the sample network.

Figure 5: Fixed-Interface Classifier Scenario



To simulate voice traffic, this example shows TCP packets sent from the host to a downstream device. On Device R2, a fixed interface classifier routes the packets into the queue defined for voice traffic.

The classifier is assigned to interface ge-0/0/0 on Device R2. As always, verification of queue assignment is done on the egress interface, which is ge-0/0/1 on Device R2.

"CLI Quick Configuration" on page 92 shows the configuration for all of the Juniper Networks devices in Figure 5 on page 91. The section "Step-by-Step Procedure" on page 93 describes the steps on Device R2.

Configuration

IN THIS SECTION

- Procedure | 92

Procedure

CLI Quick Configuration

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

Device R1

```
set interfaces ge-0/0/0 description to-R2
set interfaces ge-0/0/0 unit 0 family inet address 10.30.0.1/30
set interfaces ge-0/0/1 description to-host
set interfaces ge-0/0/1 unit 0 family inet address 172.16.50.2/30
set interfaces lo0 unit 0 family inet address 192.168.0.1/32
set protocols ospf area 0.0.0.0 interface ge-0/0/0.0
set protocols ospf area 0.0.0.0 interface ge-0/0/1.0 passive
set protocols ospf area 0.0.0.0 interface lo0.0 passive
```

Device R2

```
set interfaces ge-0/0/0 unit 0 family inet address 10.30.0.2/30
set interfaces ge-0/0/1 unit 0 family inet address 10.40.0.1/30
set interfaces lo0 unit 0 family inet address 192.168.0.2/32
set protocols ospf area 0.0.0.0 interface ge-0/0/0.0
set protocols ospf area 0.0.0.0 interface ge-0/0/1.0
set protocols ospf area 0.0.0.0 interface lo0.0 passive
set class-of-service forwarding-classes queue 0 BE-data
set class-of-service forwarding-classes queue 1 Premium-data
set class-of-service forwarding-classes queue 2 Voice
set class-of-service forwarding-classes queue 3 NC
set class-of-service interfaces ge-0/0/0 unit 0 forwarding-class Voice
```

Device R3

```
set interfaces ge-0/0/0 unit 0 family inet address 10.50.0.1/30
set interfaces ge-0/0/1 unit 0 family inet address 10.40.0.2/30
set interfaces lo0 unit 0 family inet address 192.168.0.3/32
set protocols ospf area 0.0.0.0 interface ge-0/0/0.0
set protocols ospf area 0.0.0.0 interface ge-0/0/1.0
set protocols ospf area 0.0.0.0 interface lo0.0 passive
```

Step-by-Step Procedure

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.

To enable the default DSCP BA classifier:

1. Configure the device interfaces.

   ```
   [edit interfaces]
   user@R2# set ge-0/0/0 unit 0 family inet address 10.30.0.2/30
   user@R2# set ge-0/0/1 unit 0 family inet address 10.40.0.1/30
   user@R2# set lo0 unit 0 family inet address 192.168.0.2/32
   ```

2. Configure an interior gateway protocol (IGP) or static routes.

   ```
   [edit protocols ospf area 0.0.0.0]
   user@R2# set interface ge-0/0/0.0
   user@R2# set interface ge-0/0/1.0
   user@R2# set interface lo0.0 passive
   ```

3. Configure a set of forwarding classes.

   ```
   [edit class-of-service forwarding-classes]
   user@R2# set queue 0 BE-data
   user@R2# set queue 1 Premium-data
   ```

```
user@R2# set queue 2 Voice
user@R2# set queue 3 NC
```

4. Map all traffic that arrives on ge-0/0/0.0 into the Voice queue.

```
[edit class-of-service interfaces ge-0/0/0 unit 0]
user@R2# set forwarding-class Voice
```

Results

From configuration mode, confirm your configuration by entering the `show interfaces` and `show class-of-service` commands. If the output does not display the intended configuration, repeat the instructions in this example to correct the configuration.

```
user@R2# show interfaces
ge-0/0/0 {
    unit 0 {
        family inet {
            address 10.30.0.2/30;
        }
    }
}
ge-0/0/1 {
    unit 0 {
        family inet {
            address 10.40.0.1/30;
        }
    }
}
lo0 {
    unit 0 {
        family inet {
            address 192.168.0.2/32;
        }
    }
}
```

```
user@R2# show protocols
ospf {
    area 0.0.0.0 {
        interface ge-0/0/0.0;
        interface ge-0/0/1.0;
        interface lo0.0 {
            passive;
        }
    }
}
```

```
user@R2# show class-of-service
forwarding-classes {
    queue 0 BE-data;
    queue 1 Premium-data;
    queue 2 Voice;
    queue 3 NC;
}
interfaces {
    ge-0/0/0 {
        unit 0 {
            forwarding-class Voice;
        }
    }
}
```

If you are done configuring the device, enter `commit` from configuration mode.

Verification

IN THIS SECTION

- Verifying a Fixed-Interface Classifier | **96**

Confirm that the configuration is working properly.

Verifying a Fixed-Interface Classifier

Purpose

Verify that the fixed interface classifier is enabled on the Device R2's ingress interface. Keep in mind that although the classifier operates on incoming packets, you view the resulting queue assignment on the outgoing (egress) interface.

Action

1. Clear the interface statistics on Device R2's egress interface.

```
user@R2> clear interface statistics ge-0/0/1
```

2. Using a packet generator, send TCP packets to a device that is downstream of Device R2.

   This example uses the packet generator hping.

```
root@host> sudo hping3 10.40.0.2 -c 25 -fast

HPING 10.40.0.2 (eth0 10.40.0.2): NO FLAGS are set, 40 headers + 0 data bytes
len=46 ip=10.40.0.2 ttl=62 id=8619 sport=0 flags=RA seq=0 win=0 rtt=1.9 ms
len=46 ip=10.40.0.2 ttl=62 id=8620 sport=0 flags=RA seq=1 win=0 rtt=2.8 ms
len=46 ip=10.40.0.2 ttl=62 id=8621 sport=0 flags=RA seq=2 win=0 rtt=1.9 ms
len=46 ip=10.40.0.2 ttl=62 id=8623 sport=0 flags=RA seq=3 win=0 rtt=1.8 ms
len=46 ip=10.40.0.2 ttl=62 id=8624 sport=0 flags=RA seq=4 win=0 rtt=7.1 ms
len=46 ip=10.40.0.2 ttl=62 id=8625 sport=0 flags=RA seq=5 win=0 rtt=1.8 ms
len=46 ip=10.40.0.2 ttl=62 id=8626 sport=0 flags=RA seq=6 win=0 rtt=1.8 ms
len=46 ip=10.40.0.2 ttl=62 id=8627 sport=0 flags=RA seq=7 win=0 rtt=1.9 ms
len=46 ip=10.40.0.2 ttl=62 id=8628 sport=0 flags=RA seq=8 win=0 rtt=2.0 ms
len=46 ip=10.40.0.2 ttl=62 id=8634 sport=0 flags=RA seq=9 win=0 rtt=7.4 ms
len=46 ip=10.40.0.2 ttl=62 id=8635 sport=0 flags=RA seq=10 win=0 rtt=1.8 ms
len=46 ip=10.40.0.2 ttl=62 id=8636 sport=0 flags=RA seq=11 win=0 rtt=2.0 ms
len=46 ip=10.40.0.2 ttl=62 id=8637 sport=0 flags=RA seq=12 win=0 rtt=7.8 ms
len=46 ip=10.40.0.2 ttl=62 id=8639 sport=0 flags=RA seq=13 win=0 rtt=7.0 ms
len=46 ip=10.40.0.2 ttl=62 id=8640 sport=0 flags=RA seq=14 win=0 rtt=1.8 ms
len=46 ip=10.40.0.2 ttl=62 id=8641 sport=0 flags=RA seq=15 win=0 rtt=7.2 ms
len=46 ip=10.40.0.2 ttl=62 id=8642 sport=0 flags=RA seq=16 win=0 rtt=2.1 ms
len=46 ip=10.40.0.2 ttl=62 id=8643 sport=0 flags=RA seq=17 win=0 rtt=2.0 ms
len=46 ip=10.40.0.2 ttl=62 id=8644 sport=0 flags=RA seq=18 win=0 rtt=7.3 ms
len=46 ip=10.40.0.2 ttl=62 id=8645 sport=0 flags=RA seq=19 win=0 rtt=1.7 ms
```

```
len=46 ip=10.40.0.2 ttl=62 id=8646 sport=0 flags=RA seq=20 win=0 rtt=7.1 ms
len=46 ip=10.40.0.2 ttl=62 id=8647 sport=0 flags=RA seq=21 win=0 rtt=2.0 ms
len=46 ip=10.40.0.2 ttl=62 id=8648 sport=0 flags=RA seq=22 win=0 rtt=1.7 ms
len=46 ip=10.40.0.2 ttl=62 id=8649 sport=0 flags=RA seq=23 win=0 rtt=1.8 ms
len=46 ip=10.40.0.2 ttl=62 id=8651 sport=0 flags=RA seq=24 win=0 rtt=1.8 ms
```

3. On Device R2, verify that the Voice queue is incrementing.

```
user@R2> show interfaces extensive ge-0/0/1 | find "queue counters"
  Queue counters:       Queued packets  Transmitted packets  Dropped packets
    0 BE-data                        0                    0                0
    1 Premium-data                   0                    0                0
    2 Voice                         25                   25                0
    3 NC                             3                    3                0
  Queue number:         Mapped forwarding classes
    0                   BE-data
    1                   Premium-data
    2                   Voice
    3                   NC
...
```

Meaning

The output shows that the Voice queue has incremented by 25 packets after sending 25 packets through the ge-0/0/0 interface on Device R2.

RELATED DOCUMENTATION

Understanding How Behavior Aggregate Classifiers Prioritize Trusted Traffic

Default Aliases for CoS Value Bit Patterns Overview

Understanding the SPC High-Priority Queue

The Services Processing Card (SPC) provides processing power to run integrated services such as firewall, IPsec, and IDP. All traffic traversing the SRX Series Firewall is passed to an SPC to have service processing applied. Junos OS provides a configuration option to enable packets with specific

Differentiated Services (DiffServ) code points (DSCP) precedence bits to enter different priority queues on the SPC. The Services Processing Unit (SPU) draws packets from the higher priority queues and only draws packets from lower priority queues when the higher priority queues are empty. This feature can reduce overall latency for real-time traffic, such as voice traffic.

To designate packets for the high-priority, medium-priority, or low-priority queues, use the `spu-priority` configuration statement at the [`edit class-of-service forwarding-classes` *`class`*] hierarchy level. A value of `high` places packets into the high-priority queue, a value of `medium` places packets into the medium-priority queue, and a value of `low` places packets into the low-priority queue.

RELATED DOCUMENTATION

Example: Configuring the SPC High-Priority Queue | **98**

Forwarding Classes Overview | **77**

Example: Configuring the SPC High-Priority Queue

IN THIS SECTION

- Requirements | **98**
- Overview | **98**
- Configuration | **99**
- Verification | **101**

This example shows how to configure a forwarding class for the high-priority queue on the SPC.

Requirements

This example uses the following hardware and software components:

- SRX Series firewall
- Any supported Junos OS Release

Overview

This example defines the following forwarding classes and assigns a queue number to each class:

Table 18: Forwarding Class Definition

Forwarding Class	Queue Number
best-effort	0
assured-forwarding	1
network-control	3
expedited-forwarding	2

The expedited-forwarding class is configured for the high-priority queue on the SPC.

Configuration

IN THIS SECTION

- Procedure | 99

Procedure

CLI Quick Configuration

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

```
set class-of-service forwarding-classes class best-effort queue-num 0
set class-of-service forwarding-classes class assured-forwarding queue-num 1
set class-of-service forwarding-classes class network-control queue-num 3
set class-of-service forwarding-classes class expedited-forwarding queue-num 2 spu-priority high
```

Step-by-Step Procedure

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 Junos OS CLI User Guide.

To configure the high-priority queue on the SPC:

1. Define forwarding classes and assign queue numbers.

```
[edit class-of-service forwarding-classes]
user@host# set class best-effort queue-num 0
user@host# set class assured-forwarding queue-num 1
user@host# set class network-control queue-num 3
user@host# set class expedited-forwarding queue-num 2
```

2. Configure the SPC high-priority queue.

```
[edit class-of-service forwarding-classes]
user@host# set class expedited-forwarding spu-priority high
```

Results

From configuration mode, confirm your configuration by entering the `show class-of-service forwarding-classes` command. If the output does not display the intended configuration, repeat the configuration instructions in this example to correct it.

```
[edit]
user@host# show class-of-service forwarding-classes
class best-effort queue-num 0;
class assured-forwarding queue-num 1;
class network-control queue-num 3;
class expedited-forwarding queue-num 2 spu-priority high;
```

If you are done configuring the device, enter `commit` from configuration mode.

Verification

IN THIS SECTION

- Verifying SPU High-Priority Queue Mapping | 101

Verifying SPU High-Priority Queue Mapping

Purpose

Verify that the forwarding class is mapped to the SPU high-priority queue.

Action

From operational mode, enter the `show class-of-service forwarding-class` command.

```
user@host> show class-of-service forwarding-class
```

```
Forwarding class                       ID      Queue  Restricted queue  Fabric priority
Policing priority   SPU priority
  best-effort                           0       0          0               low
normal             low
  expedited-forwarding                  1       1          1               low
normal             high
  assured-forwarding                    2       2          2               low
normal             low
  network-control                       3       3          3               low
normal             low
```

RELATED DOCUMENTATION

Understanding the SPC High-Priority Queue | 97

Understanding Queuing and Marking of Host Outbound Traffic

IN THIS SECTION

- Host Outbound Traffic Overview | 102
- Default Queuing and Marking of Host Outbound Traffic | 103
- Configured Queuing and Marking of Host Outbound Traffic | 103
- Configured Queuing and Marking of Outbound Routing Engine Traffic Only | 103
- Platform-Specific Behavior | 103

This topic covers the following information:

Host Outbound Traffic Overview

Host outbound traffic, also called locally generated traffic, consists of traffic generated by the Routing Engine and traffic generated by the distributed protocol handler.

Routing Engine Sourced Traffic

Traffic sent from the Routing Engine includes control plane packets such as OSPF Hello packets, ICMP echo reply (ping) packets, and TCP-related packets such as BGP and LDP control packets.

Distributed Protocol Handler Traffic

Distributed protocol handler traffic refers to traffic from the router's *periodic packet management* (PPM) process when it runs sessions distributed to the Packet Forwarding Engine (the default mode) in addition to the Routing Engine. The PPM process is responsible for periodic transmission of protocol Hello or other keepalive packets on behalf of its various client processes, such as Bidirectional Forwarding Detection (BFD) Protocol or Link Aggregation Control Protocol (LACP), and it also receives packets on behalf of client processes. In addition, PPM handles time-sensitive periodic processing and performs such tasks as sending process-specific packets and gathering statistics. By default, PPM sessions on the Routing Engine run distributed on the Packet Forwarding Engine, and this enables client processes to run on the Packet Forwarding Engine.

Distributed protocol handler traffic includes both IP (Layer 3) traffic such as BFD keepalivemessages and non-IP (Layer 2) traffic such as LACP control traffic on aggregated Ethernet.

Default Queuing and Marking of Host Outbound Traffic

By default, the router assigns host outbound traffic to the `best-effort` forwarding class (which maps to queue 0) or to the `network-control` forwarding class (which maps to queue 3) based on protocol. For more information, see *Default Routing Engine Protocol Queue Assignments*.

For most protocols, the router marks the type of service (ToS) field of Layer 3 packets in the host outbound traffic flow with DiffServ code point (DSCP) bits 000000 (which correlate with the `best-effort` forwarding class). For some protocols, such as BGP, the router marks the ToS field with 802.1p bits 110 (which correlate with the `network-control` forwarding class), while still assigning the packets to queue 0. The router does not remark Layer 2 traffic such as LACP control traffic on aggregated Ethernet. For more information, see *Default DSCP and DSCP IPv6 Classifiers*.

Configured Queuing and Marking of Host Outbound Traffic

You can configure a nondefault forwarding class and DSCP bits that the router uses to queue and remark host outbound traffic. These configuration settings apply to the following types of traffic:

- Packets generated by the Routing Engine
- Distributed protocol handler traffic for egress interfaces

To change these default settings, include the `forwarding-class` *`class-name`* statement and the `dscp-code-point` *`value`* statement at the `[edit class-of-service host-outbound-traffic]` hierarchy level. This feature does not affect transit traffic or incoming traffic.

The configured forwarding class override applies to all packets relating to Layer 2 protocols, Layer 3 protocols, and all application-level traffic (such as FTP or ping operations). The configured DSCP bits override value does not apply to MPLS EXP bits or IEEE 802.1p bits, however.

Configured Queuing and Marking of Outbound Routing Engine Traffic Only

To configure a nondefault forwarding class and DSCP bits that the router uses to queue and remark traffic generated by the Routing Engine only, attach an IPv4 firewall filter to the output of the router's loopback address. Use the `forwarding-class` and `dscp` filter actions to specify override values.

This feature overrides the `host-outbound-traffic` settings for the Routing Engine output traffic only.

Platform-Specific Behavior

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

Table 19: Platform-Specific Behavior

Platform	Difference
MX80 Routers	- For interfaces on MX80 routers, LACP control traffic is sent through the Routing Engine rather than through the Packet Forwarding Engine.

RELATED DOCUMENTATION

Default Routing Engine Protocol Queue Assignments

Default DSCP and DSCP IPv6 Classifiers

Example: Configuring Different Queuing and Marking Defaults for Outbound Routing Engine and Distributed Protocol Handler Traffic

Default Routing Engine Protocol Queue Assignments

Table 20 on page 105 lists the default output queues to which Routing Engine sourced traffic is mapped by protocol type. In general, control protocol packets are sent over queue 3 and management traffic is sent over queue 0. The following caveats apply to these default queue assignments:

- For all packets sent to queue 3 over a VLAN-tagged interface, the software sets the 802.1p bit to 110, except for VRRP packets, in which case the software sets the 802.1p bit to 111.
- Outgoing BFD packets should be marked with VLAN-tagged 802.1p bit to 110; however, this is true only for RE based BFD. For inline BFD, it does not modify by default.
- Although BGP and LDP TCP traffic is sent to queue 0, it is marked with 802.1p bits 110 normally used for network control.
- For IPv4 and IPv6 packets, the software copies the IP type-of-service (ToS) value into the 802.1p field independently of which queue the packets are sent out.
- For MPLS packets, the software copies the EXP bits into the 802.1p field.

Table 20: Default Queue Assignments for Packets Generated by the Routing Engine

Routing Engine Protocol	Default Queue Assignment
Adaptive Services PIC TCP tickle (keepalive packets for idle session generated with stateful firewall to probe idle TCP sessions)	Queue 0
Address Resolution Protocol (ARP)	Queue 0
Bidirectional Forwarding Detection (BFD) Protocol	Queue 3
BGP	Queue 0
BGP TCP Retransmission	Queue 3
Cisco High-Level Data Link Control (HDLC)	Queue 3
Distance Vector Multicast Routing Protocol (DVMRP)	Queue 3
Ethernet Operation, Administration, and Maintenanc e (OAM)	Queue 3
FTP	Queue 0
IS-IS Open Systems Interconnection (OSI)	Queue 3
Internet Control Message Protocol (ICMP)	Queue 0
Internet Group Management Protocol (IGMP) query	Queue 3
IGMP Report	Queue 0
Internet Key Exchange (IKE)	Queue 3

Table 20: Default Queue Assignments for Packets Generated by the Routing Engine *(Continued)*

Routing Engine Protocol	Default Queue Assignment
IP version 6 (IPv6) Neighbor Solicitation	Queue 3
IPv6 Neighbor Advertisement	Queue 3
IPv6 Router Advertisement	Queue 0
Label Distribution Protocol (LDP) User Datagram Protocol (UDP) hello	Queue 3
LDP keepalive and Session data	Queue 0
LDP TCP Retransmission	Queue 3
Link Aggregation Control Protocol (LACP)	Queue 3
Link Services (LS) PIC	If link fragmentation and interleaving (LFI) is enabled, all routing protocol packets larger than 128 bytes are transmitted from queue 0. This ensures that VoIP traffic is not affected. Fragmentation is supported on queue 0 only.
Multicast listener discovery (MLD)	Queue 0
Multicast Source Discovery Protocol (MSDP)	Queue 0
MSDP TCP Retransmission	Queue 3
NETCONF	Queue 0
NetFlow	Queue 0
OSPF protocol data unit (PDU)	Queue 3

Table 20: Default Queue Assignments for Packets Generated by the Routing Engine *(Continued)*

Routing Engine Protocol	Default Queue Assignment
Point-to-Point Protocol (PPP)	Queue 3
Protocol Independent Multicast (PIM)	Queue 3
Precision Time Protocol (PTP)	Queue 3
Real-time performance monitoring (RPM) probe packets	Queue 3
RSVP	Queue 3
Routing Information Protocol (RIP)	Queue 3
SNMP	Queue 0
SSH	Queue 0
sFlow monitoring technology	Queue 0
Telnet	Queue 0
Two-Way Active Monitoring Protocol (TWAMP)	Queue 0
Virtual Router Redundancy Protocol (VRRP)	Queue 3
`xnm-clear-text`	Queue 0
`xnm-ssl`	Queue 0

CHAPTER 5

Altering Outgoing Packets Headers with Rewrite Rules

IN THIS CHAPTER

- Rewrite Rules Overview | **108**
- Rewriting Frame Relay Headers | **109**
- Example: Configure and Apply Rewrite Rules on a Security Device | **110**

Rewrite Rules Overview

A rewrite rule modifies the appropriate class-of-service (CoS) bits in an outgoing packet. Modification of CoS bits allows the next downstream device to classify the packet into the appropriate service group. Rewriting or marking outbound packets is useful when the device is at the border of a network and must alter the CoS values to meet the policies 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.

Typically, a device rewrites CoS values in outgoing packets on the outbound interfaces of an edge device, to meet the policies of the targeted peer. After reading the current forwarding class and loss priority information associated with the packet, the transmitting device locates the chosen CoS value from a table, and writes this CoS value into the packet header.

You can configure 802.1p rewrite on logical VDSL interface, that is, `pt` interface.

Change History Table

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

Release	Description
18.2R1	Starting in Junos OS Release 18.2R1, you can configure 802.1p rewrite on logical VDSL interface, that is, `pt` interface.

Rewriting Frame Relay Headers

IN THIS SECTION

- Assigning the Default Frame Relay Rewrite Rule to an Interface | 109
- Defining a Custom Frame Relay Rewrite Rule | 110

Assigning the Default Frame Relay Rewrite Rule to an Interface

For Juniper Networks device interfaces with Frame Relay encapsulation, you can rewrite the discard eligibility (DE) bit based on the loss priority of Frame Relay traffic. For each outgoing frame with the loss priority set to low, medium-low, medium-high, or high, you can set the DE bit CoS value to `0` or `1`. You can combine a Frame Relay rewrite rule with other rewrite rules on the same interface. For example, you can rewrite both the DE bit and MPLS EXP bit.

The default Frame Relay rewrite rule contains the following settings:

```
loss-priority low code-point 0;
loss-priority medium-low code-point 0;
loss-priority medium-high code-point 1;
loss-priority high code-point 1;
```

This default rule sets the DE CoS value to `0` for each outgoing frame with the loss priority set to low or medium-low. This default rule sets the DE CoS value to `1` for each outgoing frame with the loss priority set to medium-high or high.

To assign the default rule to an interface, include the `frame-relay-de default` statement at the `[edit class-of-service interfaces interface` *`interface-name`* `unit` *`logical-unit-number`* `unit rewrite-rules]` hierarchy level:

```
[edit class-of-service interfaces interface-name unit logical-unit-number rewrite-rules]
frame-relay-de default;
```

Defining a Custom Frame Relay Rewrite Rule

To define a custom Frame Relay rewrite rule, include the following statements at the `[edit class-of-service]` hierarchy level:

```
[edit class-of-service]
rewrite-rules {
    frame-relay-de rewrite-name {
         import (rewrite-name | default);
        forwarding-class class-name {
            loss-priority level code-point (0 | 1);
        }
    }
}
```

A custom rewrite rule sets the DE bit to the `0` or `1` CoS value based on the assigned loss priority of low, medium-low, medium-high, or high for each outgoing frame.

The rule does not take effect until you apply it to a logical interface. To apply the rule to a logical interface, include the `frame-relay-de` *`map-name`* statement at the `[edit class-of-service interfaces interface` *`interface-name`* `unit` *`logical-unit-number`* `rewrite-rules]` hierarchy level:

```
[edit class-of-service interfaces interface-name unit logical-unit-number rewrite-rules]
frame-relay-de map-name;
```

RELATED DOCUMENTATION

Rewrite Rules Overview | **108**

Example: Configure and Apply Rewrite Rules on a Security Device | **110**

Example: Configure and Apply Rewrite Rules on a Security Device

IN THIS SECTION

- Requirements | **111**

- Overview | 111
- Configuration | 112
- Verification | 115

This example shows how to configure and apply rewrite rules for a device.

Requirements

Before you begin, create and configure the forwarding classes.

Overview

You can configure rewrite rules to replace CoS values on packets received from the customer or host with the values expected by other devices. You do not have to configure rewrite rules if the received packets already contain valid CoS values. Rewrite rules apply the forwarding class information and packet loss priority used internally by the device to establish the CoS value on outbound packets. After you configure rewrite rules, you must apply them to the correct interfaces.

In this example, you configure the rewrite rule for DiffServ CoS as rewrite-dscps. You specify the best-effort forwarding class as be-class, expedited forwarding class as ef-class, an assured forwarding class as af-class, and a network control class as nc-class. Finally, you apply the rewrite rule to an IRB interface.

You can apply one rewrite rule to each logical interface.

Table 21 on page 111 shows how the rewrite rules replace the DSCPs on packets in the four forwarding classes.

Table 21: Sample rewrite-dscps Rewrite Rules to Replace DSCPs

mf-classifier Forwarding Class	For CoS Traffic Type	rewrite-dscps Rewrite Rules
be-class	Best-effort traffic—Provides no special CoS handling of packets. Typically, RED drop profile is aggressive and no loss priority is defined.	Low-priority code point: 000000 High-priority code point: 000001
ef-class	Expedited forwarding traffic—Provides low loss, low delay, low jitter, assured bandwidth, and end-to-end service. Packets can be forwarded out of sequence or dropped.	Low-priority code point: 101110 High-priority code point: 101111

Table 21: Sample rewrite-dscps Rewrite Rules to Replace DSCPs *(Continued)*

mf-classifier Forwarding Class	For CoS Traffic Type	rewrite-dscps Rewrite Rules
af-class	Assured forwarding traffic—Provides high assurance for packets within the specified service profile. Excess packets are dropped.	Low-priority code point: 001010 High-priority code point: 001100
nc-class	Network control traffic—Packets can be delayed, but not dropped.	Low-priority code point: 110000 High-priority code point: 110001

Forwarding classes can be configured in a DSCP rewriter and also as an action of an IDP policy to rewrite DSCP code points. To ensure that the forwarding class is used as an action in an IDP policy, it is important that you do not configure an IDP policy and interface-based rewrite rules with the same forwarding class.

Configuration

IN THIS SECTION

- Procedure | 112

Procedure

CLI Quick Configuration

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

```
set class-of-service  rewrite-rules dscp rewrite-dscps  forwarding-class be-class loss-priority
low code-point 000000
set class-of-service  rewrite-rules dscp rewrite-dscps  forwarding-class be-class loss-priority
high code-point 000001
set class-of-service  rewrite-rules dscp rewrite-dscps  forwarding-class ef-class loss-priority
low code-point 101110
```

```
set class-of-service  rewrite-rules dscp rewrite-dscps  forwarding-class ef-class loss-priority 
high code-point 101111
set class-of-service  rewrite-rules dscp rewrite-dscps  forwarding-class af-class loss-priority 
low code-point 001010
set class-of-service  rewrite-rules dscp rewrite-dscps  forwarding-class af-class loss-priority 
high code-point 001100
set class-of-service  rewrite-rules dscp rewrite-dscps  forwarding-class nc-class loss-priority 
low code-point 110000
set class-of-service  rewrite-rules dscp rewrite-dscps  forwarding-class nc-class loss-priority 
high code-point 110001
set class-of-service  interfaces irb unit 0 rewrite-rules dscp rewrite-dscps
```

Step-by-Step Procedure

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.

To configure and apply rewrite rules for a device:

1. Configure rewrite rules for DiffServ CoS.

```
[edit]
user@host# edit class-of-service
user@host# edit rewrite-rules dscp rewrite-dscps
```

2. Configure best-effort forwarding class rewrite rules.

```
[edit class-of-service rewrite-rules dscp rewrite-dscps]
user@host# set forwarding-class be-class loss-priority low code-point 000000
user@host# set forwarding-class be-class loss-priority high code-point 000001
```

3. Configure expedited forwarding class rewrite rules.

```
[edit class-of-service rewrite-rules dscp rewrite-dscps]
user@host# set forwarding-class ef-class loss-priority low code-point 101110
user@host# set forwarding-class ef-class loss-priority high code-point 101111
```

4. Configure assured forwarding class rewrite rules.

```
[edit class-of-service rewrite-rules dscp rewrite-dscps]
user@host# set forwarding-class af-class loss-priority low code-point 001010
user@host# set forwarding-class af-class loss-priority high code-point 001100
```

5. Configure network control class rewrite rules.

```
[edit class-of-service rewrite-rules dscp rewrite-dscps]
user@host# set forwarding-class nc-class loss-priority low code-point 110000
user@host# set forwarding-class nc-class loss-priority high code-point 110001
```

6. Apply rewrite rules to an IRB interface.

```
[edit class-of-service]
user@host# set interfaces irb unit 0 rewrite-rules dscp rewrite-dscps
```

Results

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

```
[edit]
    user@host# show class-of-service
    interfaces {
        irb {
            unit 0 {
            rewrite-rules {
            dscp rewrite-dscps;
               }
            }
        }
    }
    rewrite-rules {
        dscp rewrite-dscps {
        forwarding-class be-class {
        loss-priority low code-point 000000;
            loss-priority high code-point 000001;
            }
            forwarding-class ef-class {
                loss-priority low code-point 101110;
            loss-priority high code-point 101111;
        }
        forwarding-class af-class {
            loss-priority low code-point 001010;
        loss-priority high code-point 001100;
    }
    forwarding-class nc-class {
        loss-priority low code-point 110000;
        loss-priority high code-point 110001;
        }
    }
}
```

If you are done configuring the device, enter `commit` from configuration mode.

Verification

IN THIS SECTION

- Verify Rewrite Rules Configuration | **115**

Verify Rewrite Rules Configuration

Purpose

Verify that rewrite rules are configured properly.

Action

From operational mode, enter the `show class-of-service interface irb` command.

```
user@host> show class-of-service interface irb
Physical interface: irb, Index: 130
 Maximum usable queues: 8, Queues in use: 4
 Scheduler map: <default> , Index: 2
Congestion-notification: Disabled
```

```
Logical interface: irb.10, Index: 71
Object                  Name                  Type                    Index
Rewrite-Output          rewrite-dscps         dscp                    17599
Classifier              ipprec-compatibility  ip                         13
```

Meaning

Rewrite rules are configured on IRB interface as expected.

CHAPTER 6

Defining Output Queue Properties with Schedulers

IN THIS CHAPTER

- Schedulers Overview | **117**
- Default Scheduler Settings | **123**
- Transmission Scheduling Overview | **124**
- Excess Bandwidth Sharing and Minimum Logical Interface Shaping | **126**
- Excess Bandwidth Sharing Proportional Rates | **127**
- Calculated Weights Mapped to Hardware Weights | **128**
- Weight Allocation with Only Shaping Rates or Unshaped Logical Interfaces | **129**
- Shared Bandwidth Among Logical Interfaces | **131**
- Example: Configuring Class-of-Service Schedulers on a Security Device | **133**
- Scheduler Buffer Size Overview | **139**
- Configuring Large Delay Buffers in CoS | **143**
- Example: Configuring and Applying Scheduler Maps | **149**
- Applying Scheduler Maps and Shaping Rate to Logical Interfaces and VLANs | **153**
- Example: Applying Scheduling and Shaping to VLANs | **158**

Schedulers Overview

IN THIS SECTION

- Transmit Rate | **118**
- Delay Buffer Size | **119**
- Scheduling Priority | **121**
- Shaping Rate | **121**
- Platform-Specific Schedulers Behavior | **122**

You use *schedulers* to define the 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 random early detection (RED) 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 hardware queues, packet schedulers, and RED processes that operate according to this mapping.

An individual device interface has multiple queues assigned to store packets temporarily before transmission. To determine the order to service the queues, the device uses a round-robin scheduling method based on priority and the queue's weighted round-robin (WRR) credits. Junos OS schedulers allow you to define the priority, bandwidth, delay buffer size, rate control status, and RED drop profiles to be applied to a particular queue for packet transmission.

You can configure *per-unit scheduling* (*also called logical interface scheduling*) to allow multiple output queues on a logical interface and to associate an output scheduler with each queue.

NOTE: For Juniper Network devices, when configuring the *protocol parameter* in the **drop-profile-map** statement, TCP and non-TCP values are not supported; only the value *any* is supported.

Transmit Rate

The transmission rate determines the traffic transmission bandwidth for each forwarding class you configure. The rate is specified in bits per second (bps). Each queue is allocated some portion of the bandwidth of the outgoing interface.

This bandwidth amount 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. You can limit the transmission bandwidth to the exact value you configure, or allow it to exceed the configured rate if additional bandwidth is available from other queues. This property helps ensure that each queue receives the amount of bandwidth appropriate to its level of service.

The minimum transmit rate supported on high-speed interfaces is one-ten thousandth of the speed of that interface. For example, on a Gigabit Ethernet interface with a speed of 1000 Mbps, the minimum transmit rate is 100 Kbps (1,000 Mbps x 1/10,000). You can configure transmit rates in the range 3200 bps through 160,000,000,000 bps. When the configured rate is less than the minimum transmit rate, the minimum transmit rate is used instead.

Transmit rate assigns the weighted round-robin (WRR) priority values within a given priority level and not between priorities.

The 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 7 have the following percentage of transmission capacity:

- Queue 0—95 percent
- Queue 1—0 percent
- Queue 2—0 percent
- Queue 3—0 percent
- Queue 4—0 percent
- Queue 6—0 percent
- Queue 7—5 percent

To define a transmit rate, select the appropriate option:

- To specify a transmit rate, select `rate` and type an integer from 3200 to 160,000,000,000 bits per second.
- To enforce an exact transmit rate, select `rate`.
- To specify the remaining transmission capacity, select `remainder`.
- To specify a percentage of transmission capacity, select `percent` and type an integer from 1 through 100.

Optionally, you can specify the percentage of the remainder to be used for allocating the transmit rate of the scheduler on a prorated basis. If there are still points left even after allocating the remainder percentage with the transmit rate and there are no queues, then the points are allocated point by point to each queue in a round-robin method. If the remainder percentage is not specified, the remainder value will be shared equally.

Delay Buffer Size

You can configure the delay buffer size to control congestion at the output stage. A delay buffer provides packet buffer space to absorb burst traffic up to a specified duration of delay. When the buffer is full, all packets are dropped.

To avoid performance issues with large delay buffers the maximum interface bandwidth, as used to calculate the delay buffer, is capped at 100 Mbps. Interfaces that operate above this rate are scaled down to 100 Mbps for purposes of the delay buffer calculation. The delay buffer is calculated as:

Delay-buffer (in bits) = available interface bandwidth (<= 100Mbs) x configured buffer size percentage x maximum delay buffer time (.1 seconds).

As an example, consider a 10GE interface with a configured buffer size percentage of 50%. The interface rate is scaled down to 100 Mbps, yielding this result: 100M * 0.5 * 0.1 sec} = 5,000,000 bits. This value is divided by 8 to convert bits to Bytes. The result is a buffer depth of 625,000 Bytes (0.625MB).

To define a delay buffer size for a scheduler, select the appropriate option:

- To enforce exact buffer size, select `Exact`.
- To specify a buffer size as a temporal value (microseconds), select `Temporal`.
- 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`.

Optionally, you can specify the percentage of the remainder to be used for allocating the buffer size of the scheduler on a prorated basis.

By default, sizes of the delay buffer queues 0 through 7 have the following percentage of the total available buffer space:

- Queue 0—95 percent
- Queue 1—0 percent
- Queue 2—0 percent
- Queue 3—0 percent
- Queue 4—0 percent
- Queue 5—0 percent
- Queue 6—0 percent
- Queue 7—5 percent

NOTE: A large buffer size value correlates with a greater possibility of packet delays. This might not be practical for sensitive traffic such as voice or video.

Packets are dropped from the queue if:

- The total buffer limit is exceeded.
- The queue size exceeds the total free buffer size.
- The packet buffer pool is less then 25 percent free and the queue exceeds the guaranteed minimum buffer size.

- The packet buffer pool is only 5 percent free (or less).
- The queue size exceeds the guaranteed buffer size (RED profile condition (RED-dropped)). The queue size will be restricted to be less than or equal to the free shared buffers available.

Scheduling Priority

Scheduling priority determines the order in which an output interface transmits traffic from the queues, thus ensuring that queues containing important traffic are provided better access to the outgoing interface.

The queues for an interface are divided into sets based on their priority. Each set contains queues of the same priority. The device examines the sets in descending order of priority. If at least one queue in a set has a packet to transmit, the device selects that set. If multiple queues in the set have packets to transmit, the device selects a queue from the set according to the weighted round-robin (WRR) algorithm that operates within the set.

The packets in a queue are transmitted based on the configured scheduling priority, the transmit rate, and the available bandwidth.

The scheduling priority of the scheduler determines the order in which an output interface transmits traffic from the queues. You can set scheduling priority at different levels in an order of increasing priority from low to high. A high-priority queue with a high transmission rate might lock out lower-priority traffic.

To specify a scheduling priority, select one of the following levels:

- `high`—Packets in this queue have high priority.
- `low`—Packets in this queue are transmitted last.
- `medium-low`—Packets in this queue have medium-low priority.
- `medium-high`—Packets in this queue have medium-high priority.
- `strict-high`—Packets in this queue are transmitted first.

Shaping Rate

Shaping rates control the maximum rate of traffic transmitted on an interface. You can configure the shaping rate so that the interface transmits less traffic than it is physically capable of carrying.

You can configure shaping rates on logical interfaces. By default, output scheduling is not enabled on logical interfaces. *Logical interface* scheduling (also called per-unit scheduling) allows you to enable multiple output queues on a logical interface and associate an output scheduler and shaping rate with the queues.

By default, the logical interface bandwidth is the average of unused bandwidth for the number of logical interfaces that require default bandwidth treatment. You can specify a peak bandwidth rate in bits per second (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). The range is from 1000 through 32,000,000,000 bps.

For low-speed interfaces, the queue-limit values might become lower than the interface MTU so that traffic with large packets can no longer pass through some of the queues. If you want larger-sized packets to flow through, set the buffer-size configuration in the scheduler to a larger value. For more accuracy, the 100-ms queue-limit values are calculated based on shaping rate and not on interface rates.

The shaping rate defines the minimum bandwidth allocated to a queue. The default shaping rate is 100 percent, which is the same as no shaping at all. To define a shaping rate, select the appropriate option:

- To specify shaping rate as an absolute number of bits per second, select `rate` and type an integer from 3200 to 160,000,000,000 bits per second.
- To specify shaping rate as a percentage, select `percent` and type an integer from 0 through 100.

Platform-Specific Schedulers Behavior

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

Table 22: Platform-Specific Schedulers Behavior

Platform	Difference
SRX5400, SRX5600, and SRX5800	• SRX5400, SRX5600, and SRX5800 firewalls do not support an exact value transmit rate.
vSRX	• vSRX Virtual Firewall and vSRX Virtual Firewall 3.0 instances support class of service (CoS) configurations for shapers at different Gigabit Ethernet interface speeds of 1-Gbps, 10-Gbps, 40-Gbps, and 100-Gbps.

RELATED DOCUMENTATION

Default Scheduler Settings | 123

Example: Configuring Class-of-Service Schedulers on a Security Device | 133

Scheduler Buffer Size Overview | 139

Example: Configuring and Applying Scheduler Maps | **149**

Transmission Scheduling Overview | **124**

Default Scheduler Settings

Each forwarding class has an associated scheduler priority. Only two forwarding classes, best-effort (ID 0, queue 0) and network-control (ID 3, queue 7), are used in the Junos OS default scheduler configuration.

By default, the best-effort forwarding class (queue 0) receives 95 percent, and the network-control (queue 7) receives 5 percent of the bandwidth and buffer space for the output link. The default drop profile causes the buffer to fill and then discard all packets until it again has space.

The expedited-forwarding and assured-forwarding classes have no schedulers, because by default no resources are assigned to queue 5 (ID 1) and queue 1 (ID 2). However, you can manually configure resources for the expedited-forwarding and the assured-forwarding classes.

NOTE: The ID refers to the forwarding class ID assigned by the COSD daemon. COSD assigns a forwarding class ID to every forwarding class. The ID is unique to a forwarding-class and is used as a unique identifier in any internal communication with the PFE. PFE side software knows nothing about forwarding-class names but only IDs. So, there is one-to-one mapping from forwarding class name to ID.

By default, each 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 receive a larger amount of offered load than the bandwidth allocated. If you do not want a queue to use any leftover bandwidth, you must configure it for strict allocation.

The device uses the following default scheduler settings. You can configure these settings.

```
[edit class-of-service]
schedulers {
    network-control {
        transmit-rate percent 5;
        buffer-size percent 5;
        priority low;
        drop-profile-map loss-priority any protocol any drop-profile terminal;
    }
    best-effort {
        transmit-rate percent 95;
        buffer-size percent 95;
        priority low;
        drop-profile-map loss-priority any protocol any drop-profile terminal;
    }
}
drop-profiles {
    terminal {
        fill-level 100 drop-probability 100;
    }
}
```

RELATED DOCUMENTATION

- Schedulers Overview | **117**
- Example: Configuring Class-of-Service Schedulers on a Security Device | **133**
- Scheduler Buffer Size Overview | **139**
- Example: Configuring and Applying Scheduler Maps | **149**
- Transmission Scheduling Overview | **124**

Transmission Scheduling Overview

The packets in a queue are transmitted based on their transmission priority, transmit rate, and the available bandwidth.

By default, each 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 receive a larger amount of offered load than the bandwidth allocated. A queue receiving traffic within its bandwidth configuration is considered to have positive bandwidth credit, and a queue receiving traffic in excess of its bandwidth allocation is considered to have negative bandwidth credit.

NOTE: The queues in a logical interface do not use the available buffer from other queues for packet transmission. Instead, the packets transmitted to a queue consider only the buffer size available in its own queue.

A queue with positive credit does not need to use leftover bandwidth, because it can use its own allocation. For such queues, packets are transmitted based on the priority of the queue, with packets from higher-priority queues transmitting first. The transmit rate is not considered during transmission. In contrast, a queue with negative credit needs a share of the available leftover bandwidth.

The leftover bandwidth is allocated to queues with negative credit in proportion to the configured transmit rate of the queues within a given priority set. The queues for an interface are divided into sets based on their priority. If no transmit rate is configured, each queue in the set receives an equal percentage of the leftover bandwidth. However, if a transmit rate is configured, each queue in the set receives the configured percentage of the leftover bandwidth.

Table 23 on page 125 shows a sample configuration of priority and transmit rate on six queues. The total available bandwidth on the interface is 100 Mbps.

Table 23: Sample Transmission Scheduling

Queue	Scheduling Priority	Transmit Rate	Incoming Traffic
0	Low	10%	20 Mbps
1	High	20%	20 Mbps
2	High	30%	20 Mbps
3	Low	30%	20 Mbps
4	Medium-high	No transmit rate configured	10 Mbps
5	Medium-high	No transmit rate configured	20 Mbps

In this example, queues are divided into three sets based on their priority:

- High priority set—Consists of queue 1 and queue 2. Packets use 40 Mbps (20+20) of the available bandwidth (100 Mbps) and are transmitted first. Because of positive credit, the configured transmit rate is not considered.
- Medium-high priority set—Consists of queue 4 and queue 5. Packets use 30 Mbps (10+20) of the remaining 60 Mbps bandwidth. Because of positive credit, the transmit rate is not considered. If the queues had negative credit, they would receive an equal share of the leftover bandwidth because no transmit rate is configured.
- Low priority set—Consists of queue 0 and queue 3. Packets share the 30 Mbps of leftover bandwidth based on the configured transmit rate. The distribution of bandwidth is in proportion to the assigned percentages. Because the total assigned percentage is 40 (10 + 30), each queue receives a share of

bandwidth accordingly. Thus queue 0 receives 7.5 Mbps (10/40 x 30), and queue 3 receives 22.5 Mbps (30/40 x 30).

RELATED DOCUMENTATION

Schedulers Overview | 117
Default Scheduler Settings | 123
Example: Configuring Class-of-Service Schedulers on a Security Device | 133
Scheduler Buffer Size Overview | 139
Example: Configuring and Applying Scheduler Maps | 149

Excess Bandwidth Sharing and Minimum Logical Interface Shaping

The default excess bandwidth sharing proportional rate is 32.65 Mbps (128 Kbps x 255). In order to have better weighed fair queuing (WFQ) accuracy among queues, the shaping rate configured should be larger than the excess bandwidth sharing proportional rate. Some examples are shown in Table 24 on page 126.

Table 24: Shaping Rates and WFQ Weights

Shaping Rate	Configured Queue Transmit Rate	WFQ Weight	Total Weights
10 Mbps	(30, 40, 25, 5)	(22, 30, 20, 4)	76
33 Mbps	(30, 40, 25, 5)	(76, 104, 64, 13)	257
40 Mbps	(30, 40, 25, 5)	(76, 104.64, 13)	257

With a 10-Mbps shaping rate, the total weights are 76. This is divided among the four queues according to the configured transmit rate. Note that when the shaping rate is larger than the excess bandwidth sharing proportional rate of 32.65 Mbps, the total weight on the logical interface is 257 and the WFQ accuracy will be the same.

When using the IOC (40x1GE IOC or 4x10GE IOC) on a Juniper Networks device, there are circumstances when you should configure excess bandwidth sharing and minimum logical interface shaping.

RELATED DOCUMENTATION

Schedulers Overview | 117

Excess Bandwidth Sharing Proportional Rates | 127

Calculated Weights Mapped to Hardware Weights | 128

Weight Allocation with Only Shaping Rates or Unshaped Logical Interfaces | 129

Shared Bandwidth Among Logical Interfaces | 131

Excess Bandwidth Sharing Proportional Rates

To determine a good excess bandwidth-sharing proportional rate to configure, choose the largest CIR (guaranteed rate) among all the logical interfaces (units). If the logical units have PIRs (shaping rates) only, then choose the largest PIR rate. However, this is not ideal if a single logical interface has a large WRR rate. This method can skew the distribution of traffic across the queues of the other logical interfaces. To avoid this issue, set the excess bandwidth-sharing proportional rate to a lower value on the logical interfaces where the WRR rates are concentrated. This improves the bandwidth sharing accuracy among the queues on the same logical interface. However, the excess bandwidth sharing for the logical interface with the larger WRR rate is no longer proportional.

As an example, consider five logical interfaces on the same physical port, each with four queues, all with only PIRs configured and no CIRs. The WRR rate is the same as the PIR for the logical interface. The excess bandwidth is shared proportionally with a rate of 40 Mbps. The traffic control profiles for the logical interfaces are shown in Table 25 on page 127.

Table 25: Example Shaping Rates and WFQ Weights

Shaping Rate	Configured Queue Transmit Rate	WFQ Weight	Total Weights
(Unit 0) 10 Mbps	(95, 0, 0, 5)	(60, 0, 0, 3)	63
(Unit 1) 20 Mbps	(25, 25, 25, 25)	(32, 32, 32, 32)	128
(Unit 2) 40 Mbps	(40, 30, 20, 10)	(102, 77, 51, 26)	255
(Unit 3) 200 Mbps	(70, 10, 10, 10)	(179, 26, 26, 26)	255
(Unit 4) 2 Mbps	(25, 25, 25, 25)	(5, 5, 5, 5)	20

Even though the maximum transmit rate for the queue on logical interface unit 3 is 200 Mbps, the excess bandwidth-sharing proportional rate is kept at a much lower value. Within a logical interface, this method provides a more accurate distribution of weights across queues. However, the excess bandwidth is now shared equally between unit 2 and unit 3 (total weights = 255).

RELATED DOCUMENTATION

Schedulers Overview | 117
Excess Bandwidth Sharing and Minimum Logical Interface Shaping | 126
Calculated Weights Mapped to Hardware Weights | 128
Weight Allocation with Only Shaping Rates or Unshaped Logical Interfaces | 129
Shared Bandwidth Among Logical Interfaces | 131

Calculated Weights Mapped to Hardware Weights

The calculated weight in a traffic control profile is mapped to hardware weight, but the hardware only supports a limited WFQ profile. The weights are rounded to the nearest hardware weight according to the values in Table 26 on page 128.

Table 26: Rounding Configured Weights to Hardware Weights

Traffic Control Profile Number	Number of Traffic Control Profiles	Weights	Maximum Error
1–16	16	1–16 (interval of 1)	50.00%
17–29	13	18–42 (interval of 2)	6.25%
30–35	6	45–60 (interval of 3)	1.35%
36–43	8	64–92 (interval of 4)	2.25%
44–49	6	98–128 (interval of 6)	3.06%
50–56	7	136–184 (interval of 8)	3.13%

Table 26: Rounding Configured Weights to Hardware Weights *(Continued)*

Traffic Control Profile Number	Number of Traffic Control Profiles	Weights	Maximum Error
57–62	6	194–244 (interval of 10)	2.71%
63–63	1	255–255 (interval of 11)	2.05%

As shown in Table 26 on page 128, the calculated weight of 18.9 is mapped to a hardware weight of 18, because 18 is closer to 18.9 than 20 (an interval of 2 applies in the range of 18 to 42).

RELATED DOCUMENTATION

Schedulers Overview | **117**

Excess Bandwidth Sharing and Minimum Logical Interface Shaping | **126**

Excess Bandwidth Sharing Proportional Rates | **127**

Weight Allocation with Only Shaping Rates or Unshaped Logical Interfaces | **129**

Shared Bandwidth Among Logical Interfaces | **131**

Weight Allocation with Only Shaping Rates or Unshaped Logical Interfaces

Logical interfaces with only shaping rates (PIRs) or unshaped logical interfaces (units) are given a weight of 10. A logical interface with a small guaranteed rate (CIR) might get an overall weight less than 10. To allocate a higher share of the excess bandwidth to logical interfaces with a small guaranteed rate in comparison to the logical interfaces with only shaping rates configured, a minimum weight of 20 is given to the logical interfaces with guaranteed rates configured.

For example, a logical interface configuration with five units is shown in Table 27 on page 129.

Table 27: Allocating Weights with PIR and CIR on Logical Interfaces

Logical Interface (Unit)	Traffic Control Profile	WRR Percentages	Weights
Unit 1	PIR 100 Mbps	95, 0, 0, 5	10, 1, 1, 1

Table 27: Allocating Weights with PIR and CIR on Logical Interfaces *(Continued)*

Logical Interface (Unit)	Traffic Control Profile	WRR Percentages	Weights
Unit 2	CIR 20 Mbps	25, 25, 25, 25	64, 64, 64, 64
Unit 3	PIR 40 Mbps, CIR 20 Mbps	50, 30, 15, 5	128, 76, 38, 13
Unit 4	Unshaped	95, 0, 0, 5	10, 1, 1, 1
Unit 5	CIR 1 Mbps	95, 0, 0, 5	10, 1, 1, 1

The weights for these units are calculated as follows:

- The excess bandwidth-sharing proportional rate is the maximum CIR among all the logical interfaces which is 20 Mbps (unit 2).
- Unit 1 has a PIR and unit 4 is unshaped. The weight for these units is 10.
- The weight for unit 1 queue 0 is 9.5 (10 x 95%), which translates to a hardware weight of 10.
- The weight for unit 1 queue 1 is 0 (0 x 0%) but though the weight is zero, a weight of 1 is assigned to give minimal bandwidth to queues with zero WRR.
- Unit 5 has a very small CIR (1 Mbps), and a weight of 20 is assigned to units with a small CIR.
- The weight for unit 5 queue 0 is 19 (20 x 95%), which translates to a hardware weight of 18.
- Unit 3 has a CIR of 20 Mbps, which is the same as the excess bandwidth-sharing proportional rate, so it has a total weight of 255.
- The weight of unit 3 queue 0 is 127.5 (255 x 50%), which translates to a hardware weight of 128.

RELATED DOCUMENTATION

Schedulers Overview | 117

Excess Bandwidth Sharing and Minimum Logical Interface Shaping | 126

Excess Bandwidth Sharing Proportional Rates | 127

Calculated Weights Mapped to Hardware Weights | 128

Shared Bandwidth Among Logical Interfaces | 131

Shared Bandwidth Among Logical Interfaces

As a simple example showing how bandwidth is shared among the logical interfaces, assume that all traffic is sent on queue 0. Assume also that there is a 40-Mbps load on all of the logical interfaces. Configuration details are shown in Table 28 on page 131.

Table 28: Example of Shared Bandwidth Among Logical Interfaces

Logical Interface (Unit)	Traffic Control Profile	WRR Percentages	Weights
Unit 1	PIR 100 Mbps	95, 0, 0, 5	10, 1, 1, 1
Unit 2	CIR 20 Mbps	25, 25, 25, 25	64, 64, 64, 64
Unit 3	PIR 40 Mbps, CIR 20 Mbps	50, 30, 15, 5	128, 76, 38, 13
Unit 4	Unshaped	95, 0, 0, 5	10, 1, 1, 1

When the port is shaped at 40 Mbps, because units 2 and 3 have a guaranteed rate (CIR) configured, both units 2 and 3 get 20 Mbps of shared bandwidth.

When the port is shaped at 100 Mbps, because units 2 and 3 have a guaranteed rate (CIR) configured, each of them can transmit 20 Mbps. On units 1, 2, 3, and 4, the 60 Mbps of excess bandwidth is shaped according to the values shown in Table 29 on page 131.

Table 29: First Example of Bandwidth Sharing

Logical Interface (Unit)	Calculation	Bandwidth
1	10 / (10+64+128+10) x 60 Mbps	2.83 Mbps
2	64 / (10+64+128+10) x 60 Mbps	18.11 Mbps
3	128 / (10+64+128+10) x 60 Mbps	36.22 Mbps
4	10 (10+64+128+10) x 60 Mbps	2.83 Mbps

However, unit 3 only has 20 Mbps extra (PIR and CIR) configured. This means that the leftover bandwidth of 16.22 Mbps (36.22 Mbps – 20 Mbps) is shared among units 1, 2, and 4. This is shown in Table 30 on page 132.

Table 30: Second Example of Bandwidth Sharing

Logical Interface (Unit)	Calculation	Bandwidth
1	10 / (10+64+128+10) x 16.22 Mbps	1.93 Mbps
2	64 / (10+64+128+10) x 16.22 Mbps	12.36 Mbps
4	10 (10+64+128+10) x 16.22 Mbps	1.93 Mbps

Finally, Table 31 on page 132 shows the resulting allocation of bandwidth among the logical interfaces when the port is configured with a 100-Mbps shaping rate.

Table 31: Final Example of Bandwidth Sharing

Logical Interface (Unit)	Calculation	Bandwidth
1	2.83 Mbps + 1.93 Mbps	4.76 Mbps
2	20 Mbps + 18.11 Mbps + 12.36 Mbps	50.47 Mbps
3	20 Mbps + 20 Mbps	40 Mbps
4	2.83 Mbps + 1.93 Mbps	4.76 Mbps

RELATED DOCUMENTATION

- Schedulers Overview | 117
- Excess Bandwidth Sharing and Minimum Logical Interface Shaping | 126
- Excess Bandwidth Sharing Proportional Rates | 127
- Calculated Weights Mapped to Hardware Weights | 128
- Weight Allocation with Only Shaping Rates or Unshaped Logical Interfaces | 129

Example: Configuring Class-of-Service Schedulers on a Security Device

IN THIS SECTION

- Requirements | 133
- Overview | 133
- Configuration | 134
- Verification | 138

This example shows how to configure CoS schedulers on a device.

Requirements

Before you begin, determine the buffer size allocation method to use. See "Scheduler Buffer Size Overview" on page 139.

Overview

An individual device interface has multiple queues assigned to store packets temporarily before transmission. To determine the order in which to service the queues, the device uses a round-robin scheduling method based on priority and the queue's weighted round-robin (WRR) credits. Junos OS schedulers allow you to define the priority, bandwidth, delay buffer size, rate control status, and RED drop profiles to be applied to a particular queue for packet transmission.

You configure schedulers to assign resources, priorities, and drop profiles to output queues. By default, only queues 0 and 3 have resources assigned.

NOTE: Juniper Network devices support hierarchical schedulers, including per-unit schedulers.

In this example, you configure a best-effort scheduler called be-scheduler. You set the priority as low and the buffer size to 40. You set the be-scheduler transmit-rate remainder percentage to 40. You configure an expedited forwarding scheduler called ef-scheduler and set the priority as high and the buffer size to 10. You set the ef-scheduler transmit-rate remainder percentage to 50.

Then you configure an assured forwarding scheduler called af-scheduler and set the priority as high and buffer size to 45. You set an assured forwarding scheduler transmit rate to 45. You then configure a drop profile map for assured forwarding as low and high priority. (DiffServ can have a RED drop profile associated with assured forwarding.)

Finally, you configure a network control scheduler called nc-scheduler and set the priority as low and buffer size to 5. You set a network control scheduler transmit rate to 5.

Table 32 on page 134 shows the schedulers created in this example.

Table 32: Sample Schedulers

Scheduler	For CoS Traffic Type	Assigned Priority	Allocated Portion of Queue Buffer	Allocated Portion of Remainder (Transmit Rate)
`be-scheduler`	Best-effort traffic	Low	40 percent	40 percent
`ef-scheduler`	Expedited forwarding traffic	High	10 percent	50 percent
`af-scheduler`	Assured forwarding traffic	High	45 percent	—
`nc-scheduler`	Network control traffic	Low	5 percent	—

Configuration

IN THIS SECTION

- Procedure | 135

Procedure

CLI Quick Configuration

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

```
set class-of-service schedulers be-scheduler  priority low buffer-size percent 40
set class-of-service schedulers be-scheduler  transmit-rate remainder 40
set class-of-service schedulers  ef-scheduler  priority high buffer-size percent 10
set class-of-service schedulers ef-scheduler  transmit-rate remainder 50
set class-of-service schedulers  af-scheduler  priority high buffer-size percent 45
set class-of-service schedulers  af-scheduler  transmit-rate percent 45
set class-of-service schedulers  af-scheduler  drop-profile-map loss-priority low protocol any
drop-profile af-normal
set class-of-service schedulers  af-scheduler  drop-profile-map loss-priority high protocol any
drop-profile af-with-PLP
set class-of-service schedulers nc-scheduler  priority low buffer-size percent 5
set class-of-service schedulers nc-scheduler  transmit-rate percent 5
```

Step-by-Step Procedure

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 Junos OS CLI User Guide.

To configure CoS schedulers:

1. Configure a best-effort scheduler.

   ```
   [edit]
   user@host# edit class-of-service schedulers be-scheduler
   ```

2. Specify a best-effort scheduler priority and buffer size.

   ```
   [edit class-of-service schedulers be-scheduler]
   user@host# set priority low
   user@host# set buffer-size percent 40
   ```

3. Configure a remainder option for a best-effort scheduler transmit rate.

```
[edit class-of-service schedulers be-scheduler]
user@host# set transmit-rate remainder 40
```

4. Configure an expedited forwarding scheduler.

```
[edit]
user@host# edit class-of-service schedulers ef-scheduler
```

5. Specify an expedited forwarding scheduler priority and buffer size.

```
[edit class-of-service schedulers ef-scheduler]
user@host# set priority high
user@host# set buffer-size percent 10
```

6. Configure a remainder option for an expedited forwarding scheduler transmit rate.

```
[edit class-of-service schedulers ef-scheduler]
user@host# set transmit-rate remainder 50
```

7. Configure an assured forwarding scheduler.

```
[edit]
user@host# edit class-of-service schedulers af-scheduler
```

8. Specify an assured forwarding scheduler priority and buffer size.

```
[edit class-of-service schedulers af-scheduler]
user@host# set priority high
user@host# set buffer-size percent 45
```

9. Configure an assured forwarding scheduler transmit rate.

```
[edit class-of-service schedulers af-scheduler]
user@host# set transmit-rate percent 45
```

10. Configure a drop profile map for assured forwarding low and high priority.

```
[edit class-of-service schedulers af-scheduler]
user@host# set drop-profile-map loss-priority low protocol any drop-profile af-normal
user@host# set drop-profile-map loss-priority high protocol any drop-profile af-with-PLP
```

11. Configure a network control scheduler.

```
[edit]
user@host# edit class-of-service schedulers nc-scheduler
```

12. Specify a network control scheduler priority and buffer size.

```
[edit class-of-service schedulers nc-scheduler]
user@host# set priority low
user@host# set buffer-size percent 5
```

13. Configure a network control scheduler transmit rate.

```
[edit class-of-service schedulers nc-scheduler]
user@host# set transmit-rate percent 5
```

Results

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

```
[edit]
    user@host# show class-of-service
    schedulers {
    be-scheduler {
            transmit-rate remainder 40;
            buffer-size percent 40;
            priority low;
        }
        ef-scheduler {
            transmit-rate remainder 50;
            buffer-size percent 10;
            priority high;
        }
        af-scheduler {
            transmit-rate percent 45;
            buffer-size percent 45;
            priority high;
            drop-profile-map loss-priority low protocol any drop-profile af-normal;
            drop-profile-map loss-priority high protocol any drop-profile af-with-PLP;
        }
        nc-scheduler {
            transmit-rate percent 5;
            buffer-size percent 5;
            priority low;
        }
    }
```

If you are done configuring the device, enter `commit` from configuration mode.

Verification

IN THIS SECTION

- Verifying Schedulers Configuration | 138

Verifying Schedulers Configuration

Purpose

Verify that the schedulers are configured properly.

Action

From operational mode, enter the `show class-of-service` command.

RELATED DOCUMENTATION

Schedulers Overview | 117

Default Scheduler Settings | 123

Example: Configuring and Applying Scheduler Maps | 149

Transmission Scheduling Overview | 124

Scheduler Buffer Size Overview

IN THIS SECTION

- Maximum Delay Buffer Size for vSRX Virtual Firewall Interfaces | 140
- Delay Buffer Size Allocation Methods | 140
- Delay Buffer Sizes for Queues | 141

Large bursts of traffic from faster interfaces can cause congestion and dropped packets on slower interfaces that have small delay buffers. For example, a Juniper Networks device operating at the edge of the network can drop a portion of the burst traffic it receives on an interface from a faster interface on a router at the network core. On Juniper Networks devices, you can configure large delay buffers.

To ensure that traffic is queued and transmitted properly on slower interfaces, you can configure a buffer size larger than the default maximum.

This section contains the following topics:

Maximum Delay Buffer Size for vSRX Virtual Firewall Interfaces

For a vSRX Virtual Firewall virtual machine, 1 Gbps interfaces have a default delay buffer time of 1 second, a maximum buffer time of 32 seconds, and a maximum buffer size of 128 MB. Use the following CLI command to set the maximum delay buffer time for a scheduler:

```
set class-of-service schedulers be-scheduler buffer-size temporal 32m
```

On a logical vSRX Virtual Firewall interface, the delay buffer size for a queue that does not have a specific shaping rate acts as a guaranteed minimum buffer size, and the queue is allowed to grow without any packet drops if the queue size is less then the guaranteed buffer size.

The sum of the guaranteed delay buffer sizes for all the queues acts as a pool that can be shared among the queues that do not have a specific shaping rate.

NOTE: The delay buffers are used to control the size of the queues, but do not represent actual memory. The packet buffer pool contains the actual memory used to store packets.

Packets are tail-dropped (100% probability) from the queue if:

- The total buffer limit would be exceeded.
- The queue size would exceed the total free buffer size.
- The packet buffer pool is less then 25% free and the queue exceeds the guaranteed minimum buffer size.
- The packet buffer pool is only 5% free (or less).

Packets also can be dropped by a RED profile (RED-dropped) if the queue size exceeds the guaranteed buffer size. The queue size will be restricted to be less then or equal to the free shared buffers available.

NOTE: Support for vSRX Virtual Firewall virtual machines depends on the Junos OS release in your installation.

Delay Buffer Size Allocation Methods

You can specify delay buffer sizes for each queue using schedulers. The queue buffer can be specified as a period of time (microseconds) or as a percentage of the total buffer or as the remaining buffer. Table 33 on page 141 shows different methods that you can specify for buffer allocation in queues.

Table 33: Delay Buffer Size Allocation Methods

Buffer Size Allocation Method	Description
Percentage	A percentage of the total buffer.
Temporal	A period of time, value in microseconds. When you configure a temporal buffer, you must also configure a transmit rate. The system calculates the queue buffer size by multiplying the available bandwidth of the interface times the configured temporal value and transmit rate. When you specify a temporal method, the drop profile is assigned a static buffer and the system starts dropping packets once the queue buffer size is full. By default, the other buffer types are assigned dynamic buffers that use surplus transmission bandwidth to absorb bursts of traffic.
Remainder	The remaining buffer available. The remainder is the percentage buffer that is not assigned to other queues. For example, if you assign 40 percent of the delay buffer to queue 0, allow queue 3 to keep the default allotment of 5 percent, and assign the remainder to queue 7, then queue 7 uses approximately 55 percent of the delay buffer. Optionally, you can specify the percentage of the remainder to be used for allocating the buffer size of the scheduler on a prorated basis. If the remainder percentage is not specified, the remainder value will be shared equally.

Delay Buffer Sizes for Queues

You specify delay buffer sizes for queues using schedulers. The system calculates the buffer size of a queue based on the buffer allocation method you specify for it in the scheduler. See Table 33 on page 141 for different buffer allocation methods and Table 34 on page 142 for buffer size calculations.

Table 34: Delay Buffer Allocation Method and Queue Buffer

Buffer Size Allocation Method	Queue Buffer Calculation	Example
Percentage	*available interface bandwidth* x *configured buffer size percentage* x *maximum delay buffer time* = *queue buffer*	Suppose you configure a queue on a 64 Kbps interface to use 30 percent of the available delay buffer size. The system uses the maximum available delay buffer time (4 seconds) and allocates the queue 9600 bytes of delay buffer: 64 Kbps x 0.3 x 4 s=76,800 bits=9,600 bytes
Temporal	*available interface bandwidth* x *configured transmit rate percentage* x *configured temporal buffer size* = *queue buffer*	Suppose you configure a queue on a 64 Kbps interface to use 3,000,000 microseconds (3 seconds) of delay buffer, and you configure the transmission rate to be 20 percent. The queue receives 4800 bytes of delay buffer: 64 Kbps x 0.2 x 3 s=38,400 bits=4,800 bytes If you configure a temporal value that exceeds the maximum available delay buffer time, the queue is allocated the buffer remaining after buffers are allocated for the other queues. Suppose you configure a temporal value of 6,000,000 microseconds on a 64 Kbps interface. Because this value exceeds the maximum allowed value of 4,000,000 microseconds, the queue is allocated the remaining delay buffer.

When you specify the buffer size as a percentage, the system ignores the transmit rate and calculates the buffer size based only on the buffer size percentage.

RELATED DOCUMENTATION

- Schedulers Overview | **117**
- Default Scheduler Settings | **123**
- Example: Configuring Class-of-Service Schedulers on a Security Device | **133**
- Example: Configuring and Applying Scheduler Maps | **149**
- Transmission Scheduling Overview | **124**

Configuring Large Delay Buffers in CoS

You can configure very large delay buffers using the `buffer-size-temporal` command combined with the `q-pic-large-buffer` command. The `buffer-size temporal` option in combination with `q-pic-large-buffer` can create extra-large delay buffer allocations for one or several queues on an interface.

> **NOTE**: If the configured buffer size is too low, the buffer size for the forwarding class defaults to 9192 and the following log message is displayed: "fwdd_cos_set_delay_bandwidth:queue:16 delay buffer size (1414) too low, setting to default 9192."

Configuring Large Delay Buffers

The following configuration applies to the examples that follow:

1. Configure two VLANs (one ingress, one egress) on one interface. No interface shaping rate is initially defined for this configuration.

   ```
   [edit]
   set interfaces ge-0/0/3 per-unit-scheduler
   set interfaces ge-0/0/3 vlan-tagging
   set interfaces ge-0/0/3 unit 102 vlan-id 102
   set interfaces ge-0/0/3 unit 102 family inet address 203.0.113.2/24
   set interfaces ge-0/0/3 unit 201 vlan-id 201
   set interfaces ge-0/0/3 unit 201 family inet address 198.51.100.2/24
   set routing-options static route 192.02.1/32 next-hop 198.51.100.3
   ```

2. Enable the `q-pic-large-buffer` option on the same PIC, in addition to the `buffer-size temporal` option on the queue, to create a large buffer on the queue:

   ```
   [edit]
   set chassis fpc 0 pic 0 q-pic-large-buffer
   ```

   > **NOTE**: The CLI does not provide a warning when you use `buffer-size temporal` without `q-pic-large-buffer`. When you use `buffer-size temporal`, verify that the configuration also includes the `q-pic-large buffer` command.

3. Define four forwarding-classes (queue names) for the four queues:

```
[edit]
set class-of-service forwarding-classes queue 0 be-Queue0
set class-of-service forwarding-classes queue 1 video-Queue1
set class-of-service forwarding-classes queue 2 voice-Queue2
set class-of-service forwarding-classes queue 3 nc-Queue3
```

4. Configure the forwarding classes (queue names) included in a scheduler map, applied to the egress VLAN:

```
[edit]
set class-of-service interfaces ge-0/0/3 unit 201 scheduler-map schedMapM
set class-of-service scheduler-maps schedMapM forwarding-class be-Queue0 scheduler be-Scheduler0
set class-of-service scheduler-maps schedMapM forwarding-class video-Queue1 scheduler video-Scheduler1
set class-of-service scheduler-maps schedMapM forwarding-class voice-Queue2 scheduler voice-Scheduler2
set class-of-service scheduler-maps schedMapM forwarding-class nc-Queue3 scheduler nc-Scheduler3
```

5. Set the queue priorities. Only queue priorities are initially defined, not transmit rates or buffer sizes.

```
[edit]
set class-of-service schedulers be-Scheduler0 priority low
set class-of-service schedulers video-Scheduler1 priority medium-low
set class-of-service schedulers voice-Scheduler2 priority medium-high
set class-of-service schedulers nc-Scheduler3 priority high
```

Example: Simple Configuration Using Four Queues

This configuration allocates 12,500,000 bytes of buffer for each of the four queues. To avoid exceeding the limits of the delay buffer calculation, this initial example has no interface shaping rate, scheduler transmit rate, or scheduler buffer size percent configuration.

1. Specify the maximum 4-second delay buffer on each of the four queues:

```
[edit]
set class-of-service schedulers be-Scheduler0 buffer-size temporal 4m
set class-of-service schedulers video-Scheduler1 buffer-size temporal 4m
set class-of-service schedulers voice-Scheduler2 buffer-size temporal 4m
set class-of-service schedulers nc-Scheduler3 buffer-size temporal 4m
```

Specifying `buffer-size temporal` on some or all queues uses implicit (default) or explicit transmit rate percentages as the buffer-size percentages of the temporal values for those queues. Because there are no explicitly specified transmit rate percentages, divide 100 percent by the number of configured queues (queues with schedulers configured in the scheduler map) to get the implicit (default) per-queue transmit rate percentages. Each queue gets an implicit (default) transmit rate of 100% / 4 = 25%.

In this example, specifying the maximum 4-second delay on each queue, with no shaping rate on the interface and implicit (default) per-queue transmit rates of 25 percent, the total buffer for all temporal 4m queues on an interface = 4 seconds * 100,000,000 maximum interface bps / 8 bits/byte = 4 seconds * 12,500,000 bytes = 50,000,000 bytes. Each queue specifying temporal 4m gets 25% * 50,000,000 = 12,500,000 bytes.

2. Add a shaping rate of 4 Mbps to the interface:

```
[edit]
set class-of-service interfaces ge-0/0/3 unit 201 shaping-rate 4m
```

The total buffer for all temporal 4m queues on an interface = 4 sec * 4,000,000 bps shaping-rate / 8 bits/byte = 4 sec * 500,000 bytes = 2,000,000 bytes. Therefore, each queue specifying temporal 4m receives 25% * 2,000,000 = 500,000 bytes.

When using `buffer-size temporal` on any interface queues, if you also use the `transmit-rate percent` command, or the `buffer-size percent` command, or both commands, on any of the interface queues, the buffer size calculations become more complex and the limits of available queue depth might be reached. If the configuration attempts to exceed the available memory, then at commit time two system log messages appear in the `/var/log/messages` file, the interface class-of-service configuration is ignored, and the interface class-of-service configuration reverts to the two-queue defaults:

```

Mar 11 11:02:10.239  elma-n4 elma-n4 COSMAN_FWDD: queue mem underflow for ge-0/0/3
Mar 11 11:02:10.240  elma-n4 elma-n4 cosman_compute_install_sched_params: Failed to compute
scheduler params for ge-0/0/3.Hence retaining defaults
```

When configuring `buffer-size temporal` along with `transmit-rate percent` or `buffer-size percent`, or both, you must monitor the system log to see whether the available queue depth limit has been reached.

Example: Using buffer-size temporal with Explicit transmit-rate percent Commands

To add explicit transmit rates to all four queues:

```
[edit]
set class-of-service schedulers be-Scheduler0 transmit-rate percent 10
set class-of-service schedulers video-Scheduler1 transmit-rate percent 25
set class-of-service schedulers voice-Scheduler2 transmit-rate percent 25
set class-of-service schedulers nc-Scheduler3 transmit-rate percent 40
```

For example, if an interface is shaped to 4 Mbps, the transmit rate percentage of 10 for a queue means that the bandwidth share for the specific queue is 0.4 Mbps. The queues are allocated portions of the 2,000,000 bytes of total buffer available for temporal queues on this interface, proportionally to their transmit rates. The four queues get 200,000, 500,000, 500,000, and 800,000 bytes of delay buffer, respectively.

To avoid exceeding the queue depth limits and triggering system log messages and default configuration behavior, when configuring queues with `buffer-size temporal` and `transmit rate percent` and other (non-temporal) queues with `buffer-size percent`, the following configuration rule must be followed: When one or more queues on an interface are configured with `buffer-size temporal`, the sum of the temporal queues explicitly configured transmit rate percentages plus other non-temporal queues explicitly configured buffer size percentages must not exceed 100 percent.

If the total of the temporal queues transmit rate percentages and the non-temporal queues buffer-size percentages exceeds 100 percent, the `queue mem underflow` and `Failed to compute scheduler params` system log messages appear in the messages log, the explicitly configured CLI CoS configuration for the interface is ignored, and the interface reverts to a two-queue default CoS configuration.

When `buffer-size temporal` is specified on a queue, if `transmit-rate percent` is also configured on the same queue, the queue depth configured is based on the fractional bandwidth for the queue as obtained by the specified `transmit-rate percent`.

In addition to temporal delay times specified for one or more queues using buffer size temporal, there is another delay time automatically computed for the entire interface. This interface delay time is distributed across all non-temporal queues, proportionally to their implicit (default) or explicit transmit-rate percentages. If `q-pic-large-buffer` is not enabled, the interface delay time defaults to 100 ms. As shown in Table 35 on page 147, when `q-pic-large-buffer` is enabled, interface delay time is calculated according to configured shaping rate for the interface. Because the shaping-rate configured in the example above was 4 Mbps (> 2,048,000 bps), the interface delay time for the configuration is 100 msec.

Table 35: Interface Delay Times Enabled By q-pic-large-buffer

Configured Shaping Rate (bps)	Interface Delay Time (msec) Used for Non-Temporal Queues with q-pic-large-buffer Enabled	Default Delay Time Used (msec) Without q-pic-large-buffer
64,000-255,999	4000	100
256,000 - 511,999	2000	100
512,000 - 102,3999	1000	100
1,024,000 - 2,047,999	500	100
>= 2,048,000	100	100

This example properly computes the delay buffer limits on both temporal and non-temporal queues:

1. Substitute `buffer-size percent` for `buffer-size temporal` on queues 0 and 1:

   ```
   [edit]
   delete class-of-service schedulers be-Scheduler0 buffer-size temporal 4m
   delete class-of-service schedulers video-Scheduler1 buffer-size temporal 4m
   set class-of-service schedulers be-Scheduler0    buffer-size percent 10
   set class-of-service schedulers video-Scheduler1 buffer-size percent 25
   ```

   This deletes the requirement for hard-specified 4 seconds of buffering and replaces it with a proportional limit of 10 percent (or 25 percent) of the total interface delay time for the non-temporal queues. In both cases, the queue depth is calculated based on the share of the interface bandwidth for the specific queues. Total Interface Non-Temporal Queue Memory = shaping-rate * Interface delay time (Table 1) = 4 Mbps * 0.1 seconds = 500,000 bytes per second * 0.1 seconds = 50,000 bytes, therefore queues 0 and 1 get 10% * 50,000 = 5000 bytes and 25% * 50,000 = 12,500 bytes of delay buffer, respectively.

2. Configure `buffer-size temporal` on queues 2 and 3:

   ```
   [edit]
   set class-of-service schedulers voice-Scheduler2 buffer-size temporal 4m
   set class-of-service schedulers voice-Scheduler2 transmit-rate percent 25
   ```

```
set class-of-service schedulers nc-Scheduler3    buffer-size temporal 4m
set class-of-service schedulers nc-Scheduler3    transmit-rate percent 40
```

Queues 2 and 3 still get 500,000 and 800,000 bytes of delay buffer, respectively, as previously calculated. This configuration obeys the rule that the sum of the temporal queues transmit rate percentages (25% + 40% = 65%), plus the non-temporal queues buffer size percentages (10% + 25% = 35%) do not exceed 100% (65% + 35% <= 100%).

The following example exceeds the delay buffer limit, triggering the system log messages and the default, two-queue class-of-service behavior.

Increase the buffer-size percentage from 25 percent to 26 percent for non-temporal queue 1:

```
[edit]
set class-of-service schedulers video-Scheduler1 buffer-size percent 26
```

This violates the configuration rule that the sum of the non-temporal queues buffer-size percentages (10% + 26% = 36%), plus the temporal queues transmit rate percentages (25% + 40% = 65%) now exceed 100% (36% + 65% = 101%). Therefore, the following two system log messages appear in the `/var/log/messages` file:

```


Mar 23 18:08:23  elma-n4 elma-n4 COSMAN_FWDD: %PFE-3: queue mem underflow for ge-0/0/3 q_num(3)
Mar 23 18:08:23  elma-n4 elma-n4 cosman_compute_install_sched_params: %PFE-3: Failed to compute
scheduler params for ge-0/0/3.Hence retaining defaults
```

When the delay buffer limits are exceeded, the CLI-configured class-of-service settings are not used and the default class-of-service configuration (the default scheduler-map) is assigned to the interface. This uses two queues: the forwarding-class best-effort (queue 0) has transmit rate percent 95 and buffer-size percent 95 and the forwarding-class network-control (queue 3) has the transmit rate percent 5 and buffer-size percent 5.

```


queue 0: 1,187,500 Bytes
queue 1:     9,192 Bytes
queue 2:     9,192 Bytes
queue 3:    62,500 Bytes
```

RELATED DOCUMENTATION

Example: Configuring and Applying Scheduler Maps | 149
Scheduler Buffer Size Overview | 139

Example: Configuring and Applying Scheduler Maps

IN THIS SECTION

- Requirements | 149
- Overview | 149
- Configuration | 150
- Verification | 152

This example shows how to configure and apply a scheduler map to a device’s interface.

Requirements

Before you begin:

- Create and configure the forwarding classes. See *Configuring a Custom Forwarding Class for Each Queue*.
- Create and configure the schedulers. See "Example: Configuring Class-of-Service Schedulers on a Security Device" on page 133.

Overview

After you define a scheduler, you can include it in a scheduler map, which maps a specified forwarding class to a scheduler configuration. You configure a scheduler map to assign a forwarding class to a scheduler, and then apply the scheduler map to any interface that must enforce DiffServ CoS.

After they are applied to an interface, the scheduler maps affect the hardware queues, packet schedulers, and RED drop profiles.

In this example, you create the scheduler map diffserv-cos-map and apply it to the device's Ethernet interface ge-0/0/0. The map associates the mf-classifier forwarding classes to the schedulers as shown in Table 36 on page 150.

Table 36: Sample diffserv-cos-map Scheduler Mapping

mf-classifier Forwarding Class	For CoS Traffic Type	diffserv-cos-map Scheduler
be-class	Best-effort traffic	be-scheduler
ef-class	Expedited forwarding traffic	ef-scheduler
af-class	Assured forwarding traffic	af-scheduler
nc-class	Network control traffic	nc-scheduler

Configuration

IN THIS SECTION

- Procedure | 150

Procedure

CLI Quick Configuration

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

```
 set class-of-service scheduler-maps diffserv-cos-map forwarding-class be-class scheduler be-
scheduler
set class-of-service scheduler-maps diffserv-cos-map forwarding-class ef-class scheduler ef-
scheduler
set class-of-service scheduler-maps diffserv-cos-map forwarding-class af-class scheduler af-
scheduler
set class-of-service scheduler-maps diffserv-cos-map forwarding-class nc-class scheduler nc-
scheduler
set class-of-service  interfaces ge-0/0/0 unit 0 scheduler-map diffserv-cos-map
```

Step-by-Step Procedure

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 Junos OS CLI User Guide.

To configure and apply a scheduler map to a device’s interface:

1. Configure a scheduler map for DiffServ CoS.

   ```
   [edit class-of-service]
   user@host# edit scheduler-maps diffserv-cos-map
   ```

2. Configure a best-effort forwarding class and scheduler.

   ```
    [edit class-of-service scheduler-maps diffserv-cos-map]
   user@host# set forwarding-class be-class scheduler be-scheduler
   ```

3. Configure an expedited forwarding class and scheduler.

   ```
   [edit class-of-service scheduler-maps diffserv-cos-map]
   user@host# set forwarding-class ef-class scheduler ef-scheduler
   ```

4. Configure an assured forwarding class and scheduler.

   ```
   [edit class-of-service scheduler-maps diffserv-cos-map]
   user@host# set forwarding-class af-class scheduler af-scheduler
   ```

5. Configure a network control class and scheduler.

   ```
   [edit class-of-service scheduler-maps diffserv-cos-map]
   user@host# set forwarding-class nc-class scheduler nc-scheduler
   ```

6. Apply the scheduler map to an interface.

   ```
   [edit class-of-service]
   user@host# set interfaces ge-0/0/0 unit 0 scheduler-map diffserv-cos-map
   ```

Results

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

```
[edit]
user@host# show class-of-service
interfaces {
    ge-0/0/0 {
        unit 0 {
            scheduler-map diffserv-cos-map;
        }
    }
}
scheduler-maps {
    diffserv-cos-map {
        forwarding-class be-class scheduler be-scheduler;
        forwarding-class ef-class scheduler ef-scheduler;
        forwarding-class af-class scheduler af-scheduler;
        forwarding-class nc-class scheduler nc-scheduler;
    }
}
```

If you are done configuring the device, enter `commit` from configuration mode.

Verification

IN THIS SECTION

- Verifying the Scheduler Map Configuration | 152

Verifying the Scheduler Map Configuration

Purpose

Verify that scheduler maps are configured properly.

Action

From operational mode, enter the `show class-of-service` command.

RELATED DOCUMENTATION

Default Schedulers Overview

Configuring Schedulers

Configuring Scheduler Maps

Applying Scheduler Maps and Shaping Rate to Logical Interfaces and VLANs

By default, output scheduling is not enabled on logical interfaces. Logical interfaces without shaping configured share a default scheduler. This scheduler has a committed information rate (CIR) that equals 0. (The CIR is the guaranteed rate.) The default scheduler has a peak information rate (PIR) that equals the physical interface shaping rate.

NOTE: If you apply a shaping rate, you must keep in mind that the transit statistics for physical interfaces are obtained from the packet forwarding engine, but the traffic statistics are supplied by the PIC. Therefore, if shaping is applied to the PIC, the count of packets in the transit statistics fields do not always agree with the counts in the traffic statistics. For example, the IPv6 transit statistics will not necessarily match the traffic statistics on the interface. However, at the logical interface level, both transit and traffic statistics are obtained from the Packet Forwarding Engine and will not show any difference.

Logical interface scheduling (also called *per-unit scheduling*) allows you to enable multiple output queues on a logical interface and associate customized output scheduling and shaping for each queue.

NOTE: Ingress scheduling does not support logical interface scheduling.

For supported PICs, you can configure a shaping rate for a VLAN or logical interface and oversubscribe the physical interface by including the `shaping-rate` statement at the `[edit class-of-service traffic-control-profiles]` hierarchy level. With this configuration approach, you can independently control the delay-buffer rate, as described in *Oversubscribing Interface Bandwidth*.

Physical interfaces (for example, `xe-0/0/0`, `xe-0/0/0:0`, and `ge-0/0/0`) support scheduling with any encapsulation type pertinent to that physical interface. For a single port, you cannot apply scheduling to the physical interface if you apply scheduling to one or more of the associated logical interfaces.

> **NOTE**: In the Junos OS implementation, the term *logical interfaces* generally refers to interfaces you configure by including the `unit` statement at the `[edit interfaces` *`interface-name`*`]` hierarchy level. As such, logical interfaces have the *`logical`* descriptor at the end of the interface name, as in `xe-0/0/0.1` or `xe-0/0/0:0.1`, where the logical unit number is `1`.
>
> Within the `[edit class-of-service]` hierarchy level, you cannot use the `.`*`logical`* descriptor when you assign properties to logical interfaces. Instead, you must include the `unit` statement in the configuration. For example:
>
> ```
> [edit class-of-service]
> user@host# set interfaces xe-0/0/0 unit 0 scheduler-map map1
> ```

Table 37 on page 154 shows the MICs and MPCs that support fine-grained queuing and scheduling.

Table 37: Fine-Grained Queuing and Scheduling Support by MIC or MPC Type

MPC	MIC	Supported	Example Configuration
Fixed Configuration MPCs			
16x10GE MPC	No	Yes	`[edit class-of-service interfaces ge-0/0/0 unit 1]` `scheduler-map map-1;`
32x10GE MPC4E	No	Yes	`[edit class-of-service interfaces ge-0/0/0 unit 1]` `scheduler-map map-1;`
2x100GE + 8x10GE MPC4E	No	Yes	`[edit class-of-service interfaces ge-0/0/0 unit 1]` `scheduler-map map-1;`
6x40GE + 24x10GE MPC5E	No	No	No

Table 37: Fine-Grained Queuing and Scheduling Support by MIC or MPC Type *(Continued)*

MPC	MIC	Supported	Example Configuration
6x40GE + 24x10GE MPC5EQ	No	Yes	`[edit class-of-service interfaces ge-0/0/0 unit 1]` `scheduler-map map-1;`
2x100GE + 4x10GE MPC5E	No	No	No
2x100GE + 4x10GE MPC5EQ	No	Yes	`[edit class-of-service interfaces ge-0/0/0 unit 1]` `scheduler-map map-1;`
MPCs			
MPC1	No	No	No
MPC1E	No	No	No

MPC1 Q	Any supported MIC	Yes	Example of supported configuration:
			`[edit class-of-service interfaces ge-0/0/0 unit 1]`
			`scheduler-map map-1;`

MPC1E Q	Any supported MIC	Yes	Example of supported configuration:
			`[edit class-of-service interfaces ge-0/0/0 unit 1]`
			`scheduler-map map-1;`

MPC2	No	No	No
MPC2E	No	No	No

Table 37: Fine-Grained Queuing and Scheduling Support by MIC or MPC Type *(Continued)*

MPC	MIC	Supported	Example Configuration

MPC2 Q	Any supported MIC	Yes	Example of supported configuration:
			`[edit class-of-service interfaces ge-0/0/0 unit 1]`
			`scheduler-map map-1;`

MPC2E Q	Any supported MIC	Yes	Example of supported configuration:
			`[edit class-of-service interfaces ge-0/0/0 unit 1]`
			`scheduler-map map-1;`

MPC2 EQ	Any supported MIC	Yes	Example of supported configuration:
			`[edit class-of-service interfaces ge-0/0/0 unit 1]`
			`scheduler-map map-1;`

MPC2E EQ	Any supported MIC	Yes	Example of supported configuration:
			`[edit class-of-service interfaces ge-0/0/0 unit 1]`
			`scheduler-map map-1;`

MPC2E P	No	No	No

MPC3E	10-Gigabit Ethernet MIC with SFP+	Yes	Example of supported configuration:
			`[edit class-of-service interfaces xe-0/0/0 unit 1]`
			`scheduler-map map-1;`

	40-Gigabit Ethernet MIC with QSFP+	Yes	Example of supported configuration:
			`[edit class-of-service interfaces et-0/0/0 unit 1]`
			`scheduler-map map-1;`

Table 37: Fine-Grained Queuing and Scheduling Support by MIC or MPC Type *(Continued)*

MPC	MIC	Supported	Example Configuration

	100-Gigabit Ethernet MIC with CXP	Yes	Example of supported configuration:
			`[edit class-of-service interfaces et-0/0/0 unit 1]`
			`scheduler-map map-1;`

MPC6E	Any supported MIC	Yes	Example of supported configuration:
			`[edit class-of-service interfaces et-0/0/0 unit 1]`
			`scheduler-map map-1;`

To configure scheduling on logical interfaces:

1. Enable per-unit scheduling on the interface by including the `per-unit-scheduler` statement at the `[edit interfaces` *`interface-name`*`]` hierarchy level:

   ```
   [edit interfaces interface-name]
   per-unit-scheduler;
   ```

   When including the `per-unit-scheduler` statement, you must also include the `vlan-tagging` statement or the `flexible-vlan-tagging` statement (to apply scheduling to VLANs) or the `encapsulation frame-relay` statement (to apply scheduling to DLCIs) at the `[edit interfaces` *`interface-name`*`]` hierarchy level.

   When you include this statement, the maximum number of VLANs supported is 768 on a single-port Gigabit Ethernet IQ PIC. On a dual-port Gigabit Ethernet IQ PIC, the maximum number is 384.

   See *Scaling of Per-VLAN Queuing on Non-Queuing MPCs* for scaling information on non-queuing MPCs.

2. Associate a scheduler with the interface by including the `scheduler-map` statement at the `[edit class-of-service interfaces` *`interface-name`* `unit` *`logical-unit-number`*`]` hierarchy level:

   ```
   [edit class-of-service interfaces interface-name unit logical-unit-number]
   scheduler-map map-name;
   ```

   Alternatively, associate a scheduler with the interface by including the `scheduler-map` statement at the `[edit class-of-service traffic-control-profiles` *`traffic control profile name`*`]` hierarchy level and then include the `output-traffic-control-profile` statement at the `[edit class-of-service interfaces` *`interface name`* `unit` *`logical unit number`*`]` hierarchy level.

```
[edit class-of-service traffic-control-profiles traffic control profile name]
scheduler-map map-name;
```

```
[edit class-of-service interfaces interface-name unit logical-unit-number]
output-traffic-control-profile traffic-control-profile-name;
```

3. Configure shaping on the interface by including the `shaping-rate` statement at the `[edit class-of-service interfaces` *`interface-name`* `unit` *`logical-unit-number`*`]` hierarchy level:

```
[edit class-of-service interfaces interface-name unit logical-unit-number]
shaping-rate rate;
```

NOTE: You can also apply the shaping rate to the traffic control profile.

By default, the logical interface bandwidth is the average of unused bandwidth for the number of logical interfaces that require default bandwidth treatment. 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).

RELATED DOCUMENTATION

per-unit-scheduler

Example: Applying Scheduling and Shaping to VLANs

Example: Applying Scheduling and Shaping to VLANs

IN THIS SECTION

Requirements | 159

- Overview | 159
- Configuration | 159

This example shows how to apply schedulers to individual logical interfaces.

Requirements

This example uses the following hardware and software components:

- MX Series routers
- Any supported Junos release

Overview

By default, output scheduling is not enabled on logical interfaces. Logical interfaces without shaping configured share a default scheduler. *Logical interface scheduling* (also called *per-unit scheduling*) allows you to enable multiple output queues on a logical interface and associate customized scheduling and shaping for each queue.

To enable per-unit scheduling, include the `per-unit-scheduler` statement at the `[edit interfaces` *`interface name`*`]` hierarchy level. When per-unit schedulers are enabled, you can define dedicated schedulers for logical interfaces by including the `scheduler-map` statement at the `[edit class-of-service interfaces` *`interface name`* `unit` *`logical unit number`*`]` hierarchy level. Alternatively, you can include the `scheduler-map` statement at the `[edit class-of-service traffic-control-profiles` *`traffic control profile name`*`]` hierarchy level and then include the `output-traffic-control-profile` statement at the `[edit class-of-service interfaces` *`interface name`* `unit` *`logical unit number`*`]` hierarchy level.

This example shows how to define schedulers for logical interfaces through the use of traffic control profiles.

Configuration

IN THIS SECTION

- CLI Quick Configuration | 160
- Procedure | 162
- Results | 164

CLI Quick Configuration

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

```
set interfaces xe-9/0/3 per-unit-scheduler
set interfaces xe-9/0/3 vlan-tagging
set interfaces xe-9/0/3 unit 1 vlan-id 101
set interfaces xe-9/0/3 unit 1 family inet address 10.1.1.1/24
set interfaces xe-9/0/3 unit 2 vlan-id 102
set interfaces xe-9/0/3 unit 2 family inet address 10.2.1.1/24
set class-of-service classifiers inet-precedence c8 forwarding-class be loss-priority low code-
points 000
set class-of-service classifiers inet-precedence c8 forwarding-class ef loss-priority low code-
points 001
set class-of-service classifiers inet-precedence c8 forwarding-class af loss-priority low code-
points 010
set class-of-service classifiers inet-precedence c8 forwarding-class nc loss-priority low code-
points 011
set class-of-service classifiers inet-precedence c8 forwarding-class be1 loss-priority low code-
points 100
set class-of-service classifiers inet-precedence c8 forwarding-class ef1 loss-priority low code-
points 101
set class-of-service classifiers inet-precedence c8 forwarding-class af1 loss-priority low code-
points 110
set class-of-service classifiers inet-precedence c8 forwarding-class nc1 loss-priority low code-
points 111
set class-of-service forwarding-classes queue 0 be
set class-of-service forwarding-classes queue 1 ef
set class-of-service forwarding-classes queue 2 af
set class-of-service forwarding-classes queue 3 nc
set class-of-service forwarding-classes queue 4 be1
set class-of-service forwarding-classes queue 5 ef1
set class-of-service forwarding-classes queue 6 af1
set class-of-service forwarding-classes queue 7 nc1
set class-of-service traffic-control-profiles tcp_ifd shaping-rate 2500000000
set class-of-service traffic-control-profiles tcp_ifd overhead-accounting bytes -20
set class-of-service traffic-control-profiles tcp_gold scheduler-map gold
set class-of-service traffic-control-profiles tcp_gold shaping-rate 2500000000
set class-of-service traffic-control-profiles tcp_gold overhead-accounting bytes -20
set class-of-service traffic-control-profiles tcp_gold guaranteed-rate 1g
set class-of-service traffic-control-profiles tcp_silver scheduler-map silver
set class-of-service traffic-control-profiles tcp_silver shaping-rate 1g
set class-of-service traffic-control-profiles tcp_silver overhead-accounting bytes -20
set class-of-service traffic-control-profiles tcp_silver guaranteed-rate 500m
set class-of-service interfaces xe-9/0/3 output-traffic-control-profile tcp_ifd
set class-of-service interfaces xe-9/0/3 unit 1 output-traffic-control-profile tcp_gold
set class-of-service interfaces xe-9/0/3 unit 2 output-traffic-control-profile tcp_silver
set class-of-service scheduler-maps gold forwarding-class be1 scheduler gold_internet
set class-of-service scheduler-maps gold forwarding-class ef1 scheduler gold_video
set class-of-service scheduler-maps gold forwarding-class af1 scheduler gold_voice
set class-of-service scheduler-maps gold forwarding-class nc1 scheduler gold_reserved
set class-of-service scheduler-maps silver forwarding-class be scheduler silver_internet
set class-of-service scheduler-maps silver forwarding-class ef scheduler silver_video
set class-of-service scheduler-maps silver forwarding-class af scheduler silver_voice
set class-of-service scheduler-maps silver forwarding-class nc scheduler silver_reserved
set class-of-service schedulers gold_internet excess-rate percent 40
set class-of-service schedulers gold_internet buffer-size percent 20
set class-of-service schedulers gold_internet priority low
set class-of-service schedulers gold_video transmit-rate percent 50
set class-of-service schedulers gold_video buffer-size percent 50
set class-of-service schedulers gold_voice shaping-rate percent 10
set class-of-service schedulers gold_voice buffer-size percent 10
set class-of-service schedulers gold_voice priority strict-high
set class-of-service schedulers gold_reserved excess-rate percent 20
set class-of-service schedulers gold_reserved buffer-size percent 10
set class-of-service schedulers gold_reserved priority low
set class-of-service schedulers silver_internet excess-rate percent 40
set class-of-service schedulers silver_internet buffer-size percent 20
set class-of-service schedulers silver_internet priority low
set class-of-service schedulers silver_video transmit-rate percent 50
set class-of-service schedulers silver_video buffer-size percent 50
set class-of-service schedulers silver_voice shaping-rate percent 10
set class-of-service schedulers silver_voice buffer-size percent 10
set class-of-service schedulers silver_voice priority strict-high
set class-of-service schedulers silver_reserved excess-rate percent 20
set class-of-service schedulers silver_reserved buffer-size percent 10
set class-of-service schedulers silver_reserved priority low
```

Procedure

Step-by-Step Procedure

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

1. Configure the device interfaces.

```
[edit interfaces]
user@PE1# set xe-9/0/3 per-unit-scheduler
user@PE1# set xe-9/0/3 vlan-tagging
user@PE1# set xe-9/0/3 unit 1 vlan-id 101
user@PE1# set xe-9/0/3 unit 1 family inet address 10.1.1.1/24
user@PE1# set xe-9/0/3 unit 2 vlan-id 102
user@PE1# set xe-9/0/3 unit 2 family inet address 10.2.1.1/24
```

2. Configure the classifiers.

```
[edit class-of-service]
user@PE1# set classifiers inet-precedence c8 forwarding-class be loss-priority low code-
points 000
user@PE1# set classifiers inet-precedence c8 forwarding-class ef loss-priority low code-
points 001
user@PE1# set classifiers inet-precedence c8 forwarding-class af loss-priority low code-
points 010
user@PE1# set classifiers inet-precedence c8 forwarding-class nc loss-priority low code-
points 011
user@PE1# set classifiers inet-precedence c8 forwarding-class be1 loss-priority low code-
points 100
user@PE1# set classifiers inet-precedence c8 forwarding-class ef1 loss-priority low code-
points 101
user@PE1# set classifiers inet-precedence c8 forwarding-class af1 loss-priority low code-
points 110
user@PE1# set classifiers inet-precedence c8 forwarding-class nc1 loss-priority low code-
points 111
```

3. Configure the forwarding classes.

```
[edit class-of-service]
user@PE1# set forwarding-classes queue 0 be
user@PE1# set forwarding-classes queue 1 ef
user@PE1# set forwarding-classes queue 2 af
user@PE1# set forwarding-classes queue 3 nc
user@PE1# set forwarding-classes queue 4 be1
user@PE1# set forwarding-classes queue 5 ef1
user@PE1# set forwarding-classes queue 6 af1
user@PE1# set forwarding-classes queue 7 nc1
```

4. Configure the traffic control profiles.

```
[edit class-of-service]
user@PE1# set traffic-control-profiles tcp_ifd shaping-rate 2500000000
user@PE1# set traffic-control-profiles tcp_ifd overhead-accounting bytes -20
user@PE1# set traffic-control-profiles tcp_gold scheduler-map gold
user@PE1# set traffic-control-profiles tcp_gold shaping-rate 2500000000
user@PE1# set traffic-control-profiles tcp_gold overhead-accounting bytes -20
user@PE1# set traffic-control-profiles tcp_gold guaranteed-rate 1g
user@PE1# set traffic-control-profiles tcp_silver scheduler-map silver
user@PE1# set traffic-control-profiles tcp_silver shaping-rate 1g
user@PE1# set traffic-control-profiles tcp_silver overhead-accounting bytes -20
user@PE1# set traffic-control-profiles tcp_silver guaranteed-rate 500m
```

5. Map the traffic control profiles to their respective physical or logical interface.

```
[edit class-of-service]
user@PE1# set interfaces xe-9/0/3 output-traffic-control-profile tcp_ifd
user@PE1# set interfaces xe-9/0/3 unit 1 output-traffic-control-profile tcp_gold
user@PE1# set interfaces xe-9/0/3 unit 2 output-traffic-control-profile tcp_silver
```

6. Configure the scheduler maps.

```
[edit class-of-service]
user@PE1# set scheduler-maps gold forwarding-class be1 scheduler gold_internet
user@PE1# set scheduler-maps gold forwarding-class ef1 scheduler gold_video
user@PE1# set scheduler-maps gold forwarding-class af1 scheduler gold_voice
```

```
user@PE1# set scheduler-maps gold forwarding-class nc1 scheduler gold_reserved
user@PE1# set scheduler-maps silver forwarding-class be scheduler silver_internet
user@PE1# set scheduler-maps silver forwarding-class ef scheduler silver_video
user@PE1# set scheduler-maps silver forwarding-class af scheduler silver_voice
user@PE1# set scheduler-maps silver forwarding-class nc scheduler silver_reserved
```

7. Configure the schedulers.

```
[edit class-of-service]
user@PE1# set schedulers gold_internet excess-rate percent 40
user@PE1# set schedulers gold_internet buffer-size percent 20
user@PE1# set schedulers gold_internet priority low
user@PE1# set schedulers gold_video transmit-rate percent 50
user@PE1# set schedulers gold_video buffer-size percent 50
user@PE1# set schedulers gold_voice shaping-rate percent 10
user@PE1# set schedulers gold_voice buffer-size percent 10
user@PE1# set schedulers gold_voice priority strict-high
user@PE1# set schedulers gold_reserved excess-rate percent 20
user@PE1# set schedulers gold_reserved buffer-size percent 10
user@PE1# set schedulers gold_reserved priority low
user@PE1# set schedulers silver_internet excess-rate percent 40
user@PE1# set schedulers silver_internet buffer-size percent 20
user@PE1# set schedulers silver_internet priority low
user@PE1# set schedulers silver_video transmit-rate percent 50
user@PE1# set schedulers silver_video buffer-size percent 50
user@PE1# set schedulers silver_voice shaping-rate percent 10
user@PE1# set schedulers silver_voice buffer-size percent 10
user@PE1# set schedulers silver_voice priority strict-high
user@PE1# set schedulers silver_reserved excess-rate percent 20
user@PE1# set schedulers silver_reserved buffer-size percent 10
user@PE1# set schedulers silver_reserved priority low
```

Results

From configuration mode, confirm your configuration by entering the `show interfaces` and `show class-of-service` commands. If the output does not display the intended configuration, repeat the instructions in this example to correct the configuration.

```
user@PE1# show interfaces
interfaces {
    xe-9/0/3 {
        per-unit-scheduler;
        vlan-tagging;
        unit 1 {
            vlan-id 101;
            family inet {
                address 10.1.1.1/24;
            }
        }
        unit 2 {
            vlan-id 102;
            family inet {
                address 10.2.1.1/24;
            }
        }
    }
}
```

```
user@PE1# show class-of-service
class-of-service {
    classifiers {
        inet-precedence c8 {
            forwarding-class be {
                loss-priority low code-points 000;
            }
            forwarding-class ef {
                loss-priority low code-points 001;
            }
            forwarding-class af {
                loss-priority low code-points 010;
            }
            forwarding-class nc {
                loss-priority low code-points 011;
            }
            forwarding-class be1 {
                loss-priority low code-points 100;
            }
            forwarding-class ef1 {
                loss-priority low code-points 101;
            }
            forwarding-class af1 {
                loss-priority low code-points 110;
            }
            forwarding-class nc1 {
                loss-priority low code-points 111;
            }
        }
    }
    forwarding-classes {
        queue 0 be;
        queue 1 ef;
        queue 2 af;
        queue 3 nc;
        queue 4 be1;
        queue 5 ef1;
        queue 6 af1;
        queue 7 nc1;
    }
    traffic-control-profiles {
        tcp_ifd {
            shaping-rate 2500000000;
            overhead-accounting bytes -20;
        }
        tcp_gold {
            scheduler-map gold;
            shaping-rate 2500000000;
            overhead-accounting bytes -20;
            guaranteed-rate 1g;
        }
        tcp_silver {
            scheduler-map silver;
            shaping-rate 1g;
            overhead-accounting bytes -20;
            guaranteed-rate 500m;
        }
    }
    interfaces {
        xe-9/0/3 {
            output-traffic-control-profile tcp_ifd;
            unit 1 {
                output-traffic-control-profile tcp_gold;
            }
            unit 2 {
                output-traffic-control-profile tcp_silver;
            }
        }
    }
    scheduler-maps {
        gold {
            forwarding-class be1 scheduler gold_internet;
            forwarding-class ef1 scheduler gold_video;
            forwarding-class af1 scheduler gold_voice;
            forwarding-class nc1 scheduler gold_reserved;
        }
        silver {
            forwarding-class be scheduler silver_internet;
            forwarding-class ef scheduler silver_video;
            forwarding-class af scheduler silver_voice;
            forwarding-class nc scheduler silver_reserved;
        }
    }
    schedulers {
        gold_internet {
            excess-rate percent 40;
            buffer-size percent 20;
            priority low;
        }
        gold_video {
            transmit-rate percent 50;
            buffer-size percent 50;
        }
        gold_voice {
            shaping-rate percent 10;
            buffer-size percent 10;
            priority strict-high;
        }
        gold_reserved {
            excess-rate percent 20;
            buffer-size percent 10;
            priority low;
        }
        silver_internet {
            excess-rate percent 40;
            buffer-size percent 20;
            priority low;
        }
        silver_video {
                transmit-rate percent 50;
                buffer-size percent 50;
            }
            silver_voice {
                shaping-rate percent 10;
                buffer-size percent 10;
                priority strict-high;
            }
            silver_reserved {
                excess-rate percent 20;
                buffer-size percent 10;
                priority low;
            }
        }
    }
```

If you are done configuring the device, enter **commit** from configuration mode.

RELATED DOCUMENTATION

Applying Scheduler Maps and Shaping Rate to DLCIs and VLANs

per-unit-scheduler

CHAPTER 7

Removing Delays with Strict-Priority Queues

IN THIS CHAPTER

- Strict-Priority Queue Overview | **169**
- Understanding Strict-Priority Queues | **170**
- Example: Configuring Priority Scheduling | **171**
- Example: Configuring Strict-Priority Queuing | **174**
- Example: Configuring CoS Non-Strict Priority Scheduling | **188**

Strict-Priority Queue Overview

You can configure one queue per interface to have strict-priority, which causes delay-sensitive traffic, such as voice traffic, to be removed and forwarded with minimum delay. Packets that are queued in a strict-priority queue are removed before packets in other queues, including high-priority queues.

The strict-high-priority queuing feature allows you to configure traffic policing that prevents lower priority queues from being starved. The strict-priority queue does not cause starvation of other queues because the configured policer allows the queue to exceed the configured bandwidth only when other queues are not congested. If the interface is congested, the software directs strict-priority queues to the configured bandwidth.

To prevent queue starvation of other queues, you must configure an output (egress) policer that defines a limit for the amount of traffic that the queue can service. The software services all traffic in the strict-priority queue that is under the defined limit. When strict-priority traffic exceeds the limit, the policer marks the traffic in excess of the limit as out-of-profile. If the output port is congested, the software drops out-of-profile traffic.

You can also configure a second policer with an upper limit. When strict-priority traffic exceeds the upper limit, the software drops the traffic in excess of the upper limit, regardless of whether the output port is congested. This upper-limit policer is not a requirement for preventing starvation of the lower priority queues. The policer for the lower limit, which marks the packets as out-of-profile, is sufficient to prevent starvation of other queues.

RELATED DOCUMENTATION

Understanding Strict-Priority Queues | 170
Example: Configuring Priority Scheduling | 171
Example: Configuring Strict-Priority Queuing | 174

Understanding Strict-Priority Queues

You use strict-priority queuing and policing as follows:

- Identify delay-sensitive traffic by configuring a behavior aggregate (BA) or multifield (MF) classifier.
- Minimize delay by assigning all delay-sensitive packets to the strict-priority queue.
- Prevent starvation on other queues by configuring a policer that checks the data stream entering the strict-priority queue. The policer defines a lower bound, marks the packets that exceed the lower bound as out-of-profile, and drops the out-of-profile packets if the physical interface is congested. If there is no congestion, the software forwards all packets, including the out-of-profile packets.
- Optionally, configure another policer that defines an upper bound and drops the packets that exceed the upper bound, regardless of congestion on the physical interface.

To configure strict-priority queuing and prevent starvation of other queues, include the `priority strict-high` statement at the `[edit class-of-service schedulers` *`scheduler-name`*`]` hierarchy level and the `if-exceeding` and `then out-of-profile` statements at the `[edit firewall policer` *`policer-name`*`]` hierarchy level:

```
[edit class-of-service schedulers scheduler-name]
priority strict-high;

[edit firewall policer policer-name]
if-exceeding {
    bandwidth-limit bps;
    bandwidth-percent number;
    burst-size-limit bytes;
}
then out-of-profile;
```

RELATED DOCUMENTATION

Strict-Priority Queue Overview | 169

Example: Configuring Priority Scheduling | 171
Example: Configuring Strict-Priority Queuing | 174

Example: Configuring Priority Scheduling

IN THIS SECTION

- Requirements | 171
- Overview | 171
- Configuration | 171
- Verification | 174

This example shows how to configure priority scheduling so important traffic receives better access to the outgoing interface.

Requirements

Before you begin, review how to assign forwarding classes. See "Example: Assigning Forwarding Classes to Output Queues" on page 86.

Overview

In this example, you configure CoS and a scheduler called be-sched with a medium-low priority. Then you configure scheduler map be-map to associate be-sched with the best-effort forwarding class. Finally, you apply be-map to interface ge-0/0/0.

Configuration

IN THIS SECTION

- Procedure | 172

Procedure

CLI Quick Configuration

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

```
set class-of-service schedulers be-sched priority medium-low
set class-of-service scheduler-maps be-map forwarding-class best-effort scheduler be-sched
set class-of-service interfaces ge-0/0/0 scheduler-map be-map
```

Step-by-Step Procedure

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 Junos OS CLI User Guide.

To configure priority scheduling:

1. Configure CoS and a scheduler.

```
[edit]
user@host# edit class-of-service
user@host# edit schedulers be-sched
```

2. Set a priority.

```
[edit class-of-service schedulers be-sched]
user@host# set priority medium-low
```

3. Configure a scheduler map.

```
[edit]
user@host# edit class-of-service
user@host# edit scheduler-maps be-map
```

4. Specify the best-effort forwarding class.

```
[edit class-of-service scheduler-maps be-map]
user@host# set forwarding-class best-effort scheduler be-sched
```

5. Apply best-effort map to an interface.

```
[edit]
user@host# edit class-of-service
user@host# set interfaces ge-0/0/0 scheduler-map be-map
```

Results

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

```
[edit]
    user@host# show class-of-service
    interfaces {
    ge-0/0/0 {
    scheduler-map be-map;
        }
        }
        scheduler-maps {
            be-map {
            forwarding-class best-effort scheduler be-sched;
        }
        }
        schedulers {
            be-sched {
        priority medium-low;
    }
}
```

If you are done configuring the device, enter `commit` from configuration mode.

Verification

IN THIS SECTION

- Verifying Priority Scheduling | 174

Verifying Priority Scheduling

Purpose

Verify that the priority scheduling is configured properly on a device.

Action

From configuration mode, enter the `show class-of-service` command.

RELATED DOCUMENTATION

Strict-Priority Queue Overview | 169

Understanding Strict-Priority Queues | 170

Example: Configuring Strict-Priority Queuing | 174

Example: Configuring Strict-Priority Queuing

IN THIS SECTION

- Requirements | 175
- Overview | 175
- Configuration | 175
- Verification | 187

This example shows how to configure strict-priority queuing and prevent starvation of other queues.

Requirements

Before you begin, review how to create and configure forwarding classes. See "Forwarding Classes Overview" on page 77.

Overview

In this example, you create a BA classifier to classify traffic based on the IP precedence of the packet. The classifier defines IP precedence value 101 as voice traffic and 000 as data traffic. You assign forwarding-class priority queue 0 to voice traffic and queue 1 as data traffic. You then configure the scheduler map as corp-map and voice scheduler as voice-sched.

Then you set the priority for the voice traffic scheduler as strict-high and for the data traffic scheduler as strict-low. You apply the BA classifier to input interface ge-0/0/0 and apply the scheduler map to output interface e1-1/0/0. You then configure two policers called voice-drop and voice-excess. You set the burst size limit and bandwidth limit for voice-drop policer and for voice-excess policer. You then create a firewall filter that includes the new policers and add the policer to the term.

Finally, you apply the filter to output interface e1-1/0/1 and set the IP address as 203.0.113.1/24.

Configuration

IN THIS SECTION

- Configuring a BA Classifier | **176**
- Configuring Forwarding Classes | **177**
- Configuring a Scheduler Map | **178**
- Configuring a Scheduler | **180**
- Applying a BA Classifier to an Input Interface | **181**
- Applying a Scheduler Map to an Output Interface | **182**
- Configuring Two Policers | **183**
- Applying a Filter to an Output Interface | **186**

Configuring a BA Classifier

CLI Quick Configuration

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

```
set class-of-service classifiers inet-precedence corp-traffic forwarding-class voice-class loss-
priority low code-points 101
set class-of-service classifiers inet-precedence corp-traffic forwarding-class data-class loss-
priority high code-points 000
```

Step-by-Step Procedure

To configure a BA classifier:

1. Create a BA classifier and set the IP precedence value for voice traffic.

```
[edit]
user@host# edit class-of-service classifiers inet-precedence corp-traffic forwarding-class
voice-class loss-priority low
user@host# set code-points 101
```

2. Create a BA classifier and set the IP precedence value for data traffic.

```
[edit]
user@host# edit class-of-service classifiers inet-precedence corp-traffic forwarding-class
data-class loss-priority high
user@host# set code-points 000
```

Results

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

```
[edit]
    user@host# show class-of-service
    classifiers {
    inet-precedence corp-traffic {
    forwarding-class voice-class {
        loss-priority low code-points 101;
        }
        forwarding-class data-class {
            loss-priority high code-points 000;
        }
    }
}
```

If you are done configuring the device, enter `commit` from configuration mode.

Configuring Forwarding Classes

CLI Quick Configuration

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

```
set class-of-service forwarding-classes queue 0 voice-class
set class-of-service forwarding-classes queue 1 data-class
```

Step-by-Step Procedure

To configure forwarding classes:

1. Assign priority queuing to voice traffic.

   ```
   [edit]
   user@host# set class-of-service forwarding-classes queue 0 voice-class
   ```

2. Assign priority queuing to data traffic.

   ```
   [edit]
   user@host# set class-of-service forwarding-classes queue 1 data-class
   ```

Results

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

```
[edit]
user@host# show class-of-service
forwarding-classes {
    queue 0 voice-class;
    queue 1 data-class;
}
```

If you are done configuring the device, enter `commit` from configuration mode.

Configuring a Scheduler Map

CLI Quick Configuration

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

```
set class-of-service scheduler-maps corp-map forwarding-class voice-class scheduler voice-sched
set class-of-service scheduler-maps corp-map forwarding-class data-class scheduler data-sched
```

Step-by-Step Procedure

To configure a scheduler map:

1. Configure a scheduler map and voice scheduler.

```
[edit]
user@host# edit class-of-service scheduler-maps corp-map forwarding-class voice-class
user@host# set scheduler voice-sched
```

2. Configure a scheduler map and data scheduler.

```
[edit]
user@host# edit class-of-service scheduler-maps corp-map forwarding-class data-class
user@host# set scheduler data-sched
```

Results

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

```
[edit]
user@host# show class-of-service
scheduler-maps {
    corp-map {
    forwarding-class voice-class scheduler voice-sched;
    forwarding-class data-class scheduler data-sched;
    }
}
```

If you are done configuring the device, enter `commit` from configuration mode.

Configuring a Scheduler

CLI Quick Configuration

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

```
set class-of-service schedulers voice-sched priority strict-high
set class-of-service schedulers data-sched priority lowset xxx
```

Step-by-Step Procedure

To configure schedulers:

1. Configure a voice traffic scheduler and set the priority.

   ```
   [edit]
   user@host# edit class-of-service schedulers voice-sched
   user@host# set priority strict-high
   ```

2. Configure a data traffic scheduler and set the priority.

   ```
   [edit]
   user@host# edit class-of-service schedulers data-sched
   user@host# set priority low
   ```

Results

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

```
[edit]
user@host# show class-of-service
schedulers {
    voice-sched {
    priority strict-high;
```

```
    }
    data-sched {
        priority low;
    }
}
```

If you are done configuring the device, enter `commit` from configuration mode.

Applying a BA Classifier to an Input Interface

CLI Quick Configuration

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

```
set class-of-service interfaces ge-0/0/0 unit 0 classifiers inet-precedence corp-traffic
```

Step-by-Step Procedure

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 Junos OS CLI User Guide.

To apply a BA classifier to an input interface:

1. Configure an interface.

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

2. Apply a BA classifier to an input interface.

   ```
   [edit class-of-service interfaces ge-0/0/0 unit 0]
   user@host# set classifiers inet-precedence corp-traffic
   ```

Results

From configuration mode, confirm your configuration by entering the `show class-of-service interfaces` command. If the output does not display the intended configuration, repeat the configuration instructions in this example to correct it.

```
[edit]
    user@host# show class-of-service interfaces
    ge-0/0/0 {
    unit 0 {
    classifiers {
        inet-precedence corp-traffic;
        }
    }
}
```

If you are done configuring the device, enter `commit` from configuration mode.

Applying a Scheduler Map to an Output Interface

CLI Quick Configuration

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

```
set class-of-service interfaces e1-1/0/0 unit 0 scheduler-map corp-map
```

Step-by-Step Procedure

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 Junos OS CLI User Guide.

To apply the scheduler map to an output interface:

1. Configure an interface.

```
[edit]
user@host# edit class-of-service interfaces e1-1/0/0 unit 0
```

2. Apply a scheduler map to an output interface.

```
[edit class-of-service interfaces e1-1/0/0 unit 0]
user@host# set scheduler-map corp-map
```

Results

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

```
[edit]
    user@host# show class-of-service
    interfaces {
    e1-1/0/0 {
    unit 0 {
        scheduler-map corp-map;
        }
    }
}
```

If you are done configuring the device, enter `commit` from configuration mode.

Configuring Two Policers

CLI Quick Configuration

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

```
set firewall policer voice-drop if-exceeding burst-size-limit 200000 bandwidth-limit 2000000
set firewall policer voice-drop then discard
set firewall policer voice-excess if-exceeding burst-size-limit 200000 bandwidth-limit 1000000
set firewall policer voice-excess then out-of-profile
set firewall filter voice-term term 01 from forwarding-class voice-class
set firewall filter voice-term term 01 then policer voice-drop next term
set firewall filter voice-term term 02 from forwarding-class voice-class
set firewall filter voice-term term 02 then policer voice-excess accept
```

Step-by-Step Procedure

To configure two policers:

1. Configure a policer voice drop.

```
[edit]
user@host# edit firewall policer voice-drop
user@host# set if-exceeding burst-size-limit 200000 bandwidth-limit 2000000
user@host# set then discard
```

2. Configure a policer voice excess.

```
[edit]
user@host# edit firewall policer voice-excess
user@host# set if-exceeding burst-size-limit 200000 bandwidth-limit 1000000
user@host# set then out-of-profile
```

3. Create a firewall filter that includes the new policers.

```
[edit]
user@host# edit firewall filter voice-term term 01
user@host# set from forwarding-class voice-class
user@host# set then policer voice-drop next term
```

4. Add the policer to the term.

```
[edit]
user@host# edit firewall filter voice-term term 02
user@host# set from forwarding-class voice-class
user@host# set then policer voice-excess accept
```

Results

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

```
[edit]
    user@host# show firewall
    policer voice-drop {
    if-exceeding {
    bandwidth-limit 2m;
        burst-size-limit 200k;
        }
        then discard;
        }
        policer voice-excess {
            if-exceeding {
            bandwidth-limit 1m;
            burst-size-limit 200k;
            }
                then out-of-profile;
            }
            filter voice-term {
                term 01 {
            from {
            forwarding-class voice-class;
            }
            then {
        policer voice-drop;
        next term;
            }
        }
        term 02 {
            from {
```

```
            forwarding-class voice-class;
            }
            then {
                policer voice-excess;
            accept;
        }
    }
}
```

If you are done configuring the device, enter `commit` from configuration mode.

Applying a Filter to an Output Interface

CLI Quick Configuration

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

```
set interfaces e1-1/0/1 unit 0 family inet filter output voice-term
set interfaces e1-1/0/1 unit 0 family inet address 203.0.113.1/24
```

Step-by-Step Procedure

To apply a filter to an output interface:

1. Apply a filter to an interface.

   ```
   [edit]
   user@host# edit interfaces e1-1/0/1 unit 0 family inet filter output
   user@host# set voice-term
   ```

2. Set an IP address.

   ```
   [edit]
   user@host# set interfaces e1-1/0/1 unit 0 family inet address 203.0.113.1/24
   ```

Results

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

```
[edit]
    user@host# show interfaces
    e1-1/0/1 {
    unit 0 {
    family inet {
        filter {
        output voice-term;
        }
        address 203.0.113.1/24;
        }
    }
}
```

If you are done configuring the device, enter `commit` from configuration mode.

Verification

IN THIS SECTION

- Verifying the Scheduler Map | 187
- Verifying the Interfaces | 188
- Verifying the Interface Queues | 188

Confirm that the configuration is working properly.

Verifying the Scheduler Map

Purpose

Verify that the scheduler map is configured properly.

Action

From operational mode, enter the `show class-of-service scheduler-map corp-map` command.

Verifying the Interfaces

Purpose

Verify that the interfaces are configured properly.

Action

From configuration mode, enter the `show interfaces` command.

Verifying the Interface Queues

Purpose

Verify that the interface queues are configured properly.

Action

From configuration mode, enter the `show interfaces queue` command.

RELATED DOCUMENTATION

Strict-Priority Queue Overview | 169
Understanding Strict-Priority Queues | 170
Example: Configuring Priority Scheduling | 171

Example: Configuring CoS Non-Strict Priority Scheduling

IN THIS SECTION

- Requirements | 189

- Overview | 189
- Configuration | 190
- Verification | 193

On supported platforms, you can configure non-strict priority scheduling to avoid starvation of lower priority queues.

This example shows how to assign non-strict priority scheduling to CoS queues.

Requirements

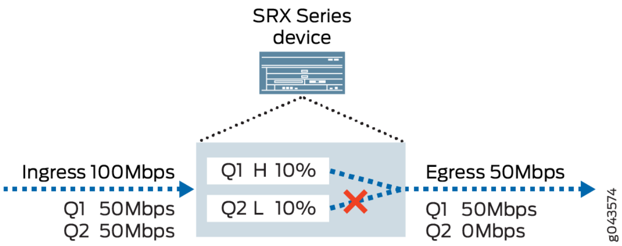
Before you begin, determine the shaping rate, schedulers, and forwarding classes for the CoS traffic. See *shaping-rate (CoS Interfaces)*, "Example: Configuring Class-of-Service Schedulers on a Security Device" on page 133, and "Example: Assigning Forwarding Classes to Output Queues" on page 86.

Overview

Traffic shaping bandwidth allocation is based on the egress (outgoing) interface that the packet traverses. If you have several traffic streams with CoS prioritized, all traffic streams across the network are sent with more bandwidth than the bandwidth on the egress interface. This can sometimes result in higher-priority queues getting all of the bandwidth and lower priority queues not getting any bandwidth, and thus being starved.

This example demonstrates how the non-strict priority feature can resolve the starvation of strict priority scheduling problem. For this scenario, you initialize two traffic streams (50 Mbps each) with CoS classifiers configured. Interface ge-0/0/1 is configured for ingress traffic, and ge-0/0/2 is configured for egress traffic with shaping enabled at 50 million. For traffic stream Q2, you set the queue priority as high and the shaping rate at 10%. For the other traffic stream Q1, you set the queue priority as low and the shaping rate at 10%. See Figure 6 on page 189.

Figure 6: CoS Traffic with HIgh and Low Priority Queues



NOTE: Since CoS is strict priority scheduling, please keep in mind that higher priority queues can starve lower priority queues.

Configuration

IN THIS SECTION

- Procedure | 190

Procedure

CLI Quick Configuration

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

```
set class-of-service interfaces ge-0/0/2 unit 0 shaping-rate 50m
set interfaces ge-0/0/2 per-unit-scheduler
set class-of-service interfaces ge-0/0/1 unit 0 classifiers dscp dscp_custom
set class-of-service classifiers dscp dscp_custom forwarding-class HIGH loss-priority low code-
points 100011
set class-of-service classifiers dscp dscp_custom forwarding-class LOW loss-priority low code-
points 100100
set class-of-service forwarding-classes queue 1 HIGH
set class-of-service forwarding-classes queue 0 LOW
set class-of-service scheduler-maps sched forwarding-class HIGH scheduler Q1
set class-of-service scheduler-maps sched forwarding-class LOW scheduler Q2
set class-of-service schedulers Q2 transmit-rate percent 10
set class-of-service schedulers Q2 priority high
set class-of-service schedulers Q1 transmit-rate percent 10
set class-of-service schedulers Q1 priority low
set-class-of-service non-strict-priority-scheduling
```

Step-by-Step Procedure

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 Junos OS CLI User Guide.

To configure non-strict priority scheduling:

1. Configure shaping rate of 50 Mbps on the egress interface.

```
[edit]
user@host# set class-of-service interfaces ge-0/0/2 unit 0 shaping-rate 50m
set interfaces ge-0/0/2 per-unit-scheduler
```

2. Configure classifiers on the ingress interface.

```
[edit]
user@host# set class-of-service interfaces ge-0/0/1 unit 0 classifiers dscp dscp_custom
```

3. Define the DSCP value to be assigned to the forwarding class.

```
[edit]
user@host# set class-of-service classifiers dscp dscp_custom forwarding-class HIGH loss-
priority low code-points 100011
user@host# set class-of-service classifiers dscp dscp_custom forwarding-class LOW loss-
priority low code-points 100100
```

4. Define the forwarding class to a queue number.

```
[edit]
user@host# set class-of-service forwarding-classes queue 1 HIGH
user@host# set class-of-service forwarding-classes queue 0 LOW
```

5. Map the forwarding classes to a scheduler to control prioritized queueing.

```
[edit]
user@host# set class-of-service scheduler-maps sched forwarding-class HIGH scheduler Q1
user@host# set class-of-service scheduler-maps sched forwarding-class LOW scheduler Q2
```

6. Define the schedulers with priority and transmit rates. The example uses the same ratio for transmit rate but defines different priorities.

```
[edit]
user@host# set class-of-service schedulers Q2 transmit-rate percent 10
user@host# set class-of-service schedulers Q2 priority high
user@host# set class-of-service schedulers Q1 transmit-rate percent 10
user@host# set class-of-service schedulers Q1 priority low
```

7. Configure the new non-strict-priority-scheduling option.

```
[edit]
user@host# set-class-of-service non-strict-priority-scheduling
```

Results

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

```
user@host> show interface queue ge-0/0/2
Queue: 0, Forwarding classes: LOW
  Queued:
    Packets              :             18085500              8571 pps
    Bytes                :          18013158000          68297136 bps
  Transmitted:
    Packets              :              3800910              2030 pps
    Bytes                :           3785706360          16178104 bps
    Tail-dropped packets :             14284525              6534 pps
Queue: 1, Forwarding classes: HIGH
  Queued:
    Packets              :             18085556              8541 pps
    Bytes                :          18013213776          68062256 bps
  Transmitted:
    Packets              :             11432620              6107 pps
    Bytes                :          11386889520          48660808 bps
    Tail-dropped packets :              6652859              2436 pps
```

You will notice that the LOW priority queue got some traffic.

NOTE: Traffic on the low priority queue is still less than the high priority queue, as the non-priority scheduling option still works to control traffic..

If you are done configuring the device, enter `commit` from configuration mode.

Verification

IN THIS SECTION

- Verifying Non-Strict Priority Configuration | 193

Verifying Non-Strict Priority Configuration

Purpose

Verify that non-strict priority scheduling is configured properly.

Action

From operational mode, enter the `show class-of-service` command.

RELATED DOCUMENTATION

non-strict-priority-scheduling

Example: Configuring and Applying Scheduler Maps | 149

Transmission Scheduling Overview | 124

CHAPTER 8

Controlling Congestion with Drop Profiles

IN THIS CHAPTER

- RED Drop Profiles Overview | 194
- RED Drop Profiles and Congestion Control | 196
- Configuring RED Drop Profiles | 198
- Example: Configuring RED Drop Profiles | 201
- Example: Configuring Segmented and Interpolated Style Profiles | 205

RED Drop Profiles Overview

IN THIS SECTION

- Default Drop Profiles | 195

A drop profile is a feature of the random early detection (RED) process that allows packets to be dropped before queues are full. Drop profiles are composed of two main values—the queue fullness and the drop probability. The queue fullness represents percentage of memory used to store packets in relation to the total amount that has been allocated for that queue. The drop probability is a percentage value that correlates to the likelihood that an individual packet is dropped from the network. These two variables are combined in a graph-like format.

A random number between 0 and 100 is calculated for each packet. This random number is plotted against the drop profile having the current queue fullness of that particular queue. When the random number falls above the graph line, the packet is transmitted onto the physical media. When the number falls below the graph line, the packet is dropped from the network.

Randomly dropped packets are counted as RED-dropped, while packets dropped for other reasons (100% probability) are counted as tail-dropped.

You specify drop probabilities in the drop profile section of the class-of-service (CoS) configuration hierarchy and reference them in each scheduler configuration. For each scheduler, you can configure multiple separate drop profiles, one for each combination of loss priority (low, medium-low, medium-high, or high) and IP transport protocol (TCP or non-TCP or any).

NOTE: For some SRX Series Firewalls, tcp and non-tcp values are not supported; only the value “any” is supported. Actual platform support depends on the Junos OS release in your implementation.

You can configure a maximum of 32 different drop profiles.

To configure RED drop profiles, include the following statements at the `[edit class-of-service]` hierarchy level of the configuration:

```
[edit class-of-service]
drop-profiles {
    profile-name {
        fill-level percentage drop-probability percentage;
        interpolate {
            drop-probability [ values ];
            fill-level [ values ];
        }
    }
}
```

Default Drop Profiles

By default, if you configure no drop profiles, RED is still in effect and functions as the primary mechanism for managing congestion. In the default RED 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.

RELATED DOCUMENTATION

Example: Configuring RED Drop Profiles | **201**

RED Drop Profiles and Congestion Control

If the device must support assured forwarding, you can control congestion by configuring random early detection (RED) drop profiles. RED drop profiles use drop probabilities for different levels of buffer fullness to determine which scheduling queue on the device is likely to drop assured forwarding packets under congested conditions. The device can drop packets when the queue buffer becomes filled to the configured percentage.

Assured forwarding traffic with the PLP (packet loss priority) bit set is more likely to be discarded than traffic without the PLP bit set. This example shows how to configure a drop probability and a queue fill level for both PLP and non-PLP assured forwarding traffic. It is only one example of how to use RED drop profiles.

The example shows how to configure the RED drop profiles listed in Table 38 on page 196.

Table 38: Sample RED Drop Profiles

Drop Profile	Drop Probability	Queue Fill Level
`af-normal`—For non-PLP (normal) assured forwarding traffic	Between 0 (never dropped) and 100 percent (always dropped)	Between 95 and 100 percent
`af-with-plp`—For PLP (aggressive packet dropping) assured forwarding traffic	Between 95 and 100 percent (always dropped)	Between 80 and 95 percent

To configure RED drop profiles for assured forwarding congestion control on the device:

1. Navigate to the top of the configuration hierarchy in either the J-Web or CLI configuration editor.
2. Perform the configuration tasks described in Table 39 on page 197.
3. If you are finished configuring the device, commit the configuration.
4. Go on to one of the following tasks:
    - To assign resources, priorities, and profiles to output queues, see "Example: Configuring Class-of-Service Schedulers on a Security Device" on page 133.
    - To apply rules to logical interfaces, see "Example: Configuring Virtual Channels" on page 230.

Table 39: Configuring RED Drop Profiles for Assured Forwarding Congestion Control

Task	CLI Configuration Editor
Navigate to the `Class of service` level in the configuration hierarchy.	From the `[edit]` hierarchy level, enter `edit class-of-service`

Configure the lower drop probability for normal, non-PLP traffic.	Enter
	`edit drop-profiles af-normal interpolate`
	`set drop-probability 0`
	`set drop-probability 100`

Configure a queue fill level for the lower non-PLP drop probability.	Enter
	`set fill-level 95`
	`set fill-level 100`

Configure the higher drop probability for PLP traffic.	From the `[edit class of service]` hierarchy level, enter
	`edit drop-profiles af-with-PLP interpolate`
	`set drop-probability 95`
	`set drop-probability 100`

Configure a queue fill level for the higher PLP drop probability.	Enter
	`set fill-level 80`
	`set fill-level 95`

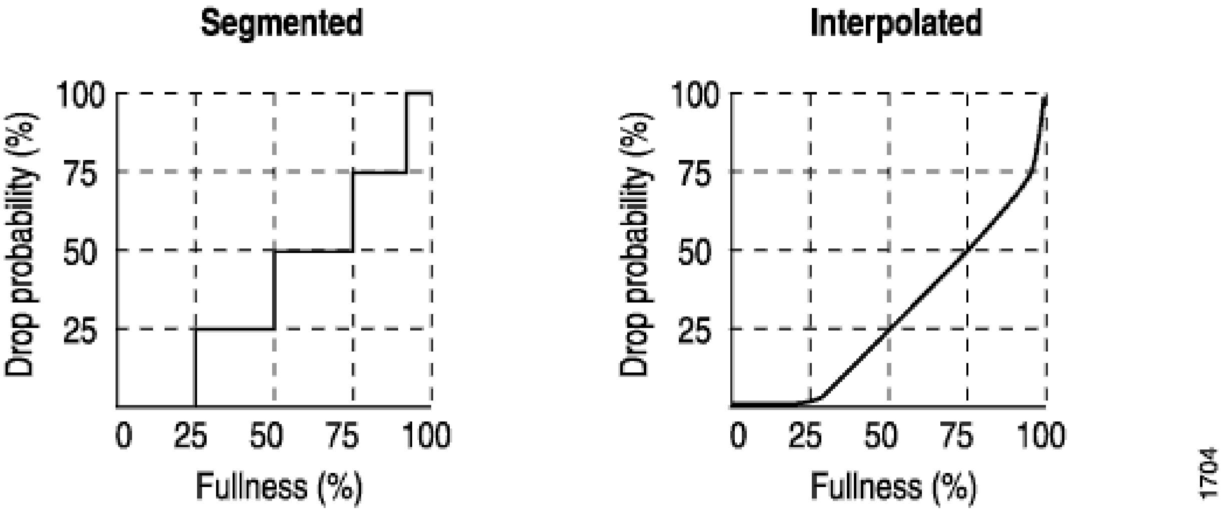
RELATED DOCUMENTATION

Example: Configuring RED Drop Profiles | 201

Configuring RED Drop Profiles

Create a segmented configuration and an interpolated configuration that correspond to the graphs in Figure 7 on page 199. The values defined in the configuration are matched to represent the data points in the graph line. In this example, the drop probability is 25 percent when the queue is 50 percent full. The drop probability increases to 50 percent when the queue is 75 percent full.

Figure 7: Segmented and Interpolated Drop Profiles



Segmented

```
class-of-service {
    drop-profiles {
        segmented-style-profile {
            fill-level 25 drop-probability 25;
            fill-level 50 drop-probability 50;
            fill-level 75 drop-probability 75;
            fill-level 95 drop-probability 100;
        }
    }
}
```

To create the profile's graph line, the software begins at the bottom-left corner, representing a 0 percent fill level and a 0 percent drop probability. This configuration draws a line directly to the right until it reaches the first defined fill level, 25 percent for this configuration. The software then continues the line vertically until the first drop probability is reached. This process is repeated for all of the defined levels and probabilities until the top-right corner of the graph is reached.

Create a smoother graph line by configuring the profile with the `interpolate` statement. This allows the software to automatically generate 64 data points on the graph beginning at (0, 0) and ending at (100, 100). Along the way, the graph line intersects specific data points, which you define as follows:

Interpolated

```
class-of-service {
    drop-profiles {
        interpolated-style-profile {
            interpolate {
                fill-level [ 50 75 ];
                drop-probability [ 25 50 ];
            }
        }
    }
}
```

RELATED DOCUMENTATION

Understanding RED Drop Profiles

Example: Configuring RED Drop Profiles

IN THIS SECTION

- Requirements | 201
- Overview | 201
- Configuration | 202
- Verification | 204

This example shows how to configure RED drop profiles.

Requirements

Before you begin, determine which type of profile you want to configure. See "Example: Configuring Segmented and Interpolated Style Profiles" on page 205.

Overview

A drop profile is a feature of the RED process that allows packets to be dropped before queues are full. Drop profiles are composed of two main values the queue fullness and the drop probability.

You can control congestion by configuring RED drop profiles, if the device supports assured forwarding. RED drop profiles use drop probabilities for different levels of buffer fullness to determine which scheduling queue on the device is likely to drop assured forwarding packets under congested conditions. The device can drop packets when the queue buffer becomes filled to the configured percentage. Assured forwarding traffic with the PLP bit set is more likely to be discarded than traffic without the PLP bit set.

In this example, you configure a drop probability and a queue fill level for both PLP and non-PLP assured forwarding traffic.

Table 40 on page 201 shows how to configure the RED drop profiles listed.

Table 40: Sample RED Drop Profiles

Drop Profile	Drop Probability	Queue Fill Level
`af-normal`—For non-PLP (normal) assured forwarding traffic	Between 0 (never dropped) and 100 percent (always dropped)	Between 95 and 100 percent

Table 40: Sample RED Drop Profiles *(Continued)*

Drop Profile	Drop Probability	Queue Fill Level
`af-with-plp`—For PLP (aggressive packet dropping) assured forwarding traffic	Between 95 and 100 percent (always dropped)	Between 80 and 95 percent

Configuration

IN THIS SECTION

- Procedure | 202

Procedure

CLI Quick Configuration

To quickly configure RED drop profiles, copy the following commands, paste them into a text file, remove any line breaks, change any details necessary to match your network configuration, copy and paste the commands into the CLI at the `[edit]` hierarchy level, and then enter `commit` from configuration mode.

```
[edit]
set class-of-service drop-profiles af-normal interpolate drop-probability 0
set class-of-service drop-profiles af-normal interpolate drop-probability 100
set class-of-service drop-profiles af-normal interpolate fill-level 95
set class-of-service drop-profiles af-normal interpolate fill-level 100
set class-of-service drop-profiles af-with-PLP interpolate drop-probability 95
set class-of-service drop-profiles af-with-PLP interpolate drop-probability 100
set class-of-service drop-profiles af-with-PLP interpolate fill-level 80
set class-of-service drop-profiles af-with-PLP interpolate fill-level 95
```

Step-by-Step Procedure

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*.

To configure RED drop profiles:

1. Configure the lower drop probability for normal, non-PLP traffic.

```
[edit]
user@host# edit class-of-service
user@host# edit drop-profiles af-normal interpolate
user@host# set drop-probability 0
user@host# set drop-probability 100
```

2. Configure a queue fill level for the lower non-PLP drop probability.

```
 [edit class-of-service drop-profiles af-normal interpolate]
user@host# set fill-level 95
user@host# set fill-level 100
```

3. Configure the higher drop probability for PLP traffic.

```
[edit]
user@host# edit class-of-service
user@host# edit drop-profiles af-with-PLP interpolate
user@host# set drop-probability 95
user@host# set drop-probability 100
```

4. Configure a queue fill level for the higher PLP drop probability.

```
[edit class-of-service drop-profiles af-with-PLP interpolate]
user@host# set fill-level 80
user@host# set fill-level 95
```

Results

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

```
user@host# show class-of-service
    drop-profiles {
```

```
    af-normal {
        interpolate {
            fill-level [ 95 100 ];
            drop-probability [ 0 100 ];
        }
    }
    af-with-PLP {
        interpolate {
            fill-level [ 80 95 ];
            drop-probability [ 95 100 ];
        }
    }
}
```

If you are done configuring the device, enter `commit` from configuration mode.

Verification

IN THIS SECTION

- Verifying RED Drop Profiles Configuration | 204

Verifying RED Drop Profiles Configuration

Purpose

Verify that the RED drop profiles are configured properly.

Action

From operational mode, enter the `show class-of-service` command.

RELATED DOCUMENTATION

RED Drop Profiles Overview | 194

Understanding RED Drop Profiles

Example: Configuring Segmented and Interpolated Style Profiles

IN THIS SECTION

- Requirements | 205
- Overview | 205
- Configuration | 205
- Verification | 209

This example shows how to configure segmented and interpolated style profiles.

Requirements

No special configuration beyond device initialization is required before configuring this feature.

Overview

In this example, you configure the segmented style profile by setting the drop probability to 25 percent when the queue is 25 percent full. The drop probability increases to 50 percent when the queue is 50 percent full. You set the drop probability to 75 percent when the queue is 75 percent full and finally the drop probability is set to 95 percent when the queue is 100 percent full.

Then you configure the interpolated style profile and set the fill level to 50 percent and 75 percent. Finally you set the drop probability to 25 percent and later to 50 percent.

Configuration

IN THIS SECTION

- Configuring Segmented Style Profiles | 206
- Configuring Interpolated Style Profiles | 207

Configuring Segmented Style Profiles

CLI Quick Configuration

To quickly configure segmented style profiles, copy the following commands and paste them into the CLI:

```
[edit]
set class-of-service drop-profiles segmented-style-profile fill-level 25 drop-probability 25
set class-of-service drop-profiles segmented-style-profile fill-level 50 drop-probability 50
set class-of-service drop-profiles segmented-style-profile fill-level 75 drop-probability 75
set class-of-service drop-profiles segmented-style-profile fill-level 95 drop-probability 100
```

Step-by-Step Procedure

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*.

To configure segmented style profiles:

1. Configure class of service.

   ```
   [edit]
   user@host# edit class-of-service
   ```

2. Configure segmented style profile.

   ```
   [edit class-of-service]
   user@host# edit drop-profiles segmented-style-profile
   ```

3. Specify fill levels and drop probabilities.

   ```
   [edit class-of-service drop-profiles segmented-style-profile]
   user@host# set fill-level 25 drop-probability 25
   user@host# set fill-level 50 drop-probability 50
   user@host# set fill-level 75 drop-probability 75
   user@host# set fill-level 95 drop-probability 100
   ```

Results

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

```
[edit]
user@host# show class-of-service
    drop-profiles {
    segmented-style-profile {
        fill-level 25 drop-probability 25;
        fill-level 50 drop-probability 50;
        fill-level 75 drop-probability 75;
        fill-level 95 drop-probability 100;
    }
}
```

If you are done configuring the device, enter **commit** from configuration mode.

Configuring Interpolated Style Profiles

CLI Quick Configuration

To quickly configure interpolated style profiles, copy the following commands and paste them into the CLI:

```
[edit]
set class-of-service drop-profiles interpolated-style-profile interpolate fill-level 50
set class-of-service drop-profiles interpolated-style-profile interpolate fill-level 75
set class-of-service drop-profiles interpolated-style-profile interpolate drop-probability 25
set class-of-service drop-profiles interpolated-style-profile interpolate drop-probability 50
```

Step-by-Step Procedure

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*.

To configure interpolated style profile:

1. Configure class of service.

```
[edit]
user@host# edit class-of-service
```

2. Configure interpolated style profile.

```
[edit class-of-service]
user@host# edit drop-profiles interpolated-style-profile interpolate
```

3. Specify fill levels.

```
[edit class-of-service drop-profiles interpolated-style-profile interpolate]
user@host# set fill-level 50
user@host# set fill-level 75
```

4. Specify drop probabilities.

```
[edit class-of-service drop-profiles interpolated-style-profile interpolate]
user@host# set drop-probability 25
user@host# set drop-probability 50
```

Results

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

```
[edit]
user@host# show class-of-service
    drop-profiles {
    interpolated-style-profile {
        fill-level [ 50 75 ];
        drop-probability [ 25 50 ];
    }
    }
}
```

If you are done configuring the device, enter **commit** from configuration mode.

Verification

IN THIS SECTION

- Verifying Segmented Style Profile Configuration | 209
- Verifying Interpolated Style Profile Configuration | 209

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

Verifying Segmented Style Profile Configuration

Purpose

Verify that the segmented style profile is configured properly.

Action

From configuration mode, enter the **show class-of-service** command.

Verifying Interpolated Style Profile Configuration

Purpose

Verify that the interpolated style profile is configured properly.

Action

From configuration mode, enter the **show class-of-service** command.

RELATED DOCUMENTATION

RED Drop Profiles Overview | 194

Understanding RED Drop Profiles

Example: Configuring RED Drop Profiles | 201

CHAPTER 9

Controlling Congestion with Explicit Congestion Notification

IN THIS CHAPTER

- CoS Explicit Congestion Notification (ECN) | 210

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 | 210
- How ECN Works | 211
- WRED Drop Profile Control of ECN Thresholds | 216
- Dynamic ECN | 219
- ECN Packets per Queue | 220
- On-Chip Buffer with ECN on PTX10002-36QDD Routers | 222
- Platform-Specific Behavior | 223

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.

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.

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

IN THIS SECTION

- ECN Bits in the DiffServ Field | **212**
- End-to-End ECN Behavior | **213**
- End-to-end ECN Behavior | **214**
- ECN Compared to PFC and Ethernet PAUSE | **216**

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 41: 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

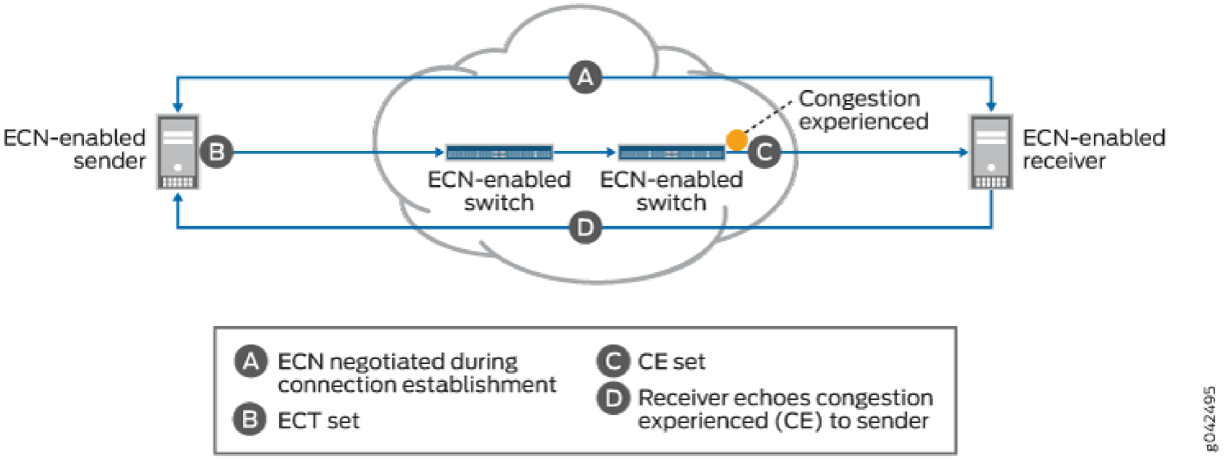
After the sending and receiving endpoints negotiate ECN, the sending endpoint marks packets as ECN-capable by setting the DiffServ ECN field to ECT(1) (01) or ECT(0) (10). Every intermediate device between the endpoints must have ECN enabled or it does not work.

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 216 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 8: Explicit Congestion Notification


A
Congestion experienced
ECN-enabled sender
B
ECN-enabled switch
ECN-enabled switch
C
ECN-enabled receiver
D
A ECN negotiated during connection establishment
B ECT set
C CE set
D Receiver echoes congestion experienced (CE) to sender
g042495

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.

   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 42: 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 | **218**
- How to Apply a WRED Drop Profile to an Output Queue: Port Scheduling | **218**

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 | **219**

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

> **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 | 220
- Configuration | 221

Overview

IN THIS SECTION

- Benefits | 221

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.

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.

RELATED DOCUMENTATION

Example: Configuring Static and Dynamic ECN

CHAPTER 10

Limiting Traffic Using Virtual Channels

IN THIS CHAPTER

- Virtual Channels Overview | 227
- Understanding Virtual Channels | 228
- Example: Configuring Virtual Channels | 230

Virtual Channels Overview

You can configure virtual channels to limit traffic sent from a corporate headquarters to its branch offices. Virtual channels might be required when the headquarters site has an expected aggregate bandwidth higher than that of the individual branch offices. The headquarters router must limit the traffic sent to each branch office router to avoid oversubscribing their links. For instance, if branch 1 has a 1.5 Mbps link and the headquarters router attempts to send 6 Mbps to branch 1, all of the traffic in excess of 1.5-Mbps is dropped in the ISP network.

You configure virtual channels on a *logical interface*. Each virtual channel has a set of eight queues with a scheduler and an optional shaper. You can use an output *firewall filter* to direct traffic to a particular virtual channel. For example, a filter can direct all traffic with a destination address for branch office 1 to virtual channel 1, and all traffic with a destination address for branch office 2 to virtual channel 2.

Although a virtual channel group is assigned to a logical interface, a virtual channel is quite different from a logical interface. The only features supported on a virtual channel are queuing, packet scheduling, and accounting. *Rewrite rules* and routing protocols apply to the entire logical interface.

When you configure virtual channels on an interface, the virtual channel group uses the same scheduler and shaper you configure at the `[edit interfaces` *`interface-name`* `unit` *`logical-unit-number`*`]` hierarchy level. In this way, virtual channels are an extension of regular scheduling and shaping and are not independent entities.

RELATED DOCUMENTATION

Understanding Virtual Channels | 228

Example: Configuring Virtual Channels on a Security Device

Understanding Virtual Channels

You configure a virtual channel to set up queuing, packet scheduling, and accounting rules to be applied to one or more logical interfaces. You must apply then the virtual channel to a particular *logical interface*.

You also create a list of virtual channels that you can assign to a virtual channel group. To define a virtual channel group that you can assign to a logical interface, include the `virtual-channel-groups` statement at the `[edit class-of-service]` hierarchy level.

The *virtual-channel-group-name* can be any name that you want. The *virtual-channel-name* must be one of the names that you define at the `[edit class-of-service virtual-channels]` hierarchy level. You can include multiple virtual channel names in a group.

The scheduler map is required. The *map-name* must be one of the scheduler maps that you configure at the `[edit class-of-service scheduler-maps]` hierarchy level. For more information, see "Example: Configuring Class-of-Service Schedulers on a Security Device" on page 133.

The shaping rate is optional. If you configure the shaping rate as a percentage, when the virtual channel is applied to a logical interface, the shaping rate is set to the specified percentage of the interface bandwidth. If you configure a shaper on a virtual channel, the shaper limits the maximum bandwidth transmitted by that virtual channel. Virtual channels without a shaper can use the full logical interface bandwidth. If there are multiple unshaped virtual channels, they share the available logical interface bandwidth equally.

When you apply the virtual channel group to a logical interface, a set of eight queues is created for each of the virtual channels in the group. The `scheduler-map` statement applies a scheduler to these queues. If you include the `shaping-rate` statement, a shaper is applied to the entire virtual channel.

You must configure one of the virtual channels in the group to be the default channel. Therefore, the `default` statement is required in the configuration of one virtual channel per channel group. Any traffic not explicitly directed to a particular channel is transmitted by this default virtual channel.

For the corresponding physical interface, you must also include the `per-unit-scheduler` statement at the `[edit interfaces interface-name]` hierarchy level as follows:

```
[edit interfaces interface-name]
per-unit-scheduler;
```

The `per-unit-scheduler` statement enables one set of output queues for each logical interface configured under the physical interface.

When you apply a virtual channel group to a logical interface, the software creates a set of eight queues for each of the virtual channels in the group.

If you apply a virtual channel group to multiple logical interfaces, the software creates a set of eight queues on each logical interface. The virtual channel names listed in the group are used on all the logical interfaces. We recommend specifying the scheduler and shaping rates in the virtual channel configuration in terms of percentages, rather than absolute rates. This allows you to apply the same virtual channel group to logical interfaces that have different bandwidths.

When you apply a virtual channel group to a logical interface, you cannot include the `scheduler-map` and `shaping-rate` statements at the [edit class-of-service interfaces *interface-name* unit *logical-unit-number*] hierarchy level. In other words, you can configure a scheduler map and a shaping rate on a logical interface, or you can configure virtual channels on the logical interface, but not both.

If you configure multiple logical interfaces on a single physical interface, each logical interface is guaranteed an equal fraction of the physical interface bandwidth as follows:

```
logical-interface-bandwidth =
physical-interface-bandwidth / number-of-logical-interfaces
```

If one or more logical interfaces do not completely use their allocation, the other logical interfaces share the excess bandwidth equally.

If you configure multiple virtual channels on a logical interface, they are each guaranteed an equal fraction of the logical interface bandwidth as follows:

```
virtual-channel-bandwidth =
logical-interface-bandwidth / number-of-virtual-channels
```

If you configure a shaper on a virtual channel, the shaper limits the maximum bandwidth transmitted by that virtual channel. Virtual channels without a shaper can use the full logical interface bandwidth. If there are multiple unshaped virtual channels, they share the available logical interface bandwidth equally.

RELATED DOCUMENTATION

Virtual Channels Overview | 227

Example: Configuring Virtual Channels on a Security Device

Example: Configuring Virtual Channels

IN THIS SECTION

- Requirements | 230
- Overview | 230
- Configuration | 230
- Verification | 236

This example shows how to create virtual channels between a headquarters and its branch office.

Requirements

Before you begin, ensure that your headquarters and branch office have a network connection where the expected aggregate bandwidth is higher for your headquarters than for your branch office. The devices at your headquarters will then be set up to limit the traffic sent to the branch office to avoid oversubscribing the link.

Overview

In this example, you create the virtual channels as branch1–vc, branch2–vc, branch3–vc, and default-vc. You then define the virtual channel group as wan-vc-group to include the four virtual channels and assign the scheduler map as bestscheduler to each virtual channel. Three of the virtual channels are shaped to 1.5 Mbps. The fourth virtual channel is default-vc, and it is not shaped so it can use the full interface bandwidth.

Then you apply them in the firewall filter as choose-vc to the device's interface t3-1/0/0. The output filter on the interface sends all traffic with a destination address matching 192.168.10.0/24 to branch1-vc, and similar configurations are set for branch2-vc and branch3-vc. Traffic not matching any of the addresses goes to the default, unshaped virtual channel.

Configuration

IN THIS SECTION

- Procedure | 231

Procedure

CLI Quick Configuration

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

```
 set class-of-service virtual-channels branch1-vc
set class-of-service virtual-channels branch2-vc
set class-of-service virtual-channels branch3-vc
set class-of-service virtual-channels default-vc
set class-of-service virtual-channel-groups wan-vc-group branch1-vc scheduler-map bestscheduler
set class-of-service virtual-channel-groups wan-vc-group branch2-vc scheduler-map bestscheduler
set class-of-service virtual-channel-groups wan-vc-group branch3-vc scheduler-map bestscheduler
set class-of-service virtual-channel-groups wan-vc-group default-vc scheduler-map bestscheduler
set class-of-service virtual-channel-groups wan-vc-group default-vc default
set class-of-service virtual-channel-groups wan-vc-group branch1-vc shaping-rate 1500000
set class-of-service virtual-channel-groups wan-vc-group branch2-vc shaping-rate 1500000
set class-of-service virtual-channel-groups wan-vc-group branch3-vc shaping-rate 1500000
set class-of-service interfaces t3-1/0/0 unit 0 virtual-channel-group wan-vc-group
set firewall family inet filter choose-vc term branch1 from destination-address 192.168.10.0/24
set firewall family inet filter choose-vc term branch1 then virtual-channel branch1-vc
set firewall family inet filter choose-vc term branch1 then accept
set firewall family inet filter choose-vc term branch2 from destination-address 192.168.20.0/24
set firewall family inet filter choose-vc term branch2 then virtual-channel branch2-vc
set firewall family inet filter choose-vc term branch2 then accept
set firewall family inet filter choose-vc term branch3 from destination-address 192.168.30.0/24
set firewall family inet filter choose-vc term branch3 then virtual-channel branch3-vc
set firewall family inet filter choose-vc term branch3 then accept
set firewall family inet filter choose-vc term default then virtual-channel default-vc
set firewall family inet filter choose-vc term default then accept
 set interfaces t3-1/0/0 unit 0  family inet filter output choose-vc
```

Step-by-Step Procedure

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 Junos OS CLI User Guide.

To configure virtual channels:

1. Define the virtual channels and the default virtual channel.

```
[edit]
user@host# edit class-of-service
user@host# set virtual-channels branch1-vc
user@host# set virtual-channels branch2-vc
user@host# set virtual-channels branch3-vc
user@host# set virtual-channels default-vc
```

2. Define the virtual channel group and assign each virtual channel a scheduler map.

```
[edit class-of-service]
user@host# set virtual-channel-groups wan-vc-group branch1-vc scheduler-map bestscheduler
user@host# set virtual-channel-groups wan-vc-group branch2-vc scheduler-map bestscheduler
user@host# set virtual-channel-groups wan-vc-group branch3-vc scheduler-map bestscheduler
user@host# set virtual-channel-groups wan-vc-group default-vc scheduler-map bestscheduler
user@host# set virtual-channel-groups wan-vc-group default-vc default
```

3. Specify a shaping rate.

```
[edit class-of-service]
user@host# set virtual-channel-groups wan-vc-group branch1-vc shaping-rate 1.5m
user@host# set virtual-channel-groups wan-vc-group branch2-vc shaping-rate 1.5m
user@host# set virtual-channel-groups wan-vc-group branch3-vc shaping-rate 1.5m
```

4. Apply the virtual channel group to the logical interface.

```
[edit class-of-service]
user@host# set interfaces t3-1/0/0 unit 0 virtual-channel-group wan-vc-group
```

5. Create the firewall filter to select the traffic.

```
[edit firewall]
user@host# set firewall family inet filter choose-vc term branch1 from destination-address
192.168.10.0/24
user@host# set firewall family inet filter choose-vc term branch1 then virtual-channel
branch1-vc
user@host# set firewall family inet filter choose-vc term branch1 then accept
```

```
user@host# set firewall family inet filter choose-vc term branch2 from destination-address
192.168.20.0/24
user@host# set firewall family inet filter choose-vc term branch2 then virtual-channel
branch2-vc
user@host# set firewall family inet filter choose-vc term branch2 then accept
user@host# set firewall family inet filter choose-vc term branch3 from destination-address
192.168.30.0/24
user@host# set firewall family inet filter choose-vc term branch3 then virtual-channel
branch3-vc
user@host# set firewall family inet filter choose-vc term branch3 then accept
user@host# set firewall family inet filter choose-vc term default then virtual-channel
default-vc
user@host# set firewall family inet filter choose-vc term default then accept
```

6. Apply the firewall filter to output traffic.

```
[edit interfaces]
user@host# set t3-1/0/0 unit 0 family inet filter output choose-vc
```

Results

From configuration mode, confirm your configuration by entering the `show class-of-service`, `show firewall`, and `show interfaces t3-1/0/0` commands. If the output does not display the intended configuration, repeat the configuration instructions in this example to correct it.

```
user@host# show class-of-service
virtual-channels {
    branch1-vc;
    branch2-vc;
    branch3-vc;
    default-vc;
}
virtual-channel-groups {
    wan-vc-group {
        branch1-vc {
        scheduler-map bestscheduler;
        shaping-rate 1500000;
    }
    branch2-vc {
        scheduler-map bestscheduler;
        shaping-rate 1500000;
    }
    branch3-vc {
        scheduler-map bestscheduler;
        shaping-rate 1500000;
    }
    default-vc {
        scheduler-map bestscheduler;
        default;
    }
}
}
interfaces {
    t3-1/0/0 {
        unit 0 {
            virtual-channel-group wan-vc-group;
        }
    }
}
[edit]
user@host# show firewall
family inet {
    filter choose-vc {
        term branch1 {
            from {
                destination-address {
                    192.168.10.0/24;
                }
            }
            then {
                virtual-channel branch1-vc;
                accept;
            }
        }
        term branch2 {
            from {
                destination-address {
                    192.168.20.0/24;
                }
            }
            then {
                virtual-channel branch2-vc;
                accept;
            }
        }
        term branch3 {
            from {
                destination-address {
                    192.168.30.0/24;
                }
            }
            then {
                virtual-channel branch1-vc;
                accept;
            }
        }
        term branch2 {
            from {
                destination-address {
                    192.168.20.0/24;
                }
            }
            then {
                virtual-channel branch2-vc;
                accept;
            }
        }
        term branch3 {
            from {
                destination-address {
                    192.168.30.0/24;
                }
            }
            then {
                virtual-channel branch3-vc;
                accept;
            }
        }
        term default {
            then {
                virtual-channel default-vc;
                accept;
            }
        }
    }
}
```

```
[edit]
user@host# show interfaces t3-1/0/0
unit 0 {
    family inet {
        filter {
            output choose-vc;
        }
    }
}
```

If you are done configuring the device, enter `commit` from configuration mode.

Verification

IN THIS SECTION

- Verifying Virtual Channel Configuration | 236

Verifying Virtual Channel Configuration

Purpose

Verify that the virtual channels are properly configured.

Action

From configuration mode, enter the `show class-of-service`, `show firewall`, and `show interfaces t3-1/0/0` commands.

RELATED DOCUMENTATION

Virtual Channels Overview | 227

Understanding Virtual Channels | 228

CHAPTER 11

Enabling Queuing for Tunnel Interfaces

IN THIS CHAPTER

- CoS Queuing for Tunnels Overview | 237
- Understanding the ToS Value of a Tunnel Packet | 242
- Example: Configuring CoS Queuing for GRE or IP-IP Tunnels | 243
- Copy Outer IP Header DSCP and ECN to Inner IP Header | 250
- Understanding CoS Support on st0 Interfaces | 251

CoS Queuing for Tunnels Overview

IN THIS SECTION

- Benefits of CoS Queuing for Tunnel Interfaces | 238
- Configuring CoS on Logical Tunnels | 238
- How CoS Queuing Works | 241
- Limitations on CoS Shapers for Tunnel Interfaces | 241

On an SRX Series Firewall running Junos OS, a tunnel interface is an internal interface and supports many of the same CoS features as a physical interface. The tunnel interface creates a virtual point-to-point link between two SRX Series Firewalls at remote points over an IP network.

For example, you can configure CoS features for generic routing encapsulation (GRE) and IP-IP tunnel interfaces. Tunneling protocols encapsulate packets inside a transport protocol.

GRE and IP-IP tunnels are used with services like IPsec and NAT to set up point-to-point VPNs. Junos OS allows you to enable CoS queuing, scheduling, and shaping for traffic exiting through these tunnel interfaces. For an example of configuring CoS Queuing for GRE tunnels, see "Example: Configuring CoS Queuing for GRE or IP-IP Tunnels" on page 243.

NOTE: CoS queuing is not supported on GRE tunnels in chassis clusters.

Benefits of CoS Queuing for Tunnel Interfaces

CoS queuing enabled for tunnel interfaces has the following benefits:

- Segregates tunnel traffic.

  Each tunnel can be shaped so that a tunnel with low-priority traffic cannot flood other tunnels that carry high-priority traffic.

  Traffic for one tunnel does not impact traffic on other tunnels.

- Controls tunnel bandwidth.

  Traffic through various tunnels is limited to not exceed a certain bandwidth.

  For example, suppose you have three tunnels to three remote sites through a single WAN interface. You can select CoS parameters for each tunnel such that traffic to some sites gets more bandwidth than traffic to other sites.

- Customizes CoS policies.

  You can apply different queuing, scheduling, and shaping policies to different tunnels based on user requirements. Each tunnel can be configured with different scheduler maps, different queue depths, and so on. Customization allows you to configure granular CoS policy providing for better control over tunnel traffic.

- Prioritizes traffic before it enters a tunnel.

  For example, CoS queuing avoids having low-priority packets scheduled ahead of high-priority packets when the interface speed is higher than the tunnel traffic speed. This feature is most useful when combined with IPsec. Typically, IPsec processes packets in a FIFO manner. However, with CoS queuing each tunnel can prioritize high-priority packets over low-priority packets. Also, each tunnel can be shaped so that a tunnel with low-priority traffic does not flood tunnels carrying high-priority traffic.

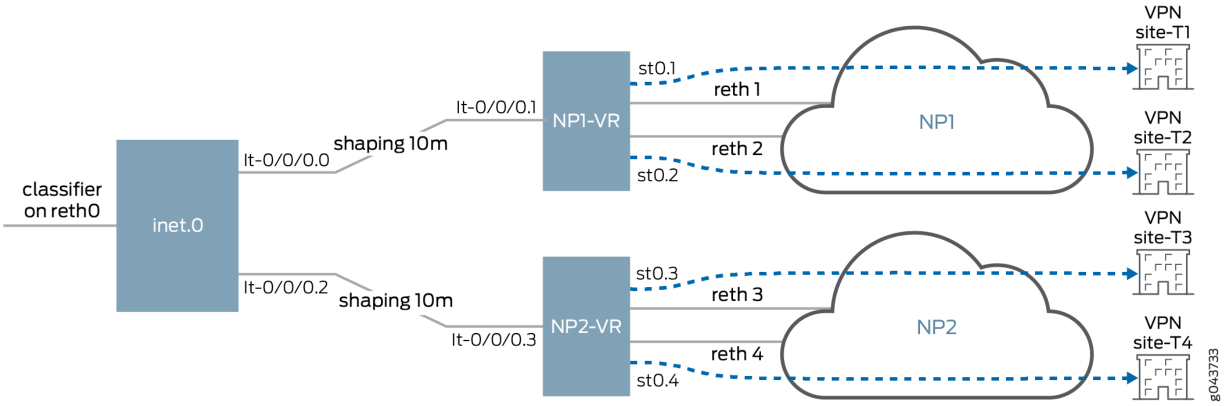
Configuring CoS on Logical Tunnels

CoS has four typical scenarios that allow connection with remote sites using secure tunnels. However different secure tunnels may connect using different reth interfaces to different Network Providers (NP). For a specific NP, limited uplink bandwidth may be used to prioritize high-priority business and to avoid blindly dropping traffic at the NP side. Currently CoS does not support queuing across physical interfaes (IFD). Having a shared policer does not work as well as queuing, the policer may drop high-priority traffic regardless of priority. To support queuing on an IFD to enable CoS features to prioritize queuing and shaping requires logical tunnel (LT) and NP configuration.

You must define a pair of logical tunnels that are one-to-one mapped to NPs and redirect traffic with routing to the LT interface before encrypting the traffic through a secure tunnel.

For example, configure lt-0/0/0.0 and lt-0/0/0.1 to connect inet 0 and NP1 (virtual router) and configure a static route to redirect traffic to NP1 as lt-0/0/0.0 next-hop. Because NP1 has 10mbps bandwidth for upstream traffic, lt-0/00.0 can be configured with 10mbps of bandwidth shaping. See Figure 9 on page 239.

Figure 9: CoS Solutions Using Logical Tunnels


classifier
on reth0
inet.0
lt-0/0/0.0
shaping 10m
lt-0/0/0.1
NP1-VR
st0.1
reth 1
reth 2
st0.2
NP1
VPN
site-T1
VPN
site-T2
lt-0/0/0.2
shaping 10m
lt-0/0/0.3
NP2-VR
st0.3
reth 3
reth 4
st0.4
NP2
VPN
site-T3
VPN
site-T4
g043733

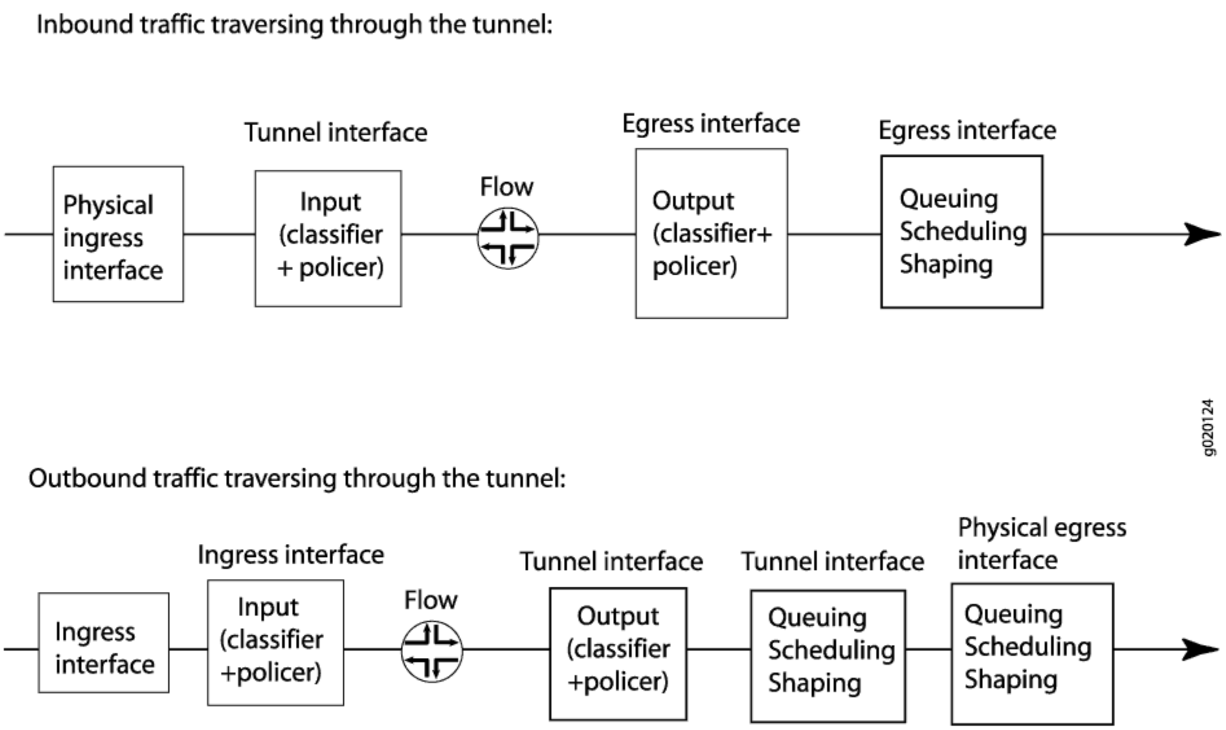
```
routing-instances {
    NP1 {
        instance-type virtual-router;
        interface lt-0/0/0.1;
        interface lo0.0;
        interface reth1.0;
        interface reth2.0;
        interface st0.1;
        interface st0.2;
        routing-options {
            static {
                route 59.200.200.1/32 next-hop <next-hop addr of ipsec tunnel st0.1>;
                route 59.200.200.2/32 next-hop <next-hop addr of ipsec tunnel st0.2>;
                route 60.60.60.1/32 next-hop st0.1;
                route 60.60.60.2/32 next-hop st0.2;
                route 58.58.58.1/32 next-hop lt-0/0/0.1;
                route 58.58.58.2/32 next-hop lt-0/0/0.1;
            }
        }
    }
    NP2 {
        instance-type virtual-router;
        interface lt-0/0/0.3;
        interface lo0.1;
        interface reth3.0;
        interface reth4.0;
        interface st0.3;
        interface st0.4;
        routing-options {
            static {
                route 59.200.200.3/32 next-hop <next-hop addr of ipsec tunnel st0.3>;
                route 59.200.200.4/32 next-hop <next-hop addr of ipsec tunnel st0.4>;
                route 60.60.60.3/32 next-hop st0.3;
                route 60.60.60.4/32 next-hop st0.4;
                route 58.58.58.3/32 next-hop lt-0/0/0.3;
                route 58.58.58.4/32 next-hop lt-0/0/0.3;
            }
        }
    }
}


routing-options {
    static {
        route 60.60.60.1/32 next-hop lt-0/0/0.0;
        route 60.60.60.2/32 next-hop lt-0/0/0.0;
        route 60.60.60.3/32 next-hop lt-0/0/0.2;
        route 60.60.60.4/32 next-hop lt-0/0/0.2;
    }
}

class-of-service {
    interfaces {
        lt-0/0/0 {
            unit 0 {
                shaping-rate 10m;
            }
            unit 2 {
                shaping-rate 10m;
            }
```

```
}
```

How CoS Queuing Works

Figure 10 on page 241 shows CoS-related processing that occurs for traffic entering and exiting a tunnel. For information on flow-based packet processing, see the Flow-Based and Packet-Based Processing User Guide for Security Devices.

Figure 10: CoS Processing for Tunnel Traffic


Inbound traffic traversing through the tunnel:
Tunnel interface
Egress interface
Egress interface
Physical ingress interface
Input (classifier + policer)
Flow
Output (classifier+ policer)
Queuing Scheduling Shaping
g020124
Outbound traffic traversing through the tunnel:
Ingress interface
Tunnel interface
Tunnel interface
Physical egress interface
Ingress interface
Input (classifier +policer)
Flow
Output (classifier +policer)
Queuing Scheduling Shaping
Queuing Scheduling Shaping

Limitations on CoS Shapers for Tunnel Interfaces

When defining a CoS shaping rate on a tunnel interface, be aware of the following restrictions:

- The shaping rate on the tunnel interface must be less than that of the physical egress interface.
- The shaping rate only measures the packet size that includes the Layer 3 packet with GRE or IP-IP encapsulation. The Layer 2 encapsulation added by the physical interface is not factored into the shaping rate measurement.

- The CoS behavior works as expected only when the physical interface carries the shaped GRE or IP-IP tunnel traffic alone. If the physical interface carries other traffic, thereby lowering the available bandwidth for tunnel interface traffic, the CoS features do not work as expected.
- You cannot configure a *logical interface* shaper and a virtual circuit shaper simultaneously on the router. If virtual circuit shaping is desired, do not define a logical interface shaper. Instead, define a shaping rate for all the virtual circuits.

RELATED DOCUMENTATION

Example: Configuring CoS Queuing for GRE or IP-IP Tunnels | **243**

Understanding the ToS Value of a Tunnel Packet

To ensure that the tunneled packet continues to have the same CoS treatment even in the physical interface, you must preserve the type-of-service (ToS) value from the inner IP header to the outer IP header.

For transit traffic, Junos OS preserves the CoS value of the tunnel packet for both GRE and IP-IP tunnel interfaces. The inner IPv4 or IPv6 ToS bits are copied to the outer IPv4 ToS header for both types of tunnel interfaces.

For Routing Engine traffic, however, the router handles GRE tunnel interface traffic differently from IP-IP tunnel interface traffic. Unlike for IP-IP tunnels, the IPv4 ToS bits are not copied to the outer IPv4 header by default. You have a configuration option to copy the ToS value from the packet's inner IPv4 header to the outer IPv4 header.

To copy the inner ToS bits to the outer IP header (which is required for some tunneled routing protocols) on packets sent by the Routing Engine, include the `copy-tos-to-outer-ip-header` statement at the logical unit hierarchy level of a GRE interface.

> **NOTE**: For IPv6 traffic, the inner ToS value is not copied to the outer IPv4 header for both GRE and IP-IP tunnel interfaces even if the `copy-tos-to-outer-ip-header` statement is specified.

This example copies the inner ToS bits to the outer IP header on a GRE tunnel:

```
[edit interfaces]
gr-0/0/0 {
    unit 0 {
        copy-tos-to-outer-ip-header;
        family inet;
    }
}
```

RELATED DOCUMENTATION

CoS Queuing for Tunnels Overview | 237

Example: Configuring CoS Queuing for GRE or IP-IP Tunnels | 243

Example: Configuring CoS Queuing for GRE or IP-IP Tunnels

IN THIS SECTION

- Requirements | 243
- Overview | 244
- Configuration | 245
- Verification | 247

This example shows how to configure CoS queuing for GRE or IP-IP tunnels.

NOTE: CoS queuing is not supported on GRE tunnels in chassis clusters.

Requirements

Before you begin:

- Establish a main office and a branch office connected by a VPN using GRE or IP-IP tunneled interfaces.
- Configure forwarding classes and schedulers. See "Example: Assigning Forwarding Classes to Output Queues" on page 86 and "Example: Configuring Class-of-Service Schedulers on a Security Device" on page 133.

- Configure a scheduler map and apply the scheduler map to the tunnel interface. See "Example: Configuring and Applying Scheduler Maps" on page 149.
- Configure classifiers and apply them to the tunnel interface. See "Example: Configure and Apply Custom BA Classifiers" on page 23.
- Create rewrite rules and apply them to the tunnel interface. See "Example: Configure and Apply Rewrite Rules on a Security Device" on page 110.

Overview

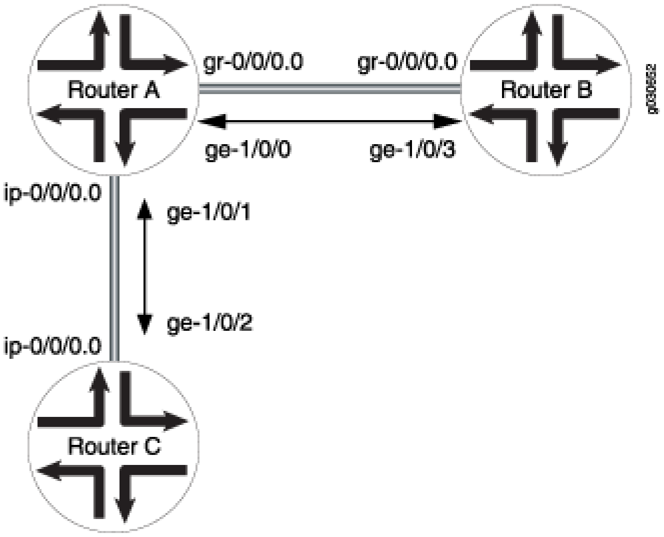
IN THIS SECTION

- Topology | **245**

In this example, you enable tunnel queuing, define the GRE tunnel interface as gr-0/0/0, (Alternatively, you could define the IP-IP tunnel interface as ip-0/0/0.) and set the per unit scheduler. You then set the GRE tunnel's line rate as 100 Mbps by using the shaper definition.

In Figure 11 on page 244, Router A has a GRE tunnel established with Router B through interface `ge-1/0/0`. Router A also has an IP-IP tunnel established with Router C through interface `ge-1/0/1`. Router A is configured so that tunnel-queuing is enabled. Router B and Router C do not have tunnel-queuing configured.

Figure 11: Configuring CoS Queuing for GRE Tunnels



Topology

Configuration

IN THIS SECTION

- Procedure | 245

Procedure

CLI Quick Configuration

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

```
set chassis fpc 0 pic 0 tunnel-queuing
set interfaces gr-0/0/0 unit 0
set interfaces gr-0/0/0 per-unit-scheduler
set class-of-services interfaces gr-0/0/0 unit 0 shaping-rate 100m
```

Step-by-Step Procedure

To configure CoS queuing for GRE tunnels:

1. Enable tunnel queuing on the device.

   ```
   [edit]
   user@host# set chassis fpc 0 pic 0 tunnel-queuing
   ```

2. Define the GRE tunnel interface.

   ```
   [edit]
   user@host# set interfaces gr-0/0/0 unit 0
   ```

3. Define the per-unit scheduler for the GRE tunnel interface.

```
[edit]
user@host# set interfaces gr-0/0/0 per-unit-scheduler
```

4. Define the GRE tunnel’s line rate by using the shaper definition.

```
[edit]
user@host# set class-of-services interfaces gr-0/0/0 unit 0 shaping-rate 100m
```

Results

From configuration mode, confirm your configuration by entering the `show class-of-service interfaces gr-0/0/0`, `show interfaces gr-0/0/0` , and `show chassis` commands. If the output does not display the intended configuration, repeat the configuration instructions in this example to correct it.

```
[edit]
    user@host# show class-of-service interfaces gr-0/0/0
    unit 0 {
    shaping-rate 100m;
    }
    [edit]
        user@host# show interfaces gr-0/0/0
        per-unit-scheduler;
    unit 0;
    [edit]
        user@host# show chassis
        fpc 0 {
        pic 0 {
        tunnel-queuing;
    }
}
```

If you are done configuring the device, enter `commit` from configuration mode.

Verification

IN THIS SECTION

- Verifying a CoS Queuing for GRE Tunnel Configuration | **247**
- Verifying a CoS Queuing for IP-IP Tunnel Configuration | **249**

Confirm that the configuration is working properly.

Verifying a CoS Queuing for GRE Tunnel Configuration

Purpose

Verify that the device is configured properly for tunnel configuration.

Action

From configuration mode, enter the `show interfaces queue gr-0/0/0.0` command.

NOTE: If you enter `gr-0/0/0.0` only, queue information for all tunnels is displayed. If you enter `gr-0/0/0.0`, queue information for the specific tunnel is displayed.

```
user@host> show  interfaces queue gr-0/0/0.0
Logical interface gr-0/0/0.0 (Index 68) (SNMP ifIndex 112)
Forwarding classes: 8 supported, 4 in use
Egress queues: 8 supported, 4 in use Burst size: 0
Queue: 0, Forwarding classes: VOICE
  Queued:
    Packets              :              7117734               7998 pps
    Bytes                :            512476848            4606848 bps
  Transmitted:
    Packets              :              4548146               3459 pps
    Bytes                :            327466512            1992912 bps
    Tail-dropped packets :                    0                  0 pps
    RED-dropped packets  :              2569421               4537 pps
     Low                 :                    0                  0 pps
     Medium-low          :                    0                  0 pps
     Medium-high         :                 0                   0 pps
     High                :           2569421                4537 pps
    RED-dropped bytes    :         184998312             2613640 bps
     Low                 :                 0                   0 bps
     Medium-low          :                 0                   0 bps
     Medium-high         :                 0                   0 bps
     High                :         184998312             2613640 bps
Queue: 1, Forwarding classes: GOLD
  Queued:
    Packets              :            117600                   0 pps
    Bytes                :           8467200                   0 bps
  Transmitted:
    Packets              :            102435                   0 pps
    Bytes                :           7375320                   0 bps
    Tail-dropped packets :                 0                   0 pps
    RED-dropped packets  :             15165                   0 pps
     Low                 :                 0                   0 pps
     Medium-low          :                 0                   0 pps
     Medium-high         :                 0                   0 pps
     High                :             15165                   0 pps
    RED-dropped bytes    :           1091880                   0 bps
     Low                 :                 0                   0 bps
     Medium-low          :                 0                   0 bps
     Medium-high         :                 0                   0 bps
     High                :           1091880                   0 bps
Queue: 2, Forwarding classes: SILVER
  Queued:
    Packets              :                 0                   0 pps
    Bytes                :                 0                   0 bps
  Transmitted:
    Packets              :                 0                   0 pps
    Bytes                :                 0                   0 bps
    Tail-dropped packets :                 0                   0 pps
    RED-dropped packets  :                 0                   0 pps
     Low                 :                 0                   0 pps
     Medium-low          :                 0                   0 pps
     Medium-high         :                 0                   0 pps
     High                :                 0                   0 pps
    RED-dropped bytes    :                 0                   0 bps
     Low                 :                 0                   0 bps
     Medium-low          :                 0                   0 bps
     Medium-high         :                 0                   0 bps
     High                :                 0                   0 bps
```

```
Queue: 3, Forwarding classes: BRONZE
  Queued:
    Packets              :                    0                    0 pps
    Bytes                :                    0                    0 bps
  Transmitted:
    Packets              :                    0                    0 pps
    Bytes                :                    0                    0 bps
    Tail-dropped packets :                    0                    0 pps
    RED-dropped packets  :                    0                    0 pps
     Low                 :                    0                    0 pps
     Medium-low          :                    0                    0 pps
     Medium-high         :                    0                    0 pps
     High                :                    0                    0 pps
    RED-dropped bytes    :                    0                    0 bps
     Low                 :                    0                    0 bps
     Medium-low          :                    0                    0 bps
     Medium-high         :                    0                    0 bps
     High                :                    0                    0 bps
```

Verifying a CoS Queuing for IP-IP Tunnel Configuration

Purpose

Verify that the device is configured properly for tunnel configuration.

Action

From configuration mode, enter the `show interfaces queue ip-0/0/0.0` command.

> **NOTE**: If you enter `ip-0/0/0.0` only, queue information for all tunnels is displayed. If you enter `ip-0/0/0.0`, queue information for the specific tunnel is displayed.

RELATED DOCUMENTATION

CoS Queuing for Tunnels Overview | 237

Understanding the ToS Value of a Tunnel Packet | 242

Copy Outer IP Header DSCP and ECN to Inner IP Header

You can copy a Differentiated Services Code Point (DSCP) (outer DSCP+ECN) field from the outer IP header encrypted packet to the inner IP header plain text message on the decryption path.

The benefit in enabling this feature is that after IPsec decryption, clear text packets can follow the inner CoS (DSCP+ECN) rules.

This feature supports chassis cluster and also supports IPv6 and IPv4. The following are supported:

- Copying outer IPv4 DSCP and Explicit Congestion Notification (ECN) field to inner IPv4 DSCP and ECN field
- Copying outer IPv6 DSCP and ECN field to inner IPv6 DSCP and ECN field
- Copying outer IPv4 DSCP and ECN field to inner IPv6 DSCP and ECN field
- Copying outer IPv6 DSCP and ECN field to inner IPv4 DSCP and ECN field

By default this feature is disabled. When you enable this feature on a VPN object, the corresponding IPsec security Association (SA) is cleared and reestablished.

- To enable the feature:

  ```
  set security ipsec vpn vpn-name copy-outer-dscp
  ```

- To disable the feature:

  ```
  delete security ipsec vpn vpn-name copy-outer-dscp
  ```

- To verify whether the feature is enabled or not:

  ```
  show security ipsec security-associations detail
  ```

RELATED DOCUMENTATION

IPsec VPN User Guide for Security Devices

show security ipsec security-associations

Understanding CoS Support on st0 Interfaces

IN THIS SECTION

- Limitations of CoS support on VPN st0 interfaces | 251
- Platform-Specific Behavior | 252

You can configure class of service (CoS) features such as classifier, policer, queuing, scheduling, shaping, rewriting rules, and virtual channels on the secure tunnel interface (st0) for point-to-point VPNs.

The st0 tunnel interface is an internal interface. Route-based VPNs can use the st0 interface to route cleartext traffic to an IPsec VPN tunnel. The st0 interface supports the following CoS features on all available SRX Series Firewalls and vSRX2.0:

- Classifiers
- Policers
- Queuing, scheduling, and shaping
- Rewrite rules
- Virtual channels

NOTE: Rewriting/remarking is the act of replacing a value in the header of an outgoing packet to identify the class assigned to the packet by the transmitting router. It is possible to rewrite each of the packet header types using each of the marking types (IEEE 802.1p, MPLS EXP, IPv4 Precedence, IPv4 DSCP, or IPv6 DSCP).

Rewrite applies on egress interfaces only.

Limitations of CoS support on VPN st0 interfaces

The following limitations apply to CoS support on VPN st0 interfaces:

- The maximum number for software queues is 2048. If the number of st0 interfaces exceeds 2048, not enough software queues can be created for all the st0 interfaces.
- Only route-based VPNs can apply CoS features on st0 interfaces. Table 43 on page 252 describes the st0 CoS feature support for different types of VPNs.

Table 43: CoS Feature Support for VPN

Classifier Features	Site-to-Site VPN (P2P)	AutoVPN (P2P)	Site-to-Site/Auto VPN /AD-VPN (P2MP)
Classifiers, policers, and rewriting markers	Supported	Supported	Supported
Queueing, scheduling, and shaping based on st0 logical interfaces	Supported	Not supported	Not supported
Queueing, scheduling, and shaping based on virtual channels	Supported	Supported	Supported

- When defining a CoS shaping rate on an st0 tunnel interface, consider the following restrictions:
  - The shaping rate on the tunnel interface must be less than that of the physical egress interface.
  - The shaping rate only measures the packet size that includes the inner Layer 3 cleartext packet with an ESP/AH header and an outer IP header encapsulation. The outer Layer 2 encapsulation added by the physical interface is not factored into the shaping rate measurement.
  - The CoS behavior works as expected when the physical interface carries the shaped GRE or IP-IP tunnel traffic only. If the physical interface carries other traffic, thereby lowering the available bandwidth for tunnel interface traffic, the CoS features do not work as expected.

Platform-Specific Behavior

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

Table 44: Platform-Specific Behavior

Platform	Difference
SRX300, SRX320, SRX340, SRX345, and SRX550HM	• On SRX300, SRX320, SRX340, SRX345, and SRX550HM firewalls, one st0 logical interface can bind to multiple VPN tunnels. The eight queues for the st0 logical interface cannot reroute the traffic to different tunnels, so pre-tunneling is not supported. • You can use the virtual channel feature as a workaround on SRX300, SRX320, SRX340, SRX345, and SRX550HM firewalls.
SRX550M, SRX5400, SRX5600, and SRX5800	• On SRX550M, SRX5400, SRX5600, and SRX5800 firewalls, bandwidth limit and burst size limit values in a policer configuration are a per-SPU, not per-system limitation. This is the same policer behavior as on the physical interface.

CHAPTER 12

Naming Components with Code-Point Aliases

IN THIS CHAPTER

- Code-Point Aliases Overview | 254
- Default CoS Values and Aliases | 255
- Example: Defining Code-Point Aliases for Bits on a Security Device | 258

Code-Point Aliases Overview

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 class-of-service (CoS) components, such as classifiers, drop-profile maps, and *rewrite rules*.

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.

The following types of code points are supported by Junos operating system (OS):

- DSCP—Defines aliases for DiffServ code point (DSCP) IPv4 values.

  You can refer to these aliases when you configure classes and define classifiers.

- DSCP-IPv6—Defines aliases for DSCP IPv6 values.

  You can refer to these aliases when you configure classes and define classifiers.

- EXP—Defines aliases for MPLS EXP bits.

  You can map MPLS EXP bits to the device forwarding classes.

- inet-precedence—Defines aliases for IPv4 precedence values.

  Precedence values are modified in the IPv4 type-of-service (ToS) field and mapped to values that correspond to levels of service.

RELATED DOCUMENTATION

Default CoS Values and Aliases | 255
Example: Defining Code-Point Aliases for Bits on a Security Device | 258

Default CoS Values and Aliases

Table 45 on page 255 shows the default mapping between the standard aliases and the bit values.

Table 45: Standard CoS Aliases and Bit Values

CoS Value Type	Alias	Bit Value
MPLS EXP	be	000
	be1	001
	ef	010
	ef1	011
	af11	100
	af12	101
	nc1/cs6	110
	nc2/cs7	111
DSCP and DSCP IPv6	ef	101110
	af11	001010
	af12	001100

Table 45: Standard CoS Aliases and Bit Values *(Continued)*

CoS Value Type	Alias	Bit Value
	af13	001110
	af21	010010
	af22	010100
	af23	010110
	af31	011010
	af32	011100
	af33	011110
	af41	100010
	af42	100100
	af43	100110
	be	000000
	cs1	001000
	cs2	010000
	cs3	011000
	cs4	100000

Table 45: Standard CoS Aliases and Bit Values *(Continued)*

CoS Value Type	Alias	Bit Value
	cs5	101000
	nc1/cs6	110000
	nc2/cs7	111000
IEEE 802.1	be	000
	be1	001
	ef	010
	ef1	011
	af11	100
	af12	101
	nc1/cs6	110
	nc2/cs7	111
IP precedence	be	000
	be1	001
	ef	010
	ef1	011

Table 45: Standard CoS Aliases and Bit Values *(Continued)*

CoS Value Type	Alias	Bit Value
	af11	100
	af12	101
	nc1/cs6	110
	nc2/cs7	111

RELATED DOCUMENTATION

Code-Point Aliases Overview | 254
Example: Defining Code-Point Aliases for Bits on a Security Device | 258

Example: Defining Code-Point Aliases for Bits on a Security Device

IN THIS SECTION

- Requirements | 258
- Overview | 259
- Configuration | 259
- Verification | 260

This example shows how to define code-point aliases for bits on a device.

Requirements

Before you begin, determine which default mapping to use. See "Default CoS Values and Aliases" on page 255.

Overview

In this example, you configure class of service and specify names and values for the CoS code-point aliases that you want to configure. Finally, you specify CoS value using the appropriate formats.

Configuration

IN THIS SECTION

- Procedure | 259

Procedure

Step-by-Step Procedure

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 Junos OS CLI User Guide.

To define code-point aliases for bits on a device:

1. Configure class of service.

```
[edit]
user@host# edit class-of-service
```

2. Specify CoS values.

```
[edit class-of-service]
user@host# set code-point-aliases dscp my1 110001
user@host# set code-point-aliases dscp my2 101110
user@host# set code-point-aliases dscp be 000001
user@host# set code-point-aliases dscp cs7 110000
```

3. If you are done configuring the device, commit the configuration.

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

Verification

To verify the configuration is working properly, enter the `show class-of-service code-point-aliases dscp` command.

RELATED DOCUMENTATION

Code-Point Aliases Overview | **254**

3
PART

Configuring Class of Service Scheduler Hierarchy

- Controlling Traffic by Configuring Scheduler Hierarchy | 262

CHAPTER 13

Controlling Traffic by Configuring Scheduler Hierarchy

IN THIS CHAPTER

- Understanding Hierarchical Schedulers | **262**
- Understanding Internal Scheduler Nodes | **266**
- Example: Configuring a Four-Level Scheduler Hierarchy | **267**
- Example: Controlling Remaining Traffic | **285**

Understanding Hierarchical Schedulers

Hierarchical schedules consist of nodes and queues. Queues terminate the CLI hierarchy. Nodes can be either root nodes, leaf nodes, or internal (non-leaf) nodes. Internal nodes are nodes that have other nodes as “children” in the hierarchy. For example, if an `interface-set` statement is configured with a *logical interface* (such as unit 0) and queue, then the `interface-set` is an internal node at level 2 of the hierarchy. However, if there are no traffic control profiles configured on logical interfaces, then the interface set is at level 3 of the hierarchy.

Table 46 on page 262 shows how the configuration of an interface set or logical interface affects the terminology of hierarchical scheduler nodes.

Table 46: Hierarchical Scheduler Nodes

Root Node (Level 1)	Internal Node (Level 2)	Leaf Node (Level 3)	Queue (Level 4)
Physical interface	Interface set	Logical interfaces	One or more queues
Physical interface	–	Interface set	One or more queues
Physical interface	–	Logical interfaces	One or more queues

When used, the interface set level of the hierarchy falls between the physical interface level (level 1) and the logical interface (level 3). Queues are always level 4 of the hierarchy. The schedulers hold the information about the queues, the last level of the hierarchy. In all cases, the properties for level 4 of the hierarchical schedulers are determined by the scheduler map.

Hierarchical schedulers add CoS parameters to the new interface set level of the configuration. They use traffic control profiles to set values for parameters such as shaping rate (the peak information rate [PIR]), guaranteed rate (the committed information rate [CIR] on these interfaces), and scheduler maps (the queues and resources assigned to traffic).

The following CoS configuration places the following parameters in traffic control profiles at various levels:

- Traffic control profile at the port level (`tcp-port-level1`):
    - A shaping rate (PIR) of 100 Mbps
    - A delay buffer rate of 100 Mbps
- Traffic control profile at the interface set level (`tcp-interface-level2`):
    - A shaping rate (PIR) of 60 Mbps
    - A guaranteed rate (CIR) of 40 Mbps
- Traffic control profile at the logical interface level (`tcp-unit-level3`):
    - A shaping rate (PIR) of 50 Mbps
    - A guaranteed rate (CIR) of 30 Mbps
    - A scheduler map called smap1 to hold various queue properties (level 4)
    - A delay buffer rate of 40 Mbps

In this case, the traffic control profiles look like this:

```
[edit class-of-service traffic-control-profiles]
tcp-port-level1 { # This is the physical port level
    shaping-rate 100m;
    delay-buffer-rate 100m;
}
tcp-interface-level2 { # This is the interface set level
    shaping-rate 60m;
    guaranteed-rate 40m;
}
tcp-unit-level3 { # This is the logical interface level
```

```
    shaping-rate 50m;
    guaranteed-rate 30m;
    scheduler-map smap1;
    delay-buffer-rate 40m;
}
```

Once configured, the traffic control profiles must be applied to the proper places in the CoS interfaces hierarchy.

```
[edit class-of-service interfaces]
interface-set level-2 {
    output-traffic-control-profile tcp-interface-level-2;
}
ge-0/1/0 {
    output-traffic-control-profile tcp-port-level-1;
    unit 0 {
        output-traffic-control-profile tcp-unit-level-3;
    }
}
```

Interface sets can be defined as a list of logical interfaces, for example, unit 100, unit 200, and so on. Service providers can use these statements to group interfaces to apply scheduling parameters such as guaranteed rate and shaping rate to the traffic in the groups. Interface sets are currently only used by CoS, but they are applied at the `[edit interfaces]` hierarchy level so that they might be available to other services.

All traffic heading downstream must be gathered into an interface set with the `interface-set` statement at the [edit class-of-service interfaces] hierarchy level.

NOTE: Ranges are not supported; you must list each logical interface separately.

Although the interface set is applied at the [edit interfaces] hierarchy level, the CoS parameters for the interface set are defined at the [edit class-of-service interfaces] hierarchy level, usually with the `output-traffic-control-profile` *`profile-name`* statement.

You cannot specify an interface set mixing the logical interface, S-VLAN, or VLAN outer tag list forms of the `interface-set` statement. A logical interface can only belong to one interface set. If you try to add the same logical interface to different interface sets, the commit will fail.

This example will generate a commit error:

```
[edit interfaces]
interface-set set-one {
    ge-2/0/0 {
        unit 0;
        unit 2;
    }
}
interface-set set-two {
    ge-2/0/0 {
        unit 1;
        unit 3;
        unit 0; # COMMIT ERROR! Unit 0 already belongs to -set-one.
    }
}
```

Members of an interface set cannot span multiple physical interfaces. Only one physical interface is allowed to appear in an interface set.

This configuration is not supported:

```
[edit interfaces]
interface-set set-group {
    ge-0/0/1 {
        unit 0;
        unit 1;
    }
    ge-0/0/2 { # This type of configuration is NOT supported in the same interface set!
        unit 0;
        unit 1;
    }
}
```

You can configure many logical interfaces under an interface. However, only a subset of them might have a traffic control profile attached. For example, you can configure three logical interfaces (units) over the same service VLAN, but you can apply a traffic control profile specifying best-effort and voice queues to only one of the logical interface units. Traffic from the two remaining logical interfaces is considered remaining traffic.

The scheduler map configured at individual interfaces (Level 3), interface sets (Level 2), or physical ports (Level 1), defines packet scheduling behavior at different levels. You can group logical interfaces in an interface set and configure the interfaces with scheduler maps. Any egress packet arriving at the physical or logical interfaces will be handled by the interface specific scheduler. If the scheduler map is not configured at the interface level, the packet will be handled by the scheduler configured at the interface set level or the port level.

RELATED DOCUMENTATION

Example: Configuring a Four-Level Scheduler Hierarchy | 267

Example: Controlling Remaining Traffic | 285

Understanding Internal Scheduler Nodes | 266

Understanding Internal Scheduler Nodes

A node in the hierarchy is considered internal if either of the following conditions apply:

- One of its children nodes has a traffic control profile configured and applied.
- You configure the `internal-node` statement.

There are more resources available at the *logical interface* (unit) level than at the interface set level. It might be desirable to configure all resources at a single level, rather than spread over several levels. The `internal-node` statement provides this flexibility. This can be a helpful configuration device when interface-set queuing without logical interfaces is used exclusively on the interface.

You can use the `internal-node` statement to raise the interface set without children to the same level as the other configured interface sets with children, allowing them to compete for the same set of resources.

Using the `internal-node` statement allows statements to all be scheduled at the same level with or without children.

The following example makes the interface sets `if-set-1` and `if-set-2` internal:

```
[edit class-of-service interfaces ]
interface-set {
    if-set-1 {
        internal-node;
        output-traffic-control-profile tcp-200m-no-smap;
    }
    if-set-2 {
        internal-node;
        output-traffic-control-profile tcp-100m-no-smap;
    }
}
```

If an interface set has logical interfaces configured with a traffic control profile, then the use of the `internal-node` statement has no effect.

Internal nodes can specify a `traffic-control-profile-remaining` statement.

RELATED DOCUMENTATION

- Understanding Hierarchical Schedulers | 262
- Example: Configuring a Four-Level Scheduler Hierarchy | 267
- Example: Controlling Remaining Traffic | 285

Example: Configuring a Four-Level Scheduler Hierarchy

IN THIS SECTION

- Requirements | 267
- Overview | 268
- Configuration | 269
- Verification | 284

This example shows how to configure a 4-level hierarchy of schedulers.

Requirements

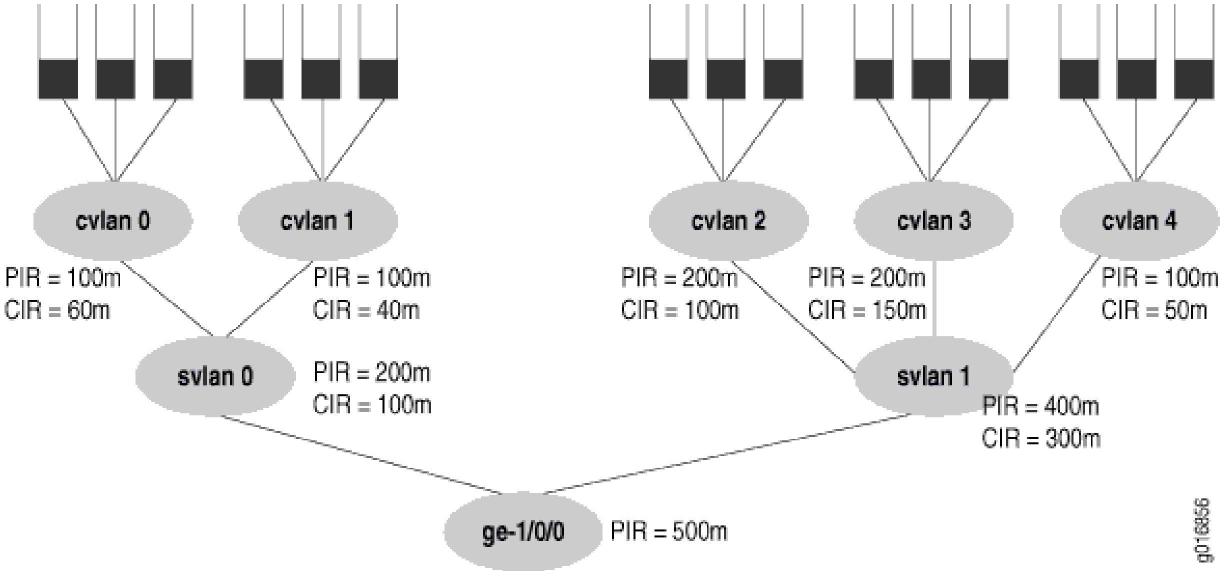
Before you begin:

- Review how to configure schedulers. See "Example: Configuring Class-of-Service Schedulers on a Security Device" on page 133.
- Review RED drop profiles. See Understanding RED Drop Profiles.
- Review how to configure and apply scheduler maps. See "Example: Configuring and Applying Scheduler Maps" on page 149.

Overview

The configuration parameters for this example are shown in Figure 12 on page 268. The queues are shown at the top of the figure with the other three levels of the hierarchy below.

Figure 12: Building a Scheduler Hierarchy



cvlan 0
cvlan 1
cvlan 2
cvlan 3
cvlan 4
PIR = 100m
CIR = 60m
PIR = 100m
CIR = 40m
PIR = 200m
CIR = 100m
PIR = 200m
CIR = 150m
PIR = 100m
CIR = 50m
svlan 0
PIR = 200m
CIR = 100m
svlan 1
PIR = 400m
CIR = 300m
ge-1/0/0
PIR = 500m
g016856

Figure 12 on page 268's PIR values will be configured as the shaping rates, and the CIRs will be configured as the guaranteed rate on the Ethernet interface `ge-1/0/0`. The PIR can be oversubscribed (that is, the sum of the children PIRs can exceed the parent's, as in `svlan 1`, where 200 + 200 + 100 exceeds the parent rate of 400). However, the sum of the children node level's CIRs must never exceed the parent node's CIR, as shown in all the service VLANs (otherwise, the guaranteed rate could never be provided in all cases).

NOTE: Although a shaping rate can be applied directly to the physical interface, hierarchical schedulers must use a traffic control profile to hold the shaping rate parameter.

The keyword to configure hierarchical schedulers is at the physical interface level, as are VLAN tagging and the VLAN IDs. In this example, the interface sets are defined by logical interfaces (units) and not outer VLAN tags. All VLAN tags in this example are customer VLAN tags.

The traffic control profiles in this example are for both the service VLAN level (logical interfaces) and the customer VLAN (VLAN tag) level.

This example shows all details of the CoS configuration for the `ge-1/0/0` interface in Figure 12 on page 268.

Configuration

IN THIS SECTION

- Configuring the Interfaces | 269
- Configuring the Interface Sets | 271
- Configuring the Forwarding Classes | 272
- Configuring the Traffic Control Profiles | 273
- Configuring the Schedulers | 276
- Configuring the Drop Profiles | 279
- Configuring the Scheduler Maps | 280
- Applying Traffic Control Profiles | 282

This section contains the following topics:

Configuring the Interfaces

CLI Quick Configuration

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

```
set interfaces ge-1/0/0 hierarchical-scheduler
set interfaces ge-1/0/0 vlan-tagging
set interfaces ge-1/0/0 unit 0 vlan-id 100
set interfaces ge-1/0/0 unit 1 vlan-id 101
set interfaces ge-1/0/0 unit 2 vlan-id 102
set interfaces ge-1/0/0 unit 3 vlan-id 103
set interfaces ge-1/0/0 unit 4 vlan-id 104
```

Step-by-Step Procedure

To configure the interfaces:

1. Create the physical interface, and enable hierarchical scheduling and VLAN tagging.

   ```
   [edit interfaces ge-1/0/0]
   user@host# set hierarchical-scheduler
   user@host# set vlan-tagging
   ```

2. Create logical interfaces and assign VLAN IDs.

   ```
   [edit interface ge-1/0/0]
   user@host# set unit 0 vlan-id 100
   user@host# set unit 1 vlan-id 101
   user@host# set unit 2 vlan-id 102
   user@host# set unit 3 vlan-id 103
   user@host# set unit 4 vlan-id 104
   ```

Results

From configuration mode, confirm your configuration by entering the `show interface ge-1/0/0` command. If the output does not display the intended configuration, repeat the configuration instructions in this example to correct it.

```
[edit]
user@host# show interface ge-1/0/0
hierarchical-scheduler;
vlan-tagging;
unit 0 {
    vlan-id 100;
}
unit 1 {
    vlan-id 101;
}
unit 2 {
    vlan-id 102;
}
unit 3 {
    vlan-id 103;
}
unit 4 {
```

```
    vlan-id 104;
}
```

If you are done configuring the device, enter `commit` from configuration mode.

Configuring the Interface Sets

CLI Quick Configuration

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

```
set interfaces interface-set svlan-0 interface ge-1/0/0 unit 0
set interfaces interface-set svlan-0 interface ge-1/0/0 unit 1
set interfaces interface-set svlan-1 interface ge-1/0/0 unit 2
set interfaces interface-set svlan-1 interface ge-1/0/0 unit 3
set interfaces interface-set svlan-1 interface ge-1/0/0 unit 4
```

Step-by-Step Procedure

To configure the interface sets:

1. Create the first logical interface and its CoS parameters.

```
[edit interfaces]
user@host# set interface-set svlan-0 interface ge-1/0/0 unit 0
user@host# set interface-set svlan-0 interface ge-1/0/0 unit 1
```

2. Create the second logical interface and its CoS parameters.

```
[edit interfaces]
user@host# set interface-set svlan-1 interface ge-1/0/0 unit 2
user@host# set interface-set svlan-1 interface ge-1/0/0 unit 3
user@host# set interface-set svlan-1 interface ge-1/0/0 unit 4
```

Results

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

```
[edit]
user@host# show interfaces
interface-set svlan-0 {
    interface ge-1/0/0 {
        unit 0;
        unit 1;
    }
}
interface-set svlan-1 {
    interface ge-1/0/0 {
        unit 2;
        unit 3;
        unit 4;
    }
}
```

If you are done configuring the device, enter `commit` from configuration mode.

Configuring the Forwarding Classes

CLI Quick Configuration

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

```
set class-of-service forwarding-classes queue 1 data
set class-of-service forwarding-classes queue 2 video
set class-of-service forwarding-classes queue 3 voice
```

Step-by-Step Procedure

To configure the forwarding classes:

1. Specify a queue number and map it to a class name.

```
[edit class-of-service forwarding-classes]
user@host# set queue 1 data
user@host# set queue 2 video
user@host# set queue 3 voice
```

Results

From configuration mode, confirm your configuration by entering the `show class-of-service forwarding-classes` command. If the output does not display the intended configuration, repeat the configuration instructions in this example to correct it.

```
[edit]
user@host# show class-of-service forwarding-classes
queue 1 data;
queue 2 video;
queue 3 voice;
```

If you are done configuring the device, enter `commit` from configuration mode.

Configuring the Traffic Control Profiles

CLI Quick Configuration

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

```
set class-of-service traffic-control-profiles tcp-500m-shaping-rate shaping-rate 500m
set class-of-service traffic-control-profiles tcp-svlan0 shaping-rate 200m
set class-of-service traffic-control-profiles tcp-svlan0 guaranteed-rate 100m
set class-of-service traffic-control-profiles tcp-svlan0 delay-buffer-rate 300m
set class-of-service traffic-control-profiles tcp-svlan1 shaping-rate 400m
set class-of-service traffic-control-profiles tcp-svlan1 guaranteed-rate 300m
set class-of-service traffic-control-profiles tcp-svlan1 delay-buffer-rate 100m
set class-of-service traffic-control-profiles tcp-cvlan0 shaping-rate 100m
set class-of-service traffic-control-profiles tcp-cvlan0 guaranteed-rate 60m
set class-of-service traffic-control-profiles tcp-cvlan0 scheduler-map tcp-map-cvlan0
set class-of-service traffic-control-profiles tcp-cvlan1 shaping-rate 100m
set class-of-service traffic-control-profiles tcp-cvlan1 guaranteed-rate 40m
set class-of-service traffic-control-profiles tcp-cvlan1 scheduler-map tcp-map-cvlan1
set class-of-service traffic-control-profiles tcp-cvlan2 shaping-rate 200m
set class-of-service traffic-control-profiles tcp-cvlan2 guaranteed-rate 100m
set class-of-service traffic-control-profiles tcp-cvlan2 scheduler-map tcp-map-cvlanx
set class-of-service traffic-control-profiles tcp-cvlan3 shaping-rate 200m
set class-of-service traffic-control-profiles tcp-cvlan3 guaranteed-rate 150m
set class-of-service traffic-control-profiles tcp-cvlan3 scheduler-map tcp-map-cvlanx
set class-of-service traffic-control-profiles tcp-cvlan4 shaping-rate 100m
set class-of-service traffic-control-profiles tcp-cvlan4 guaranteed-rate 50m
set class-of-service traffic-control-profiles tcp-cvlan4 scheduler-map tcp-map-cvlanx
```

Step-by-Step Procedure

To configure the traffic control profiles:

1. Create the traffic control profile for the physical interface.

```
[edit class-of-service traffic-control-profiles]
user@host# tcp-500m-shaping-rate shaping-rate 500m
```

2. Create the traffic control profiles and parameters for the S-VLAN (logical interfaces) level.

```
[edit class-of-service traffic-control-profiles]
user@host# set tcp-svlan0 shaping-rate 200m
user@host# set tcp-svlan0 guaranteed-rate 100m
user@host# set tcp-svlan0 delay-buffer-rate 300m
user@host# set tcp-svlan1 shaping-rate 400m
user@host# set tcp-svlan1 guaranteed-rate 300m
user@host# set tcp-svlan1 delay-buffer-rate 100m
```

3. Create the traffic control profiles and parameters for the C-VLAN (VLAN tags) level.

```
[edit class-of-service traffic-control-profiles]
user@host# set tcp-cvlan0 shaping-rate 100m
user@host# set tcp-cvlan0 guaranteed-rate 60m
user@host# set tcp-cvlan0 scheduler-map tcp-map-cvlan0
user@host# set tcp-cvlan1 shaping-rate 100m
```

```
user@host# set tcp-cvlan1 guaranteed-rate 40m
user@host# set tcp-cvlan1 scheduler-map tcp-map-cvlan1
user@host# set tcp-cvlan2 shaping-rate 200m
user@host# set tcp-cvlan2 guaranteed-rate 100m
user@host# set tcp-cvlan2 scheduler-map tcp-map-cvlanx
user@host# set tcp-cvlan3 shaping-rate 200m
user@host# set tcp-cvlan3 guaranteed-rate 150m
user@host# set tcp-cvlan3 scheduler-map tcp-map-cvlanx
user@host# set tcp-cvlan4 shaping-rate 100m
user@host# set tcp-cvlan4 guaranteed-rate 50m
user@host# set tcp-cvlan4 scheduler-map tcp-map-cvlanx
```

Results

From configuration mode, confirm your configuration by entering the `show class-of-service traffic-control-profiles` command. If the output does not display the intended configuration, repeat the configuration instructions in this example to correct it.

```
[edit]
user@host# show class-of-service traffic-control-profiles
tcp-500m-shaping-rate {
    shaping-rate 500m;
}
tcp-svlan0 {
    shaping-rate 200m;
    guaranteed-rate 100m;
    delay-buffer-rate 300m; # This parameter is not shown in the figure
}
tcp-svlan1 {
    shaping-rate 400m;
    guaranteed-rate 300m;
    delay-buffer-rate 100m; # This parameter is not shown in the figure
}
tcp-cvlan0 {
    shaping-rate 100m;
    guaranteed-rate 60m;
    scheduler-map tcp-map-cvlan0; # This example applies scheduler maps to customer VLANs
}
tcp-cvlan1 {
    shaping-rate 100m;
    guaranteed-rate 40m;
    scheduler-map tcp-map-cvlan1; # This example applies scheduler maps to customer VLANs
}
tcp-cvlan2 {
    shaping-rate 200m;
    guaranteed-rate 100m;
    scheduler-map tcp-map-cvlanx; # This example applies scheduler maps to customer VLANs
}
tcp-cvlan3 {
    shaping-rate 200m;
    guaranteed-rate 150m;
    scheduler-map tcp-map-cvlanx; # This example applies scheduler maps to customer VLANs
}
tcp-cvlan4 {
    shaping-rate 100m;
    guaranteed-rate 50m;
    scheduler-map tcp-map-cvlanx; # This example applies scheduler maps to customer VLANs
}
```

If you are done configuring the device, enter `commit` from configuration mode.

Configuring the Schedulers

CLI Quick Configuration

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

```
set class-of-service schedulers sched-cvlan0-qx transmit-rate 20m
set class-of-service schedulers sched-cvlan0-qx buffer-size temporal 100k
set class-of-service schedulers sched-cvlan0-qx priority low
set class-of-service schedulers sched-cvlan0-qx drop-profile-map loss-priority low protocol any
drop-profile dp-low
set class-of-service schedulers sched-cvlan0-qx drop-profile-map loss-priority high protocol any
drop-profile dp-high
set class-of-service schedulers sched-cvlan1-q0 transmit-rate 20m
set class-of-service schedulers sched-cvlan1-q0 buffer-size percent 40
set class-of-service schedulers sched-cvlan1-q0 priority high
set class-of-service schedulers sched-cvlan1-q0 drop-profile-map loss-priority low protocol any
drop-profile dp-low
set class-of-service schedulers sched-cvlan1-q0 drop-profile-map loss-priority high protocol any
```

```
drop-profile dp-high
set class-of-service schedulers sched-cvlanx-qx transmit-rate percent 30
set class-of-service schedulers sched-cvlanx-qx buffer-size percent 30
set class-of-service schedulers sched-cvlanx-qx drop-profile-map loss-priority low protocol any
drop-profile dp-low
set class-of-service schedulers sched-cvlanx-qx drop-profile-map loss-priority high protocol any
drop-profile dp-high
set class-of-service schedulers sched-cvlan1-qx transmit-rate 10m
set class-of-service schedulers sched-cvlan1-qx buffer-size temporal 100k
set class-of-service schedulers sched-cvlan1-qx drop-profile-map loss-priority low protocol any
drop-profile dp-low
set class-of-service schedulers sched-cvlan1-qx drop-profile-map loss-priority high protocol any
drop-profile dp-high
```

Step-by-Step Procedure

To configure the schedulers:

1. Create the schedulers and their parameters.

```
[edit class-of-service schedulers]
user@host# set sched-cvlan0-qx priority low transmit-rate 20m
user@host# set sched-cvlan0-qx buffer-size temporal 100k
user@host# set sched-cvlan0-qx priority low
user@host# set sched-cvlan0-qx drop-profile-map loss-priority low protocol any drop-profile
dp-low
user@host# set sched-cvlan0-qx drop-profile-map loss-priority high protocol any drop-profile
dp-high
user@host# set sched-cvlan1-q0 priority high transmit-rate 20m
user@host# set sched-cvlan1-q0 buffer-size percent 40
user@host# set sched-cvlan1-q0 priority high
user@host# set sched-cvlan1-q0 drop-profile-map loss-priority low protocol any drop-profile
dp-low
user@host# set sched-cvlan1-q0 drop-profile-map loss-priority high protocol any drop-profile
dp-high
user@host# set sched-cvlanx-qx transmit-rate percent 30
user@host# set sched-cvlanx-qx buffer-size percent 30
user@host# set sched-cvlanx-qx drop-profile-map loss-priority low protocol any drop-profile
dp-low
user@host# set sched-cvlanx-qx drop-profile-map loss-priority high protocol any drop-profile
dp-high
user@host# set sched-cvlan1-qx transmit-rate 10m
```

```
user@host# set sched-cvlan1-qx buffer-size temporal 100k
user@host# set sched-cvlan1-qx drop-profile-map loss-priority low protocol any drop-profile
dp-low
user@host# set sched-cvlan1-qx drop-profile-map loss-priority high protocol any drop-profile
dp-high
```

Results

From configuration mode, confirm your configuration by entering the `show class-of-service schedulers` command. If the output does not display the intended configuration, repeat the configuration instructions in this example to correct it.

```
[edit]
user@host# show class-of-service schedulers
sched-cvlan0-qx {
    transmit-rate 20m;
    buffer-size temporal 100k;
    priority low;
    drop-profile-map loss-priority low protocol any drop-profile dp-low;
    drop-profile-map loss-priority high protocol any drop-profile dp-high;
}
sched-cvlan1-q0 {
    transmit-rate 20m;
    buffer-size percent 40;
    priority high;
    drop-profile-map loss-priority low protocol any drop-profile dp-low;
    drop-profile-map loss-priority high protocol any drop-profile dp-high;
}
sched-cvlanx-qx {
    transmit-rate percent 30;
    buffer-size percent 30;
    drop-profile-map loss-priority low protocol any drop-profile dp-low;
    drop-profile-map loss-priority high protocol any drop-profile dp-high;
}
sched-cvlan1-qx {
    transmit-rate 10m;
    buffer-size temporal 100k;
    drop-profile-map loss-priority low protocol any drop-profile dp-low;
    drop-profile-map loss-priority high protocol any drop-profile dp-high;
}
```

If you are done configuring the device, enter `commit` from configuration mode.

Configuring the Drop Profiles

CLI Quick Configuration

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

```
set class-of-service drop-profiles dp-low interpolate fill-level 80 drop-probability 80
set class-of-service drop-profiles dp-low interpolate fill-level 100 drop-probability 100
set class-of-service drop-profiles dp-high interpolate fill-level 60 drop-probability 80
set class-of-service drop-profiles dp-high interpolate fill-level 80 drop-probability 100
```

Step-by-Step Procedure

To configure the drop profiles:

1. Create the low drop profile.

   ```
   [edit class-of-service drop-profiles]
   user@host# set dp-low interpolate fill-level 80 drop-probability 80
   user@host# set dp-low interpolate fill-level 100 drop-probability 100
   ```

2. Create the high drop profile.

   ```
   [edit class-of-service drop-profiles]
   user@host# set dp-high interpolate fill-level 60 drop-probability 80
   user@host# set dp-high interpolate fill-level 80 drop-probability 100
   ```

Results

From configuration mode, confirm your configuration by entering the `show class-of-service drop-profiles` command. If the output does not display the intended configuration, repeat the configuration instructions in this example to correct it.

```
[edit]
user@host# show class-of-service drop-profiles
dp-low {
    interpolate {
        fill-level [ 80 100 ];
        drop-probability [ 80 100 ];
    }
}
dp-high {
    interpolate {
        fill-level [ 60 80 ];
        drop-probability [ 80 100 ];
    }
}
```

If you are done configuring the device, enter `commit` from configuration mode.

Configuring the Scheduler Maps

CLI Quick Configuration

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

```
set class-of-service scheduler-maps tcp-map-cvlan0 forwarding-class voice scheduler sched-cvlan0-
qx
set class-of-service scheduler-maps tcp-map-cvlan0 forwarding-class video scheduler sched-cvlan0-
qx
set class-of-service scheduler-maps tcp-map-cvlan0 forwarding-class data scheduler sched-cvlan0-
qx
set class-of-service scheduler-maps tcp-map-cvlan1 forwarding-class voice scheduler sched-cvlan1-
q0
set class-of-service scheduler-maps tcp-map-cvlan1 forwarding-class video scheduler sched-cvlan1-
```

```
qx
set class-of-service scheduler-maps tcp-map-cvlan1 forwarding-class data scheduler sched-cvlan1-
qx
set class-of-service scheduler-maps tcp-map-cvlanx forwarding-class voice scheduler sched-cvlanx-
qx
set class-of-service scheduler-maps tcp-map-cvlanx forwarding-class video scheduler sched-cvlanx-
qx
set class-of-service scheduler-maps tcp-map-cvlanx forwarding-class data scheduler sched-cvlanx-
qx
```

Step-by-Step Procedure

To configure three scheduler maps:

1. Create the first scheduler map.

```
[edit class-of-service scheduler-maps]
user@host# set tcp-map-cvlan0 forwarding-class voice scheduler sched-cvlan0-qx
user@host# set tcp-map-cvlan0 forwarding-class video scheduler sched-cvlan0-qx
user@host# set tcp-map-cvlan0 forwarding-class data scheduler sched-cvlan0-qx
```

2. Create the second scheduler map.

```
[edit class-of-service scheduler-maps]
user@host# set tcp-map-cvlan1 forwarding-class voice scheduler sched-cvlan1-q0
user@host# set tcp-map-cvlan1 forwarding-class video scheduler sched-cvlan1-qx
user@host# set tcp-map-cvlan1 forwarding-class data scheduler sched-cvlan1-qx
```

3. Create the third scheduler map.

```
[edit class-of-service scheduler-maps]
user@host# set tcp-map-cvlanx forwarding-class voice scheduler sched-cvlanx-qx
user@host# set tcp-map-cvlanx forwarding-class video scheduler sched-cvlanx-qx
user@host# set tcp-map-cvlanx forwarding-class data scheduler sched-cvlanx-qx
```

Results

From configuration mode, confirm your configuration by entering the `show class-of-service scheduler-maps` command. If the output does not display the intended configuration, repeat the configuration instructions in this example to correct it.

```
[edit]
user@host# show class-of-service scheduler-maps
tcp-map-cvlan0 {
    forwarding-class voice scheduler sched-cvlan0-qx;
    forwarding-class video scheduler sched-cvlan0-qx;
    forwarding-class data scheduler sched-cvlan0-qx;
}
tcp-map-cvlan1 {
    forwarding-class voice scheduler sched-cvlan1-q0;
    forwarding-class video scheduler sched-cvlan1-qx;
    forwarding-class data scheduler sched-cvlan1-qx;
}
tcp-map-cvlanx {
    forwarding-class voice scheduler sched-cvlanx-qx;
    forwarding-class video scheduler sched-cvlanx-qx;
    forwarding-class data scheduler sched-cvlanx-qx;
}
```

If you are done configuring the device, enter `commit` from configuration mode.

Applying Traffic Control Profiles

CLI Quick Configuration

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

```
set class-of-service interfaces ge-1/0/0 output-traffic-control-profile tcp-500m-shaping-rate
set class-of-service interfaces ge-1/0/0 unit 0 output-traffic-control-profile tcp-cvlan0
set class-of-service interfaces ge-1/0/0 unit 1 output-traffic-control-profile tcp-cvlan1
set class-of-service interfaces ge-1/0/0 unit 2 output-traffic-control-profile tcp-cvlan2
set class-of-service interfaces ge-1/0/0 unit 3 output-traffic-control-profile tcp-cvlan3
set class-of-service interfaces ge-1/0/0 unit 4 output-traffic-control-profile tcp-cvlan4
set class-of-service interfaces interface-set svlan0 output-traffic-control-profile tcp-svlan0
set class-of-service interfaces interface-set svlan1 output-traffic-control-profile tcp-svlan1
```

Step-by-Step Procedure

To apply traffic control profiles:

1. Apply the traffic control profile for the interface.

```
[edit class-of-service interfaces ge-1/0/0]
user@host# set output-traffic-control-profile tcp-500m-shaping-rate
```

2. Apply the traffic control profiles for the C-VLANs.

```
[edit class-of-service interfaces ge-1/0/0]
user@host# set unit 0 output-control-traffic-control-profile tcp-cvlan0
user@host# set unit 1 output-control-traffic-control-profile tcp-cvlan1
user@host# set unit 2 output-control-traffic-control-profile tcp-cvlan2
user@host# set unit 3 output-control-traffic-control-profile tcp-cvlan3
user@host# set unit 4 output-control-traffic-control-profile tcp-cvlan4
```

3. Apply the traffic control profiles for the S-VLANs.

```
[edit class-of-service interfaces]
user@host# set interface-set-svlan0 output-control-traffic-control-profile tcp-svlan0
user@host# set interface-set-svlan1 output-control-traffic-control-profile tcp-svlan1
```

Results

From configuration mode, confirm your configuration by entering the `show class-of-service interfaces` command. If the output does not display the intended configuration, repeat the configuration instructions in this example to correct it.

```
[edit]
user@host# show class-of-service interfaces
ge-1/0/0 {
    output-traffic-control-profile tcp-500m-shaping-rate;
    unit 0 {
            output-traffic-control-profile tcp-cvlan0;
        }
        unit 1 {
            output-traffic-control-profile tcp-cvlan1;
        }
        unit 2 {
            output-traffic-control-profile tcp-cvlan2;
        }
        unit 3 {
            output-traffic-control-profile tcp-cvlan3;
        }
        unit 4 {
            output-traffic-control-profile tcp-cvlan4;
        }
    }
    interface-set svlan0 {
        output-traffic-control-profile tcp-svlan0;
    }
    interface-set svlan1 {
        output-traffic-control-profile tcp-svlan1;
    }
```

If you are done configuring the device, enter `commit` from configuration mode.

Verification

IN THIS SECTION

- Verifying Scheduler Hierarchy Configuration | 284

Verifying Scheduler Hierarchy Configuration

Purpose

Verify that the scheduler hierarchy is configured properly.

Action

From operational mode, enter the following commands:

- `show interface ge-1/0/0`
- `show class-of-service interfaces`
- `show class-of-service traffic-control-profiles`
- `show class-of-service schedulers`
- `show class-of-service drop-profiles`
- `show class-of-service scheduler-maps`

RELATED DOCUMENTATION

Understanding Hierarchical Schedulers | 262
Example: Controlling Remaining Traffic | 285
Understanding Internal Scheduler Nodes | 266

Example: Controlling Remaining Traffic

IN THIS SECTION

- Requirements | 285
- Overview | 286
- Configuration | 288
- Verification | 292

This example shows how to control remaining traffic from the remaining logical interfaces.

Requirements

Before you begin:

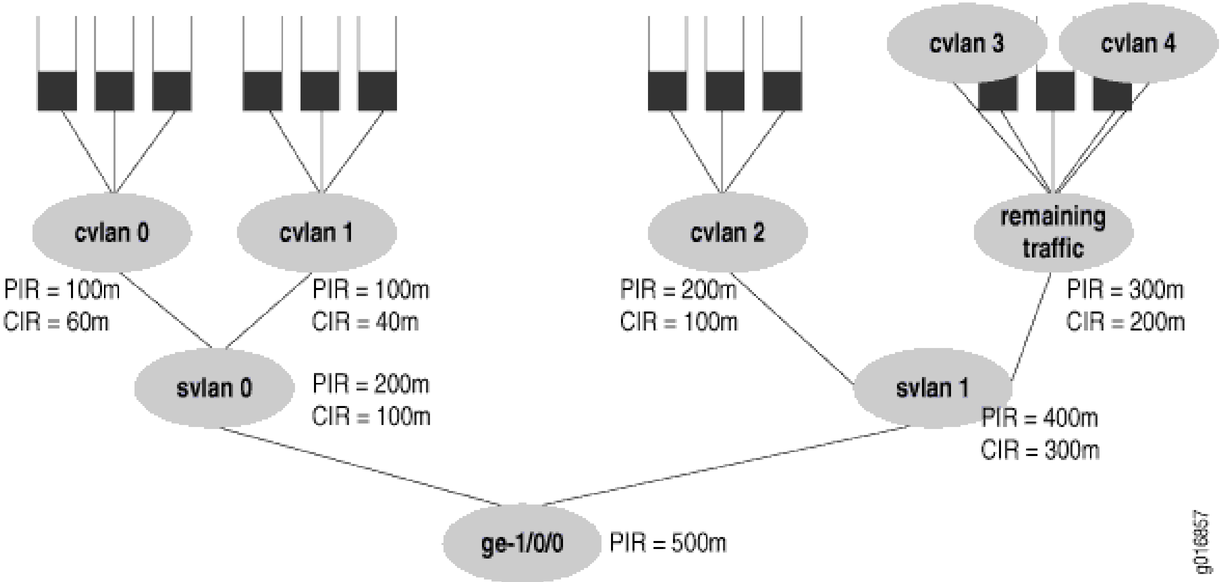
- Review how to configure schedulers. See "Example: Configuring Class-of-Service Schedulers on a Security Device" on page 133.
- Review how to configure and apply scheduler maps. See "Example: Configuring and Applying Scheduler Maps" on page 149.

Overview

To configure transmit rate guarantees for the remaining traffic, you configure the `output-traffic-control-profile-remaining` statement specifying a guaranteed rate for the remaining traffic. Without this statement, the remaining traffic gets a default, minimal bandwidth. Similarly, you can specify the `shaping-rate` and `delay-buffer-rate` statements in the traffic control profile referenced with the `output-traffic-control-profile-remaining` statement to shape and provide buffering for remaining traffic.

In the interface shown in Figure 13 on page 286, customer VLANs 3 and 4 have no explicit traffic control profile. However, the service provider might want to establish a shaping and guaranteed transmit rate for aggregate traffic heading for those C-VLANs. The solution is to configure and apply a traffic control profile for all remaining traffic on the interface.

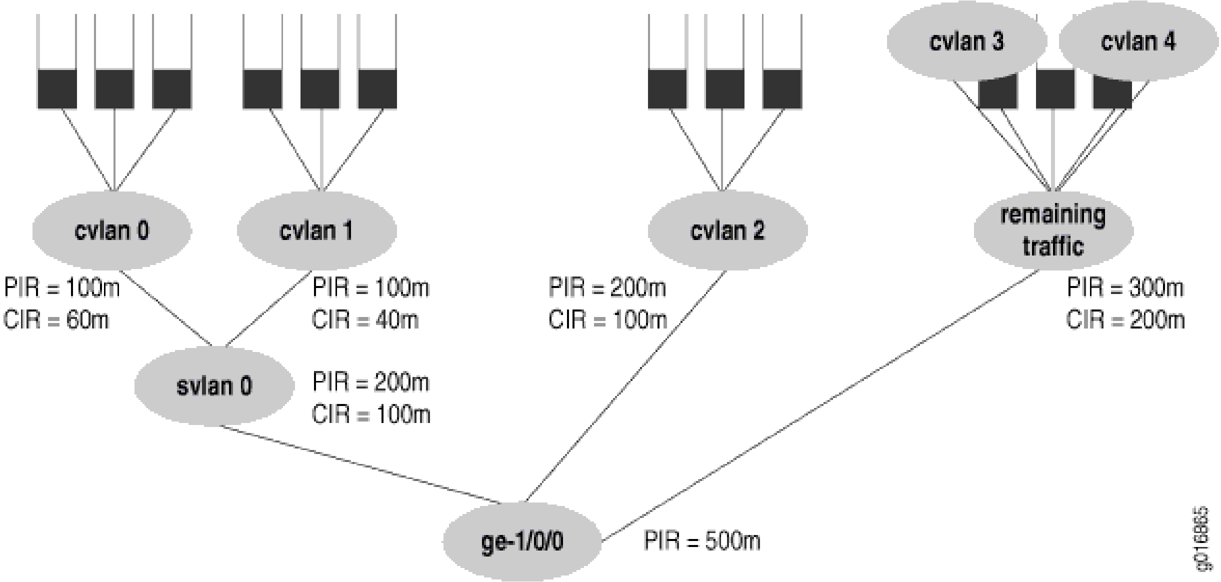
Figure 13: Example 1 Handling Remaining Traffic with no Explicit Traffic Control Profile


cvlan 3
cvlan 4
cvlan 0
cvlan 1
cvlan 2
remaining traffic
PIR = 100m
CIR = 60m
PIR = 100m
CIR = 40m
PIR = 200m
CIR = 100m
PIR = 300m
CIR = 200m
svlan 0
PIR = 200m
CIR = 100m
svlan 1
PIR = 400m
CIR = 300m
ge-1/0/0
PIR = 500m
g016857

Example 1 considers the case where C-VLANs 3 and 4 have no explicit traffic control profile, yet need to establish a shaping and guaranteed transmit rate for traffic heading for those C-VLANs. The solution is to add a traffic control profile to the `svlan1` interface set. This example builds on the example used in "Example: Configuring a Four-Level Scheduler Hierarchy" on page 267 and does not repeat all configuration details, only those at the S-VLAN level.

Next, consider Example 2 shown in Figure 14 on page 287.

Figure 14: Example 2 Handling Remaining Traffic with an Interface Set



In Example 2, `ge-1/0/0` has five logical interfaces (C-VLAN 0, 1, 2, 3 and 4), and S-VLAN 0, which are covered by the interface set:

- Scheduling for the interface set `svlan0` is specified by referencing an `output-traffic-control-profile` statement, which specifies the `guaranteed-rate`, `shaping-rate`, and `delay-buffer-rate` statement values for the interface set. In this example, the output traffic control profile called `tcp-svlan0` guarantees 100 Mbps and shapes the interface set `svlan0` to 200 Mbps.

- Scheduling and queuing for remaining traffic of `svlan0` is specified by referencing an `output-traffic-control-profile-remaining` statement, which references a `scheduler-map` statement that establishes queues for the remaining traffic. The specified traffic control profile can also configure guaranteed, shaping, and delay-buffer rates for the remaining traffic. In Example 2, `output-traffic-control-profile-remaining tcp-svlan0-rem` references `scheduler-map smap-svlan0-rem`, which calls for a best-effort queue for remaining traffic (that is, traffic on unit 3 and unit 4, which is not classified by the `svlan0` interface set). The example also specifies a `guaranteed-rate` of 200 Mbps and a `shaping-rate` of 300 Mbps for all remaining traffic.

- Scheduling and queuing for logical interface `ge-1/0/0 unit 1` is configured “traditionally” and uses an `output-traffic-control-profile` specified for that unit. In this example, `output-traffic-control-profile tcp-ifl1` specifies scheduling and queuing for `ge-1/0/0 unit 1`.

Configuration

IN THIS SECTION

- Controlling Remaining Traffic With No Explicit Traffic Control Profile | 288
- Controlling Remaining Traffic With An Interface Set | 290

This section contains the following topics:

Controlling Remaining Traffic With No Explicit Traffic Control Profile

CLI Quick Configuration

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

```
set class-of-service interfaces interface-set svlan0 output-traffic-control-profile tcp-svlan0
set class-of-service interfaces interface-set svlan1 output-traffic-control-profile tcp-svlan1
set class-of-service interfaces interface-set svlan1 output-traffic-control-profile-remaining
tcp-svlan1-remaining
set class-of-service traffic-control-profiles tcp-svlan1 shaping-rate 400m guaranteed-rate 300m
set class-of-service traffic-control-profiles tcp-svlan1-remaining shaping-rate 300m guaranteed-
rate 200m scheduler-map smap-remainder
```

Step-by-Step Procedure

To control remaining traffic with no explicit traffic control profile:

1. Set the logical interfaces for the S-VLANs.

```
[edit class-of-service interfaces]
user@host# set interface-set svlan0 output-traffic-control-profile tcp-svlan0
user@host# set interface-set svlan1 output-traffic-control-profile tcp-svlan1
user@host# set interface-set svlan1 output-traffic-control-profile-remaining tcp-svlan1-
remaining
```

2. Set the shaping and guaranteed transmit rates for traffic heading for those C-VLANs.

```
[edit class-of-service traffic-control-profiles]
user@host# set tcp-svlan1 shaping-rate 400m guaranteed-rate 300m
user@host# set tcp-svlan1-remaining shaping-rate 300m guaranteed-rate 200m scheduler-map smap-
remainder
```

Results

From configuration mode, confirm your configuration by entering the `show class-of-service interfaces` and `show class-of-service traffic-control-profiles` commands. If the output does not display the intended configuration, repeat the configuration instructions in this example to correct it.

```
[edit]
user@host# show class-of-service interfaces
interface-set svlan0 {
    output-traffic-control-profile tcp-svlan0;
}
interface-set svlan1 {
    output-traffic-control-profile tcp-svlan1;
    output-traffic-control-profile-remaining tcp-svlan1-remaining; # For all remaining traffic
}

[edit]
user@host# show class-of-service traffic-control-profiles
tcp-svlan1 {
    shaping-rate 400m;
    guaranteed-rate 300m;
}
tcp-svlan1-remaining {
    shaping-rate 300m;
    guaranteed-rate 200m;
    scheduler-map smap-remainder; # this smap is not shown in detail
}
```

If you are done configuring the device, enter `commit` from configuration mode.

Controlling Remaining Traffic With An Interface Set

CLI Quick Configuration

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

```
set class-of-service interfaces interface-set svlan0 output-traffic-control-profile tcp-svlan0
set class-of-service interfaces ge-1/0/0 output-traffic-control-profile-remaining tcp-svlan0-rem
unit 1output-traffic-control-profile tcp-ifl1
set class-of-service traffic-control-profiles tcp-svlan0 shaping-rate 200m guaranteed-rate 100m
set class-of-service traffic-control-profiles tcp-svlan0-rem shaping-rate 300m guaranteed-rate
200m scheduler-map smap-svlan0-rem
set class-of-service traffic-control-profiles tcp-ifl1 scheduler-map smap-ifl1
set class-of-service scheduler-maps smap-svlan0-rem forwarding-class best-effort scheduler-sched-
foo
set class-of-service scheduler-maps smap-ifl1 forwarding-class best-effort scheduler-sched-bar
set class-of-service scheduler-maps smap-ifl1 forwarding-class assured-forwarding scheduler-
sched-bar
```

Step-by-Step Procedure

To control remaining traffic with an interface set:

1. Set the interface set for the S-VLAN.

```
[edit class-of-service interfaces]
user@host# set interface-set svlan0 output-traffic-control-profile tcp-svlan0
user@host# set ge-1/0/0 output-traffic-control-profile-remaining tcp-svlan0-rem unit 1output-
traffic-control-profile tcp-ifl1
```

2. Set the traffic control profiles.

```
[edit class-of-service traffic-control-profiles]
user@host# set tcp-svlan0 shaping-rate 200m guaranteed-rate 100m
user@host# set tcp-svlan0-rem shaping-rate 300m guaranteed-rate 200m scheduler-map smap-
```

```
svlan0-rem
user@host# set tcp-ifl1 scheduler-map smap-ifl1
```

3. Set the scheduler map.

```
[edit class-of-service scheduler-maps]
user@host# set smap-svlan0-rem forwarding-class best-effort scheduler-sched-foo
user@host# set smap-ifl1 forwarding-class best-effort scheduler-sched-bar
user@host# set smap-ifl1 forwarding-class assured-forwarding scheduler-sched-bar
```

Results

From configuration mode, confirm your configuration by entering the `show class-of-service interfaces`, `show class-of-service traffic-control-profiles` , and `show class-of-service scheduler-maps` commands. If the output does not display the intended configuration, repeat the configuration instructions in this example to correct it. Example 2 does not include the `[edit interfaces]` configuration.

```
[edit]
user@host# show class-of-service interfaces
interface-set {
    svlan0 {
        output-traffic-control-profile tcp-svlan0; # Guarantee & shaper for svlan0
    }
}
ge-1/0/0 {
    output-traffic-control-profile-remaining tcp-svlan0-rem
        # Unit 3 and 4 are not explicitly configured, but captured by “remaining'
        unit 1 {
        output-traffic-control-profile tcp-ifl1; # Unit 1 be & ef queues
    }
}

[edit]
user@host# show class-of-service traffic-control-profiles
tcp-svlan0 {
    shaping-rate 200m;
    guaranteed-rate 100m;
}
tcp-svlan0-rem {
    shaping-rate 300m;
    guaranteed-rate 200m;
    scheduler-map smap-svlan0-rem; # This specifies queues for remaining traffic
}
tcp-ifl1 {
    scheduler-map smap-ifl1;
}

[edit]
user@host# show class-of-service scheduler-maps
smap-svlan0-rem {
    forwarding-class best-effort scheduler sched-foo;
}
smap-ifl1 {
    forwarding-class best-effort scheduler sched-bar;
    forwarding-class assured-forwarding scheduler sched-baz;
}
```

If you are done configuring the device, enter `commit` from configuration mode.

The configuration for the referenced schedulers is not given for this example.

Verification

IN THIS SECTION

- Verifying Remaining Traffic Control | 292

Verifying Remaining Traffic Control

Purpose

Verify that the remaining traffic is controlled properly.

Action

From operational mode, enter the following commands:

- `show class-of-service interfaces`
- `show class-of-service traffic-control-profiles`

- `show class-of-service scheduler-maps`

RELATED DOCUMENTATION

Understanding Hierarchical Schedulers | **262**

4

PART

Configuring Class of Service for IPv6

- Configuring Class of Service for IPv6 Traffic | **295**

CHAPTER 14

Configuring Class of Service for IPv6 Traffic

IN THIS CHAPTER

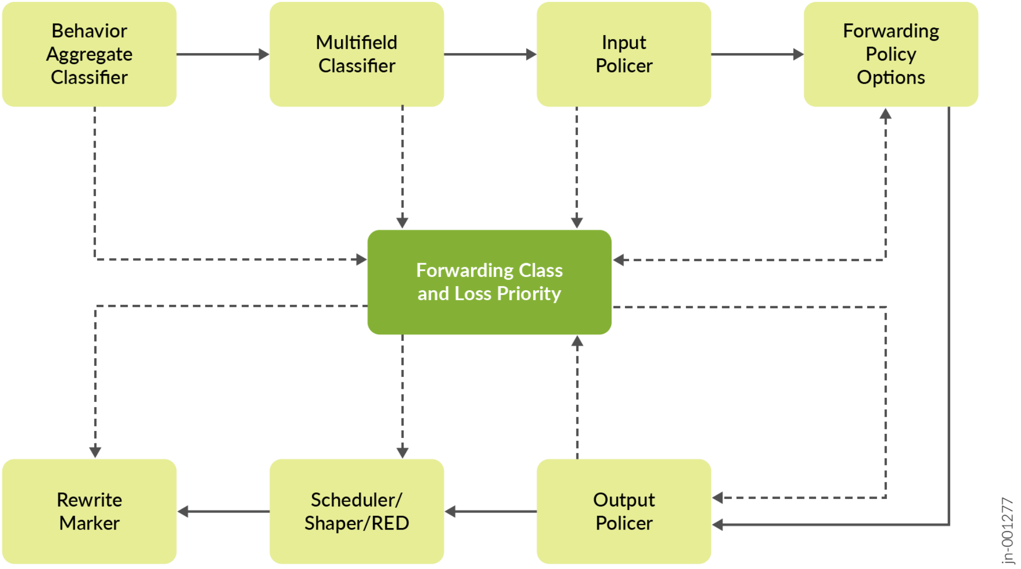
- CoS Functions for IPv6 Traffic Overview | **295**
- Understanding CoS with DSCP IPv6 BA Classifier | **297**
- Example: Configuring CoS with DSCP IPv6 BA Classifiers | **298**
- Understanding DSCP IPv6 Rewrite Rules | **302**
- Example: Configuring CoS with DSCP IPv6 Rewrite Rules | **303**

CoS Functions for IPv6 Traffic Overview

Class-of-service (CoS) processing for IPv6 traffic uses the IPv6 DiffServ code point (DSCP) value. The IPv6 DSCP value is the first six bits in the 8-bit Traffic Class field of the IPv6 header. The DSCP value is used to determine the behavior aggregate (BA) classification for the packet entering the network device. You use classifier rules to map the DSCP code points to a forwarding class and packet loss priority. You use *rewrite rules* to map the forwarding class and packet loss priority back to DSCP values on packets exiting the device.

Figure 15 on page 296 shows the components of the CoS features for Juniper Networks devices, illustrating the sequence in which they interact.

Figure 15: Packet Flow Through an SRX Series Firewall


Behavior Aggregate Classifier
Multifield Classifier
Input Policer
Forwarding Policy Options
Forwarding Class and Loss Priority
Rewrite Marker
Scheduler/ Shaper/RED
Output Policer
jn-001277

NOTE: Not all CoS features are supported on all devices.

- CoS components perform the following operations:

  BA classifier rules map DSCP code points to a forwarding class and loss priority. The forwarding class and loss priority determine the per-hop behavior of the packet throughout the system. The forwarding class associates a packet with an outbound transmission queue. Loss priority affects the scheduling of a packet without affecting the relative ordering of packets. BA classification is a simple way that “downstream” nodes can honor the CoS objectives that were encoded “upstream.”

  See "Example: Configuring CoS with DSCP IPv6 BA Classifiers" on page 298.

- Multifield classifier rules overwrite the initial forwarding class and loss priority determination read by the BA classifier rule. You typically use multifield classifier rules on nodes close to the content origin, where a packet might not have been encoded with the desired DSCP values in the headers. A multifield classifier rule assigns packets to a forwarding class and assigns a packet loss priority based on filters, such as source IP, destination IP, port, or application.

  See "Example: Configure and Apply a Firewall Filter for a Multifield Classifier" on page 68.

- Input policers meter traffic to see if traffic flow exceeds its service level. Policers might discard, change the forwarding class and loss priority, or set the packet loss priority bit of a packet. A packet for which the packet loss priority bit is set has an increased probability of being dropped during congestion.

- Scheduler maps are applied to interfaces and associate the outgoing packets with a scheduler and a forwarding class.

  The scheduler manages the output transmission queue, including:

  - Buffer size—Defines the period for which a packet is stored during congestion.
  - Scheduling priority and transmit rate—Determines the order in which a packet is transmitted.
  - Drop profile—Defines how aggressively to drop a packet that is using a particular scheduler.

  See "Example: Configuring Class-of-Service Schedulers on a Security Device" on page 133.

- Output policers meter traffic and might change the forwarding class and loss priority of a packet if a traffic flow exceeds its service level.

- Rewrite rules map forwarding class and packet loss priority to DSCP values. You typically use rewrite rules in conjunction with multifield classifier rules close to the content origin, or when the device is at the border of a network and must alter the code points to meet the policies of the targeted peer.

  See "Example: Configuring CoS with DSCP IPv6 Rewrite Rules" on page 303.

Only BA classification rules and rewrite rules require special consideration to support CoS for IPv6 traffic. The program logic for the other CoS features is not sensitive to differences between IPv4 and IPv6 traffic.

RELATED DOCUMENTATION

Understanding CoS with DSCP IPv6 BA Classifier | 297
Example: Configuring CoS with DSCP IPv6 BA Classifiers | 298
Understanding DSCP IPv6 Rewrite Rules | 302
Example: Configuring CoS with DSCP IPv6 Rewrite Rules | 303

Understanding CoS with DSCP IPv6 BA Classifier

RELATED DOCUMENTATION

Example: Configuring CoS with DSCP IPv6 BA Classifiers | 298
CoS Functions for IPv6 Traffic Overview | 295
Understanding DSCP IPv6 Rewrite Rules | 302

Example: Configuring CoS with DSCP IPv6 Rewrite Rules | 303

Example: Configuring CoS with DSCP IPv6 BA Classifiers

IN THIS SECTION

- Requirements | 298
- Overview | 298
- Configuration | 298
- Verification | 302

This example shows how to associate an interface with a default or user-defined DSCP IPv6 BA classifier.

Requirements

Before you begin, configure the ge-0/0/0 interface on the device for IPv6 and define your user-defined DSCP IPv6 classifier settings. See "Understanding CoS with DSCP IPv6 BA Classifier" on page 297.

Overview

In this example, you configure CoS and define forwarding classes. You create the behavior aggregate classifier for DiffServ CoS as dscp-ipv6-example and import the default DSCP IPv6 classifier.

You then specify the best-effort forwarding class as be-class, the expedited forwarding class as ef-class, the assured forwarding class as af-class, and the network control forwarding class as nc-class. Finally, you apply your user-defined classifier to interface ge-0/0/0.

Configuration

IN THIS SECTION

- Procedure | 299

Procedure

CLI Quick Configuration

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

```
set class-of-service forwarding-classes queue 0 be-class
set class-of-service forwarding-classes queue 1 ef-class
set class-of-service forwarding-classes queue 2 af-class
set class-of-service forwarding-classes queue 3 nc-class
set class-of-service classifiers dscp-ipv6 dscp-ipv6-example import default
set class-of-service classifiers dscp-ipv6 dscp-ipv6-example forwarding-class be-class loss-
priority high code-points 000001
set class-of-service classifiers dscp-ipv6 dscp-ipv6-example forwarding-class ef-class loss-
priority high code-points 101111
set class-of-service classifiers dscp-ipv6 dscp-ipv6-example forwarding-class af-class loss-
priority high code-points 001100
set class-of-service classifiers dscp-ipv6 dscp-ipv6-example forwarding-class nc-class loss-
priority high code-points 110001
set class-of-service interfaces ge-0/0/0 unit 0 classifiers dscp-ipv6 dscp-ipv6-example
```

Step-by-Step Procedure

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 Junos OS CLI User Guide.

To configure CoS with a user-defined DSCP IPv6 BA classifier:

1. Configure CoS.

   ```
   [edit]
   user@host# edit class-of-service
   ```

2. Define forwarding classes.

   ```
   [edit class-of-service]
   user@host# set forwarding-classes queue 0 be-class
   ```

```
user@host# set forwarding-classes queue 1 ef-class
user@host# set forwarding-classes queue 2 af-class
user@host# set forwarding-classes queue 3 nc-class
```

3. Create a behavior aggregate classifier for DiffServ CoS.

```
[edit class-of-service]
user@host# edit classifiers dscp-ipv6 dscp-ipv6-example
```

4. Import a DSCP IPv6 classifier.

```
[edit class-of-service classifiers dscp-ipv6 dscp-ipv6-example]
user@host# set import default
```

5. Specify a best-effort forwarding class classifier.

```
[edit class-of-service classifiers dscp-ipv6 dscp-ipv6-example]
user@host# set forwarding-class be-class loss-priority high code-points 000001
```

6. Specify an expedited forwarding class classifier.

```
[edit class-of-service classifiers dscp-ipv6 dscp-ipv6-example]
user@host# set forwarding-class ef-class loss-priority high code-points 101111
```

7. Specify an assured forwarding class classifier.

```
[edit class-of-service classifiers dscp-ipv6 dscp-ipv6-example]
user@host# set forwarding-class af-class loss-priority high code-points 001100
```

8. Specify a network control forwarding class classifier.

```
[edit class-of-service classifiers dscp-ipv6 dscp-ipv6-example]
user@host# set forwarding-class nc-class loss-priority high code-points 110001
```

9. Associate a user-defined classifier with an interface.

```
[edit class-of-service]
user@host# set interfaces ge-0/0/0 unit 0 classifiers dscp-ipv6 dscp-ipv6-example
```

Results

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

```
[edit]
user@host# show class-of-service
classifiers {
    dscp-ipv6 dscp-ipv6-example {
    import default;
    forwarding-class be-class {
        loss-priority high code-points 000001;
    }
    forwarding-class ef-class {
        loss-priority high code-points 101111;
    }
    forwarding-class af-class {
        loss-priority high code-points 001100;
    }
    forwarding-class nc-class {
        loss-priority high code-points 110001;
        }
    }
    }
    forwarding-classes {
        queue 0 be-class;
        queue 1 ef-class;
        queue 2 af-class;
        queue 3 nc-class;
    }
    interfaces {
        ge-0/0/0 {
        unit 0 {
        classifiers {
        dscp-ipv6 dscp-ipv6-example;
```

```
            }
        }
    }
}
```

If you are done configuring the device, enter **commit** from configuration mode.

Verification

IN THIS SECTION

- Verifying the CoS with DSCP IPv6 BA Classifier Configuration | 302

Verifying the CoS with DSCP IPv6 BA Classifier Configuration

Purpose

Verify that the user-defined DSCP IPv6 BA classifier is associated with an interface.

Action

From configuration mode, enter the **show class-of-service** command.

RELATED DOCUMENTATION

Understanding CoS with DSCP IPv6 BA Classifier | 297
CoS Functions for IPv6 Traffic Overview | 295
Understanding DSCP IPv6 Rewrite Rules | 302
Example: Configuring CoS with DSCP IPv6 Rewrite Rules | 303

Understanding DSCP IPv6 Rewrite Rules

After Junos OS CoS processing, a rewrite rule maps the forwarding class and loss priority after Junos OS CoS processing to a corresponding DSCP value specified in the rule. Typically, you use *rewrite rules* to alter the CoS values in outgoing packets to meet the requirements of the targeted peer.

You can use the CLI show command to display the configuration for the CoS classifiers. The following command shows the configuration of the default DSCP IPv6 rewrite rule:

```
user@host# show class-of-service rewrite-rule type dscp-ipv6
Rewrite rule: dscp-ipv6-default, Code point type: dscp-ipv6, Index: 32
  Forwarding class                    Loss priority       Code point
  best-effort                         low                 000000
  best-effort                         high                000000
  expedited-forwarding                low                 101110
  expedited-forwarding                high                101110
  assured-forwarding                  low                 001010
  assured-forwarding                  high                001100
  network-control                     low                 110000
  network-control                     high                111000
```

RELATED DOCUMENTATION

Example: Configuring CoS with DSCP IPv6 Rewrite Rules | 303

CoS Functions for IPv6 Traffic Overview | 295

Understanding CoS with DSCP IPv6 BA Classifier | 297

Example: Configuring CoS with DSCP IPv6 BA Classifiers | 298

Example: Configuring CoS with DSCP IPv6 Rewrite Rules

IN THIS SECTION

- Requirements | 304
- Overview | 304
- Configuration | 304
- Verification | 307

This example shows how to associate an interface with a default or user-defined DSCP IPv6 rewrite rule. Typically, you use rewrite rules to alter CoS values in outgoing packets to meet the requirements of the targeted peer.

Requirements

Before you begin, configure the ge-0/0/0 interface on the device for IPv6 and define your user-defined DSCP IPv6 rewrite rules.

Overview

In this example, you configure CoS and create a user-defined rewrite rule called rewrite-ipv6-dscps. You then specify rewrite rules for the best-effort forwarding class as be-class, the expedited forwarding class as ef-class, the assured forwarding class as af-class, and the network control forwarding class as nc-class. Finally, you associate interface ge-0/0/0 with the user-defined rule.

Configuration

IN THIS SECTION

- Procedure | 304

Procedure

CLI Quick Configuration

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

```
set class-of-service rewrite-rules dscp-ipv6 rewrite-ipv6-dscps
set class-of-service rewrite-rules dscp-ipv6 rewrite-ipv6-dscps forwarding-class be-class loss-
priority low code-point 000000
set class-of-service rewrite-rules dscp-ipv6 rewrite-ipv6-dscps forwarding-class be-class loss-
priority high code-point 000001
set class-of-service rewrite-rules dscp-ipv6 rewrite-ipv6-dscps forwarding-class ef-class loss-
priority low code-point 101110
set class-of-service rewrite-rules dscp-ipv6 rewrite-ipv6-dscps forwarding-class ef-class loss-
priority high code-point 101111
set class-of-service rewrite-rules dscp-ipv6 rewrite-ipv6-dscps forwarding-class af-class loss-
```

```
priority low code-point 001010
set class-of-service rewrite-rules dscp-ipv6 rewrite-ipv6-dscps forwarding-class af-class loss-
priority high code-point 001100
set class-of-service rewrite-rules dscp-ipv6 rewrite-ipv6-dscps forwarding-class nc-class loss-
priority low code-point 110000
set class-of-service rewrite-rules dscp-ipv6 rewrite-ipv6-dscps forwarding-class nc-class loss-
priority high code-point 110001
set class-of-service interfaces ge-0/0/0 unit 0 rewrite-rules dscp-ipv6 rewrite-ipv6-dscps
```

Step-by-Step Procedure

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 Junos OS CLI User Guide.

To configure a CoS with a user-defined DSCP IPv6 rewrite rule:

1. Configure CoS.

   ```
   [edit]
   user@host# edit class-of-service
   ```

2. Create a user-defined rewrite rule.

   ```
   [edit class-of-service]
   user@host# edit rewrite-rules dscp-ipv6 rewrite-ipv6-dscps
   ```

3. Specify rewrite rules for the best-effort forwarding class.

   ```
   [edit class-of-service rewrite-rules dscp-ipv6 rewrite-ipv6-dscps]
   user@host# set forwarding-class be-class loss-priority low code-point 000000
   user@host# set forwarding-class be-class loss-priority high code-point 000001
   ```

4. Specify rewrite rules for the expedited-forwarding forwarding class.

   ```
   [edit class-of-service rewrite-rules dscp-ipv6 rewrite-ipv6-dscps]
   user@host# set forwarding-class ef-class loss-priority low code-point 101110
   user@host# set forwarding-class ef-class loss-priority high code-point 101111
   ```

5. Specify rewrite rules for the assured-forwarding forwarding class.

```
[edit class-of-service rewrite-rules dscp-ipv6 rewrite-ipv6-dscps]
user@host# set forwarding-class af-class loss-priority low code-point 001010
user@host# set forwarding-class af-class loss-priority high code-point 001100
```

6. Specify rewrite rules for the network-control forwarding class.

```
[edit class-of-service rewrite-rules dscp-ipv6 rewrite-ipv6-dscps]
user@host# set forwarding-class nc-class loss-priority low code-point 110000
user@host# set forwarding-class nc-class loss-priority high code-point 110001
```

7. Associate an interface with a user-defined rule.

```
[edit class-of-service]
user@host# set interfaces ge-0/0/0 unit 0 rewrite-rules dscp-ipv6 rewrite-ipv6-dscps
```

Results

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

```
[edit]
user@host# show class-of-service
interfaces {
    ge-0/0/0 {
        unit 0 {
            rewrite-rules {
                dscp-ipv6 rewrite-ipv6-dscps;
            }
        }
    }
}
rewrite-rules {
    dscp-ipv6 rewrite-ipv6-dscps {
        forwarding-class be-class {
            loss-priority low code-point 000000;
            loss-priority high code-point 000001;
        }
        forwarding-class ef-class {
            loss-priority low code-point 101110;
            loss-priority high code-point 101111;
        }
        forwarding-class af-class {
            loss-priority low code-point 001010;
            loss-priority high code-point 001100;
        }
        forwarding-class nc-class {
            loss-priority low code-point 110000;
            loss-priority high code-point 110001;
        }
    }
}
```

If you are done configuring the device, enter **commit** from configuration mode.

Verification

IN THIS SECTION

- Verifying the CoS with DSCP IPv6 Rewrite Rule Configuration | 307

Verifying the CoS with DSCP IPv6 Rewrite Rule Configuration

Purpose

Verify that the user-defined CoS with DSCP IPv6 rewrite rule is associated with an interface.

Action

From configuration mode, enter the **show class-of-service** command.

RELATED DOCUMENTATION

Understanding DSCP IPv6 Rewrite Rules | 302
CoS Functions for IPv6 Traffic Overview | 295

Understanding CoS with DSCP IPv6 BA Classifier | **297**

Example: Configuring CoS with DSCP IPv6 BA Classifiers | **298**

5

PART

Configuring Class of Service for I/O Cards

- Configuring Class of Service for I/O Cards | **310**

CHAPTER 15

Configuring Class of Service for I/O Cards

IN THIS CHAPTER

- PIR-Only and CIR Mode Overview | **310**
- Understanding Priority Propagation | **312**
- Understanding IOC Hardware Properties | **315**
- Understanding IOC Map Queues | **317**
- WRED on the IOC Overview | **318**
- MDRR on the IOC Overview | **324**
- CoS Support on the SRX5000 Module Port Concentrator Overview | **326**
- Example: Configuring CoS on SRX5000 Firewalls with an MPC | **327**

PIR-Only and CIR Mode Overview

IN THIS SECTION

- PIR-only Mode | **310**
- CIR Mode | **311**

The actual behavior of many CoS parameters, especially the shaping rate and guaranteed rate, depends on whether the physical interface is operating in one of the following modes:

PIR-only Mode

In PIR-only (peak information rate) mode, one or more nodes perform shaping. The physical interface is in PIR-only mode if no child (or grandchild) node under the port has a guaranteed rate configured. The mode of the port is important because in PIR-only mode, the scheduling across the child nodes is in proportion to their shaping rates (PIRs) and not the guaranteed rates (CIRs). This can be important if the observed behavior is not what is anticipated.

In PIR-only mode, nodes cannot send if they are above the configured shaping rate. Table 47 on page 311 shows the mapping between the configured priority and the hardware priority for PIR-only.

Table 47: Internal Node Queue Priority for PIR-Only Mode

Configured Priority	Hardware Priority
Strict-high	0
High	0
Medium-high	1
Medium-low	1
Low	2

CIR Mode

In CIR (committed information rate) mode, one or more nodes applies a guaranteed rate and might perform shaping. A physical interface is in CIR mode if at least one child (or grandchild) node has a guaranteed rate configured. In addition, any child or grandchild node under the physical interface can have a shaping rate configured. Only the guaranteed rate matters. In CIR mode, nodes that do not have a guaranteed rate configured are assumed to have a very small guaranteed rate (queuing weight).

In CIR mode, the priority for each internal node depends on whether the highest active child node is above or below the guaranteed rate. Table 48 on page 311 shows the mapping between the highest active child's priority and the hardware priority below and above the guaranteed rate.

Table 48: Internal Node Queue Priority for CIR Mode

Configured Priority of Highest Active Child Node	Hardware Priority Below Guaranteed Rate	Hardware Priority Above Guaranteed Rate
Strict-high	0	0

Table 48: Internal Node Queue Priority for CIR Mode *(Continued)*

Configured Priority of Highest Active Child Node	Hardware Priority Below Guaranteed Rate	Hardware Priority Above Guaranteed Rate
High	0	3
Medium-high	1	3
Medium-low	1	3
Low	2	3

RELATED DOCUMENTATION

Understanding Priority Propagation | 312

Understanding IOC Hardware Properties | 315

Understanding IOC Map Queues | 317

WRED on the IOC Overview | 318

MDRR on the IOC Overview | 324

Understanding Priority Propagation

SRX firewalls with input/output cards (IOCs) perform priority propagation. Priority propagation is useful for mixed traffic environments when, for example, you want to make sure that the voice traffic of one customer does not suffer from the data traffic of another customer. Nodes and queues are always serviced in the order of their priority. The priority of a queue is decided by configuration (the default priority is low) in the scheduler. However, not all elements of hierarchical schedulers have direct priorities configured. Internal nodes, for example, must determine their priority in other ways.

The priority of any internal node is decided as follows:

- By the highest priority of an active child (interface sets only take the highest priority of their active children
- Whether the node is above its configured guaranteed rate (CIR) or not (this is relevant only if the physical interface is in CIR mode)

Each queue has a configured priority and a hardware priority. Table 49 on page 313 shows the usual mapping between the configured priority and the hardware priority.

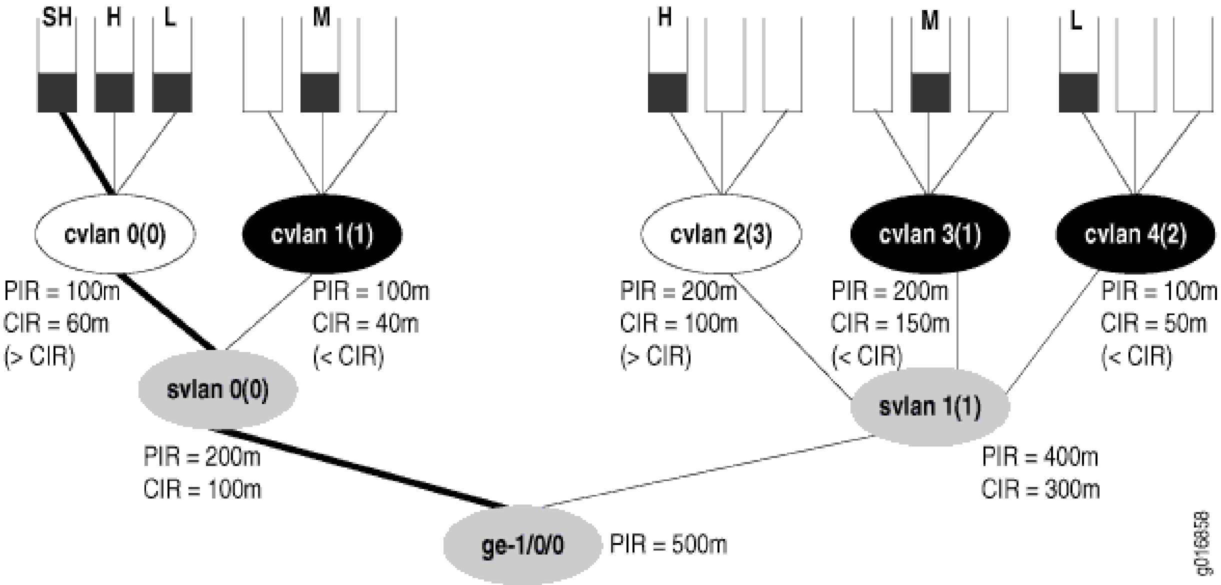
Table 49: Queue Priority

Configured Priority	Hardware Priority
Strict-high	0
High	0
Medium-high	1
Medium-low	1
Low	2

Figure 16 on page 314 shows a physical interface with hierarchical schedulers configured. The configured priorities are shown for each queue at the top of the figure. The hardware priorities for each node are shown in parentheses. Each node also shows any configured shaping rate (PIR) or guaranteed rate (CIR) and whether or not the queues are above or below the CIR. The nodes are shown in one of the following three states:

- Above the CIR (clear)
- Below the CIR (dark)
- Condition where the CIR does not matter (gray)

Figure 16: Hierarchical Schedulers and Priorities


SH
H
L
M
H
M
L
cvlan 0(0)
cvlan 1(1)
cvlan 2(3)
cvlan 3(1)
cvlan 4(2)
PIR = 100m
CIR = 60m
(> CIR)
PIR = 100m
CIR = 40m
(< CIR)
PIR = 200m
CIR = 100m
(> CIR)
PIR = 200m
CIR = 150m
(< CIR)
PIR = 100m
CIR = 50m
(< CIR)
svlan 0(0)
svlan 1(1)
PIR = 200m
CIR = 100m
PIR = 400m
CIR = 300m
ge-1/0/0
PIR = 500m
g016858

In Figure 16 on page 314, the strict high queue for C-VLAN 0 (cvlan 0) receives service first, even though the C-VLAN is above the configured CIR. Once that queue has been drained, and the priority of the node has become 3 instead of 0 (because of the lack of strict-high traffic), the system moves on to the medium queues (cvlan 1 and cvlan 3), draining them in a round-robin fashion where empty queues lose their hardware priority. The low queue on cvlan 4 (priority 2) is sent next because that mode is below the CIR. Then, the high queues on cvlan 0 and cvlan2 (both now with priority 3) are drained in a round-robin fashion, and finally the low queue on cvlan 0 is drained (because svlan 0 has a priority of 3).

RELATED DOCUMENTATION

- PIR-Only and CIR Mode Overview | 310
- Understanding IOC Hardware Properties | 315
- Understanding IOC Map Queues | 317
- WRED on the IOC Overview | 318
- MDRR on the IOC Overview | 324

Understanding IOC Hardware Properties

IN THIS SECTION

- Platform-Specific IOC Behavior | **316**

On IOCs, you can configure 15 VLAN sets per Gigabit Ethernet (40x1GE IOC) port and 255 VLAN sets per 10-Gigabit Ethernet (4x10GE IOC) port. The IOC performs priority propagation from one hierarchy level to another, and drop statistics are available on the IOC per color per queue instead of just per queue.

Table 50 on page 315 compares the major properties of the Packet Forwarding Engine within the IOC.

Table 50: Packet Forwarding Engine Properties within 40x1GE IOC and 4x10GE IOC

Feature	PFE Within 40x1GE IOC and 4x10GE IOC
Number of usable queues	16,000
Number of shaped logical interfaces	2,000 with 8 queues each, or 4,000 with 4 queues each.
Number of hardware priorities	4
Priority propagation	Yes
Dynamic mapping	Yes: schedulers per port are not fixed.
Drop statistics	Per queue per color (PLP high, low)

Additionally, the IOC features also support hierarchical weighted random early detection (WRED).

The IOC supports the following hierarchical scheduler characteristics:

- Shaping at the physical interface level
- Shaping and scheduling at the service VLAN interface set level
- Shaping and scheduling at the customer VLAN *logical interface* level

- Scheduling at the queue level

The IOC supports the following features for scalability:

- 16,000 queues per PFE
- 4 PFEs per IOC
  - 4000 schedulers at logical interface level (level 3) with 4 queues each
  - 2000 schedulers at logical interface level (level 3) with 8 queues each
- 255 schedulers at the interface set level (level 2) per 1-port PFE on a 10-Gigabit Ethernet IOC (4x10GE IOC)
- 15 schedulers at the interface set level (level 2) per 10-port PFE on a 1-Gigabit Ethernet IOC (40x1GE IOC)
- About 400 milliseconds of buffer delay (this varies by packet size and if large buffers are enabled)
- 4 levels of priority (strict-high, high, medium, and low)

NOTE: The `exact` option for a `transmit-rate` (`transmit-rate rate exact`) is not supported on the IOCs on SRX Series Firewalls.

NOTE: The above information is mostly for IOC1 cards. For MPC (IOC2), MPC3 (IOC3), and IOC4 cards (which use a subset of the CoS features available on IOC1), you can configure IEEE 802.1p classifiers, IEEE 802.1p rewrites, eight priority queues, and schedulers. After configuration, the classifiers and rewrites can be applied to logical interfaces, and queues and schedulers can be applied to physical interfaces.

- Due to hardware limitation, per-unit-scheduler or hierarchical-scheduler is not supported. Only the default mode is supported for egress scheduling and queuing.
- When an SPU is too busy to process every ingress packets from NG-IOCs, some high priority packets - for example, voice packets - may be delayed or dropped.

Platform-Specific IOC Behavior

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

Table 51: Platform-Specific IOC Behavior

Platform	Difference
SRX5600 and SRX5800	• On SRX5600 and SRX5800 firewalls, two IOCs (40x1GE IOC and 4x10GE IOC) are supported on which you can configure schedulers and queues. • SRX5600 and SRX5800 firewalls with IOCs have Packet Forwarding Engines that can support up to 512 MB of frame memory, and packets are stored in 512-byte frames.

RELATED DOCUMENTATION

PIR-Only and CIR Mode Overview | 310
Understanding Priority Propagation | 312
Understanding IOC Map Queues | 317
WRED on the IOC Overview | 318
MDRR on the IOC Overview | 324

Understanding IOC Map Queues

The manner in which the IOC maps a queue to a scheduler depends on whether 8 queues or 4 queues are configured. By default, a scheduler at level 3 has 4 queues. Level 3 scheduler X controls queue X*4 to X*4+3, so that scheduler 100 (for example) controls queues 400 to 403. However, when 8 queues per scheduler are enabled, the odd-numbered schedulers are disabled, allowing twice the number of queues per subscriber as before. With 8 queues, level 3 scheduler X controls queue X*4 to X*4+7, so that scheduler 100 (for example) now controls queues 400 to 407.

You configure the `max-queues-per-interface` statement to set the number of queues at 4 or 8 at the FPC level of the hierarchy. Changing this statement will result in a restart of the FPC.

The IOC maps level 3 (customer VLAN) schedulers in groups to level 2 (service VLAN) schedulers. Sixteen contiguous level 3 schedulers are mapped to level 2 when 4 queues are enabled, and 8 contiguous level 3 schedulers are mapped to level 2 when 8 queues are enabled. All the schedulers in the group should use the same queue priority mapping. For example, if the queue priorities of one scheduler are high, medium, and low, all members of the group should have the same queue priority.

Groups at level 3 to level 2 can be mapped at any time. However, a group at level 3 can only be unmapped from a level 2 scheduler, and only if all the schedulers in the group are free. Once unmapped, a level 3 group can be remapped to any level 2 scheduler. There is no restriction on the number of level 3 groups that can be mapped to a particular level 2 scheduler. There can be 256 level 3 groups, but fragmentation of the scheduler space can reduce the number of schedulers available. In other words, there are scheduler allocation patterns that might fail even though there are free schedulers.

In contrast to level 3 to level 2 mapping, the IOC maps level 2 (service VLAN) schedulers in a fixed mode to level 1 (physical interface) schedulers. On 40-port Gigabit Ethernet IOCs, there are 16 level 1 schedulers, and 10 of these are used for the physical interfaces. There are 256 level 2 schedulers, or 16 per level 1 schedulers. A level 1 scheduler uses level schedulers X*16 through X*16+15. Therefore level 1 scheduler 0 uses level 2 schedulers 0 through 15, level 1 scheduler 1 uses level 2 schedulers 16 through 31, and so on. On 4-port 10-Gigabit Ethernet PICs, there is one level 1 scheduler for the physical interface, and 256 level 2 schedulers are mapped to the single level 1 scheduler.

The maximum number of level 3 (customer VLAN) schedulers that can be used is 4076 (4 queues) or 2028 (8 queues) for the 10-port Gigabit Ethernet Packet Forwarding Engine and 4094 (4 queues) or 2046 (8 queues) for the 10-Gigabit Ethernet Packet Forwarding Engine.

RELATED DOCUMENTATION

PIR-Only and CIR Mode Overview | 310

Understanding Priority Propagation | 312

Understanding IOC Hardware Properties | 315

WRED on the IOC Overview | 318

MDRR on the IOC Overview | 324

show class-of-service spu-queue statistics

WRED on the IOC Overview

IN THIS SECTION

- Shapers at the Logical Interface Level (Level 3) | 320
- Shapers at the Interface Set Level (Level 2) | 322
- Shapers at the Port Level (Level 1) | 322

Shaping to drop out-of-profile traffic is done on the IOC at all levels except the queue level. However, weighed random early discard (WRED) is done at the queue level with much the same result. With WRED, the decision to drop or send the packet is made before the packet is placed in the queue.

WRED shaping on the IOC involves two levels. The probabilistic drop region establishes a minimum and a maximum queue depth. Below the minimum queue depth, the drop probability is 0 (send). Above the maximum level, the drop probability is 100 (certainty).

There are four drop profiles associated with each queue. These correspond to each of four loss priorities (low, medium-low, medium-high, and high). Sixty-four sets of four drop profiles are available (32 for ingress and 32 for egress). In addition, there are eight WRED scaling profiles in each direction.

An example of an IOC drop profile for expedited forwarding traffic is as follows:

```
[edit class-of-service drop-profiles]
drop-ef {
    fill-level 20 drop-probability 0; # Minimum Q depth
    fill-level 100 drop-probability 100; # Maximum Q depth
}
```

NOTE: You can specify only two fill levels for the IOC.

You can configure the **interpolate** statement, but only two fill levels are used. The **delay-buffer-rate** statement in the traffic control profile determines the maximum queue size. This delay buffer rate is converted to packet delay buffers, where one buffer is equal to 512 bytes. For example, at 10 Mbps, the IOC will allocate 610 delay buffers when the delay buffer rate is set to 250 milliseconds. The WRED threshold values are specified in terms of absolute buffer values.

The WRED scaling factor multiples all WRED thresholds (both minimum and maximum) by the value specified. There are eight values in all: 1, 2, 4, 8, 16, 32, 64, and 128. The WRED scaling factor is chosen to best match the user-configured drop profiles. This is done because the hardware supports only certain values of thresholds (all values must be a multiple of 16). So if the configured value of a threshold is 500 (for example), the multiple of 16 is 256 and the scaling factor applied is 2, making the value 512, which allows the value of 500 to be used. If the configured value of a threshold is 1500, the multiple of 16 is 752 and the scaling factor applied is 2, making the value 1504, which allows the value of 1500 to be used.

Hierarchical RED is used to support the oversubscription of the delay buffers (WRED is configured only at the queue, physical interface, and PIC levels). Hierarchical RED works with WRED as follows:

- If any level accepts the packet (the queue depth is less than the minimum buffer levels), this level accepts the packet.

- If any level probabilistically drops the packet, then this level drops the packet.

However, these rules might lead to the accepting of packets under loaded conditions that might otherwise have been dropped. In other words, the *logical interface* will accept packets if the physical interface is not congested.

Because of the limits placed on shaping thresholds used in the hierarchy, there is a granularity associated with the IOCs. The shaper accuracies differ at various levels of the hierarchy:

- Level 3
- Level 2
- Level 1

Shapers at the logical interface level (level 3) are more accurate than shapers at the interface set level (level 2) or at the port level (level 1).

This section contains the following topics:

Shapers at the Logical Interface Level (Level 3)

Because of the limits placed on shaping thresholds used in the hierarchy, there is a granularity associated with the IOCs. The shaper accuracies differ at various levels of the hierarchy, with shapers at the logical interface level (level 3) being more accurate than shapers at the interface set level (level 2) or at the port level (level 1). Table 52 on page 320 shows the accuracy of the logical interface shaper at various speeds for Ethernet ports operating at 1 Gbps.

Table 52: Shaper Accuracy of 1-Gbps Ethernet at the Logical Interface Level

Range of Logical Interface Shaper	Step Granularity
Up to 4.096 Mbps	16 Kbps
4.096 to 8.192 Mbps	32 Kbps
8.192 to 16.384 Mbps	64 Kbps
16.384 to 32.768 Mbps	128 Kbps
32.768 to 65.535 Mbps	256 Kbps

Table 52: Shaper Accuracy of 1-Gbps Ethernet at the Logical Interface Level *(Continued)*

Range of Logical Interface Shaper	Step Granularity
65.535 to 131.072 Mbps	512 Kbps
131.072 to 262.144 Mbps	1024 Kbps
262.144 to 1 Gbps	4096 Kbps

Table 53 on page 321 shows the accuracy of the logical interface shaper at various speeds for Ethernet ports operating at 10 Gbps.

Table 53: Shaper Accuracy of 10-Gbps Ethernet at the Logical Interface Level

Range of Logical Interface Shaper	Step Granularity
Up to 10.24 Mbps	40 Kbps
10.24 to 20.48 Mbps	80 Kbps
10.48 to 40.96 Mbps	160 Kbps
40.96 to 81.92 Mbps	320 Kbps
81.92 to 163.84 Mbps	640 Kbps
163.84 to 327.68 Mbps	1280 Kbps
327.68 to 655.36 Mbps	2560 Kbps
655.36 to 2611.2 Mbps	10240 Kbps
2611.2 to 5222.4 Mbps	20480 Kbps
5222.4 to 10 Gbps	40960 Kbps

Shapers at the Interface Set Level (Level 2)

Table 54 on page 322 shows the accuracy of the interface set shaper at various speeds for Ethernet ports operating at 1 Gbps.

Table 54: Shaper Accuracy of 1-Gbps Ethernet at the Interface Set Level

Range of Interface Set Shaper	Step Granularity
Up to 20.48 Mbps	80 Kbps
20.48 Mbps to 81.92 Mbps	320 Kbps
81.92 Mbps to 327.68 Mbps	1.28 Mbps
327.68 Mbps to 1 Gbps	20.48 Mbps

Table 55 on page 322 shows the accuracy of the interface set shaper at various speeds for Ethernet ports operating at 10 Gbps.

Table 55: Shaper Accuracy of 10-Gbps Ethernet at the Interface Set Level

Range of Interface Set Shaper	Step Granularity
Up to 128 Mbps	500 Kbps
128 Mbps to 512 Mbps	2 Mbps
512 Mbps to 2.048 Gbps	8 Mbps
2.048 Gbps to 10 Gbps	128 Mbps

Shapers at the Port Level (Level 1)

Table 56 on page 323 shows the accuracy of the physical port shaper at various speeds for Ethernet ports operating at 1 Gbps.

Table 56: Shaper Accuracy of 1-Gbps Ethernet at the Physical Port Level

Range of Physical Port Shaper	Step Granularity
Up to 64 Mbps	250 Kbps
64 Mbps to 256 Mbps	1 Mbps
256 Mbps to 1 Gbps	4 Mbps

Table 57 on page 323 shows the accuracy of the physical port shaper at various speeds for Ethernet ports operating at 10 Gbps.

Table 57: Shaper Accuracy of 10-Gbps Ethernet at the Physical Port Level

Range of Physical Port Shaper	Step Granularity
Up to 640 Mbps	2.5 Mbps
640 Mbps to 2.56 Gbps	10 Mbps
2.56 Gbps to 10 Gbps	40 Mbps

RELATED DOCUMENTATION

- PIR-Only and CIR Mode Overview | **310**
- Understanding Priority Propagation | **312**
- Understanding IOC Hardware Properties | **315**
- Understanding IOC Map Queues | **317**
- MDRR on the IOC Overview | **324**

MDRR on the IOC Overview

The guaranteed rate CIR at the interface set level is implemented by using modified deficit round-robin (MDRR). The IOC hardware provides four levels of strict priority. There is no restriction on the number of queues for each priority. MDRR is used among queues of the same priority. Each queue has one priority when it is under the guaranteed rate and another priority when it is over the guaranteed rate but still under the shaping rate PIR. The IOC hardware implements the priorities with 256 service profiles. Each service profile assigns eight priorities for eight queues. One set is for logical interfaces under the guaranteed rate and another set is for logical interfaces over the guaranteed rate but under the shaping rate. Each service profile is associated with a group of 16 level 3 schedulers, so there is a unique service profile available for all 256 groups at level 3, giving 4,096 logical interfaces.

Junos OS provides three priorities for traffic under the guaranteed rate and one reserved priority for traffic over the guaranteed rate that is not configurable. Junos OS provides three priorities when there is no guaranteed rate configured on any *logical interface*.

Table 58 on page 324 shows the relationship between Junos OS priorities and the IOC hardware priorities below and above the guaranteed rate CIR.

Table 58: Junos Priorities Mapped to IOC Hardware Priorities

Junos OS Priority	IOC Hardware Priority Below Guaranteed Rate	IOC Hardware Priority Above Guaranteed Rate
Strict-high	High	High
High	High	Low
Medium-high	Medium-high	Low
Medium-low	Medium-high	Low
Low	Medium-low	Low

The Junos OS parameters are set in the scheduler map:

```
[edit class-of-service schedulers]
best-effort-scheduler {
    transmit-rate percent 30; # if no shaping rate
    buffer-size percent 30;
    priority high;
}
expedited-forwarding-scheduler {
    transmit-rate percent 40; # if no shaping rate
    buffer-size percent 40;
    priority strict-high;
}
```

NOTE: The use of both a shaping rate and a guaranteed rate at the interface set level (level 2) is not supported.

MDRR is provided at three levels of the scheduler hierarchy of the IOC with a granularity of 1 through 255. There are 64 MDRR profiles at the queue level, 16 at the interface set level, and 32 at the physical interface level.

Queue transmit rates are used for queue-level MDRR profile weight calculation. The queue MDRR weight is calculated differently based on the mode set for sharing excess bandwidth. If you configure the **equal** option for excess bandwidth, then the queue MDRR weight is calculated as:

Queue weight = (255 * Transmit-rate-percentage) / 100

If you configure the **proportional** option for excess bandwidth, which is the default, then the queue MDRR weight is calculated as:

Queue weight = Queue-transmit-rate / Queue-base-rate, where

Queue-transmit-rate = (Logical-interface-rate * Transmit-rate-percentage) / 100, and

Queue-base-rate = Excess-bandwidth-proportional-rate / 255

To configure the way that the IOC should handle excess bandwidth, configure the **excess-bandwidth-share** statement at the [edit interface-set *interface-set-name*] hierarchy level. By default, the excess bandwidth is set to **proportional** with a default value of 32.64 Mbps. In this mode, the excess bandwidth is shared in the ratio of the logical interface shaping rates. If set to **equal**, the excess bandwidth is shared equally among the logical interfaces.

The following example sets the excess bandwidth sharing to proportional at a rate of 100 Mbps with a shaping rate of 80 Mbps:

```
[edit interface-set example-interface-set]
excess-bandwidth-share proportional 100m;
output-traffic-control-profile PIR-80Mbps;
```

Shaping rates established at the logical interface level are used to calculate the MDRR weights used at the interface set level. The 16 MDRR profiles are set to initial values, and the closest profile with rounded values is chosen. By default, the physical port MDRR weights are preset to the full bandwidth on the interface.

RELATED DOCUMENTATION

PIR-Only and CIR Mode Overview | 310

Understanding Priority Propagation | 312

Understanding IOC Hardware Properties | 315

Understanding IOC Map Queues | 317

WRED on the IOC Overview | 318

CoS Support on the SRX5000 Module Port Concentrator Overview

The SRX5000 Module Port Concentrator (SRX5K-MPC) for the SRX5600 and SRX5800 uses the Trio chipset-based queuing model, which supports class of service (CoS) characteristics that are optimized compared to CoS characteristics supported by the standard queuing model. These CoS features enable SRX5600 and SRX5800 firewalls to achieve end-to-end quality of service and protect the network using various security functions.

CoS features on the SRX5600 and SRX5800 firewalls provide differentiated services to traffic in addition to the best-effort packet processing. The main CoS features include classification, CoS field rewriting, queuing, scheduling, and traffic shaping.

When a network experiences congestion and delay, you can use the CoS features to classify packets; assign them with different levels of packet loss priority, delay, and throughput; and mark their CoS-related fields defined in Layer 2 and Layer 3 headers.

The MPC supports the following CoS features:

- BA classifier based on IEEE 802.1p for packet classification (Layer 2 headers) for priority bits of ingress packets
- Rewrite rule based on IEEE 802.1p for priority bits of egress packets

> **NOTE**: You can configure up to 32 IEEE 802.1p rewriters on each SRX5K-MPC on the SRX5600 and SRX5800 firewalls.

- Eight priority queues per port with configurable schedulers at the egress physical interface

By default, the MPC supports eight queues. You can use the following CLI statement to change that setting to four queues:

```
set chassis fpc fpc-number pic pic-number max-queues-per-interface 4
```

Changing to four-queue mode limits that number of configurable queues to four on the MPC. This does not have any effect on the performance.

The CoS features on the MPC have the following limitations:

- On the MPC, the per-unit-scheduler or the hierarchical-scheduler is not supported. For egress scheduling and queuing, only the default mode is supported.
- When an SPU is too busy to process every ingress packet from the MPC, some high-priority packets, such as voice packets, might be delayed or dropped by the SRX5600 or SRX5800.

NOTE: The total number of classifiers supported on a Services Processing Unit (SPU) is 79. Three classifiers are installed on the SPU as default classifiers in the Layer 3 mode, independent of any CoS configuration, which leaves 76 classifiers that can be configured using the CoS CLI commands. The default classifiers number might vary in future releases or in different modes. You can verify the number of default classifiers installed on the SPU to determine how many classifiers can be configured using the CoS CLI commands.

RELATED DOCUMENTATION

Example: Configuring CoS on SRX5000 Firewalls with an MPC | **327**

Example: Configuring CoS on SRX5000 Firewalls with an MPC

IN THIS SECTION

- Requirements | **328**
- Overview | **328**
- Configuration | **330**
- Verification | **338**

This example shows how to configure CoS on an SRX5000 line firewall with an MPC.

Requirements

This example uses the following hardware and software components:

- SRX5600 with an SRX5K-MPC
- Any supported Junos OS Release

Before you begin:

- Understand CoS. See "Understanding Class of Service" on page 2.
- Understand chassis cluster configuration. See *Example: Configuring an Active/Passive Chassis Cluster on SRX5800 Devices*.
- Understand chassis cluster redundant interface configuration. See *Example: Configuring Chassis Cluster Redundant Ethernet Interfaces*.

No special configuration beyond device initialization is required before configuring this feature.

Overview

In this example, you create a behavior aggregate (BA) classifier to classify traffic based on the IEEE 802.1p value of the packet and assign forwarding-class priority queue to the traffic. You then configure the scheduler map and set the priority for the traffic.

By default, the SRX5K-MPC supports eight queues. In this example, you are configuring eight queues.

You apply the BA classifier to the input interface and apply the scheduler map to the output interface.

Table 59 on page 328 and Table 60 on page 329 show forwarding class details with priority, assigned queue numbers, and allocated queue buffers used in this example.

Table 59: Forwarding Class Samples

Forwarding Class	Queue Number
BE	0
SIG	1
AF	2

Table 59: Forwarding Class Samples *(Continued)*

Forwarding Class	Queue Number
Bronze-class	3
Silver-class	4
Gold-class	5
Control	6
VOIP	7

Table 60: Scheduler Samples

Scheduler	For CoS Traffic Type	Assigned Priority	Allocated Portion of Queue Buffer (Transmit Rate)
s-be	0	low	15
s-sig	1	low	15
s-af	2	medium-low	20
s-bronze	3	medium-low	20
s-silver	4	medium-high	10
s-gold	5	medium-high	10
s-nc	6	high	5
s-voip	7	high	5

Configuration

IN THIS SECTION

- Procedure | 330

Procedure

CLI Quick Configuration

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

```
set class-of-service classifiers ieee-802.1 c802 forwarding-class BE loss-priority low code-
points 000
set class-of-service classifiers ieee-802.1 c802 forwarding-class SIG loss-priority low code-
points 001
set class-of-service classifiers ieee-802.1 c802 forwarding-class AF loss-priority low code-
points 010
set class-of-service classifiers ieee-802.1 c802 forwarding-class Bronze-Class loss-priority low
code-points 011
set class-of-service classifiers ieee-802.1 c802 forwarding-class Silver-Class loss-priority low
code-points 100
set class-of-service classifiers ieee-802.1 c802 forwarding-class Gold-Class loss-priority low
code-points 101
set class-of-service classifiers ieee-802.1 c802 forwarding-class Central loss-priority low code-
points 110
set class-of-service classifiers ieee-802.1 c802 forwarding-class VOIP loss-priority low code-
points 111
set class-of-service forwarding-classes class BE queue-num 0
set class-of-service forwarding-classes class SIG queue-num 1
set class-of-service forwarding-classes class AF queue-num 2
set class-of-service forwarding-classes class Bronze-Class queue-num 3
set class-of-service forwarding-classes class Silver-Class queue-num 4
set class-of-service forwarding-classes class Gold-Class queue-num 5
set class-of-service forwarding-classes class Control queue-num 6
set class-of-service forwarding-classes class VOIP queue-num 7
set class-of-service scheduler-maps test forwarding-class BE scheduler s-be
set class-of-service scheduler-maps test forwarding-class SIG scheduler s-sig
set class-of-service scheduler-maps test forwarding-class AF scheduler s-af
set class-of-service scheduler-maps test forwarding-class Bronze-Class scheduler s-bronze
set class-of-service scheduler-maps test forwarding-class Silver-Class scheduler s-silver
set class-of-service scheduler-maps test forwarding-class Gold-Class scheduler s-gold
set class-of-service scheduler-maps test forwarding-class Control scheduler s-nc
set class-of-service scheduler-maps test forwarding-class VOIP scheduler s-voip
set class-of-service rewrite-rules ieee-802.1 rw802 forwarding-class BE loss-priority low code-point 000
set class-of-service rewrite-rules ieee-802.1 rw802 forwarding-class SIG loss-priority low code-point 001
set class-of-service rewrite-rules ieee-802.1 rw802 forwarding-class AF loss-priority low code-point 010
set class-of-service rewrite-rules ieee-802.1 rw802 forwarding-class Bronze-Class loss-priority low code-point 011
set class-of-service rewrite-rules ieee-802.1 rw802 forwarding-class Silver-Class loss-priority low code-point 100
set class-of-service rewrite-rules ieee-802.1 rw802 forwarding-class Gold-Class loss-priority low code-point 101
set class-of-service rewrite-rules ieee-802.1 rw802 forwarding-class Control loss-priority low code-point 110
set class-of-service rewrite-rules ieee-802.1 rw802 forwarding-class VOIP loss-priority low code-point 111
set class-of-service schedulers s-be transmit-rate percent 15
set class-of-service schedulers s-be priority low
set class-of-service schedulers s-sig transmit-rate percent 15
set class-of-service schedulers s-sig priority low
set class-of-service schedulers s-af transmit-rate percent 20
set class-of-service schedulers s-af priority medium-low
set class-of-service schedulers s-bronze transmit-rate percent 20
set class-of-service schedulers s-bronze priority medium-low
set class-of-service schedulers s-silver transmit-rate percent 10
set class-of-service schedulers s-silver priority medium-high
set class-of-service schedulers s-gold transmit-rate percent 10
set class-of-service schedulers s-gold priority medium-high
set class-of-service schedulers s-nc transmit-rate percent 5
set class-of-service schedulers s-nc priority high
set class-of-service schedulers s-voip transmit-rate percent 5
set class-of-service schedulers s-voip priority high
set class-of-service interfaces reth0 unit 0 classifiers ieee-802.1 c802
set class-of-service interfaces reth0 unit 0 rewrite-rules ieee-802.1 rw802
set class-of-service interfaces reth0 scheduler-map test
set class-of-service interfaces reth0 shaping-rate 1g
```

Step-by-Step Procedure

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 Junos OS CLI User Guide.

To configure forwarding classes:

1. Configure a classifier.

   ```
   [edit class-of-service]
   user@host# set classifiers ieee-802.1 c802 forwarding-class BE loss-priority low code-points
   000
   user@host# set classifiers ieee-802.1 c802 forwarding-class SIG loss-priority low code-points
   001
   user@host# set classifiers ieee-802.1 c802 forwarding-class AF loss-priority low code-points
   010
   user@host# set classifiers ieee-802.1 c802 forwarding-class Bronze-Class loss-priority low
   code-points 011
   user@host# set classifiers ieee-802.1 c802 forwarding-class Silver-Class loss-priority low
   code-points 100
   user@host# set classifiers ieee-802.1 c802 forwarding-class Gold-Class loss-priority low code-
   points 101
   user@host# set classifiers ieee-802.1 c802 forwarding-class Central loss-priority low code-
   points 110
   user@host# set classifiers ieee-802.1 c802 forwarding-class VOIP loss-priority low code-
   points 111
   ```

2. Assign best-effort traffic to queue.

   ```
   [edit class-of-service forwarding-classes class]
   user@host# set BE queue-num 0
   user@host# set SIG queue-num 1
   user@host# set AF queue-num 2
   user@host# set Bronze-Class queue-num 3
   user@host# set Silver-Class queue-num 4
   user@host# set Gold-Class queue-num 5
   ```

```
user@host# set Control queue-num 6
user@host# set VOIP queue-num 7
```

3. Define mapping of forwarding classes to packet schedulers.

```
[edit class-of-service]
user@host# set scheduler-maps test forwarding-class BE scheduler s-be
user@host# set scheduler-maps test forwarding-class SIG scheduler s-sig
user@host# set scheduler-maps test forwarding-class AF scheduler s-af
user@host# set scheduler-maps test forwarding-class Bronze-Class scheduler s-bronze
user@host# set scheduler-maps test forwarding-class Silver-Class scheduler s-silver
user@host# set scheduler-maps test forwarding-class Gold-Class scheduler s-gold
user@host# set scheduler-maps test forwarding-class Control scheduler s-nc
user@host# set scheduler-maps test forwarding-class VOIP scheduler s-voip
```

4. Configure the CoS rewrite rules to map the forwarding class to the desired value for the 802.1p field.

```
[edit class-of-service]
user@host# set rewrite-rules ieee-802.1 rw802 forwarding-class BE loss-priority low code-
point 000
user@host# set rewrite-rules ieee-802.1 rw802 forwarding-class SIG loss-priority low code-
point 001
user@host# set rewrite-rules ieee-802.1 rw802 forwarding-class AF loss-priority low code-
point 010
user@host# set rewrite-rules ieee-802.1 rw802 forwarding-class Bronze-Class loss-priority low
code-point 011
user@host# set rewrite-rules ieee-802.1 rw802 forwarding-class Silver-Class loss-priority low
code-point 100
user@host# set rewrite-rules ieee-802.1 rw802 forwarding-class Gold-Class loss-priority low
code-point 101
user@host# set rewrite-rules ieee-802.1 rw802 forwarding-class Control loss-priority low code-
point 110
user@host# set rewrite-rules ieee-802.1 rw802 forwarding-class VOIP loss-priority low code-
point 111
```

5. Configure eight packet schedulers with scheduling priority and transmission rates.

```
[edit class-of-service]
user@host# set schedulers s-be transmit-rate percent 15
user@host# set schedulers s-be priority low
```

```
user@host# set schedulers s-sig transmit-rate percent 15
user@host# set schedulers s-sig priority low
user@host# set schedulers s-af transmit-rate percent 20
user@host# set schedulers s-af priority medium-low
user@host# set schedulers s-bronze transmit-rate percent 20
user@host# set schedulers s-bronze priority medium-low
user@host# set schedulers s-silver transmit-rate percent 10
user@host# set schedulers s-silver priority medium-high
user@host# set schedulers s-gold transmit-rate percent 10
user@host# set schedulers s-gold priority medium-high
user@host# set schedulers s-nc transmit-rate percent 5
user@host# set schedulers s-nc priority high
user@host# set schedulers s-voip transmit-rate percent 5
user@host# set schedulers s-voip priority high
```

6. Apply the classifier and rewrite rules to interfaces.

```
[edit class-of-service]
user@host# set interfaces reth0 unit 0 classifiers ieee-802.1 c802
user@host# set interfaces reth1 unit 0 rewrite-rules ieee-802.1 rw802
```

7. Apply the scheduler-map “test” to an interface.

```
[edit class-of-service]
user@host# set interfaces reth0 scheduler-map test
```

8. Apply the shaping rates to control the maximum rate of traffic transmitted on an interface.

```
[edit class-of-service]
user@host# set interfaces reth0 shaping-rate 1g
```

Results

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

```
classifiers {
    ieee-802.1 c802 {
        forwarding-class BE {
            loss-priority low code-points 000;
        }
        forwarding-class SIG {
            loss-priority low code-points 001;
        }
        forwarding-class AF {
            loss-priority low code-points 010;
        }
        forwarding-class Bronze-Class {
            loss-priority low code-points 011;
        }
        forwarding-class Silver-Class {
            loss-priority low code-points 100;
        }
        forwarding-class Gold-Class {
            loss-priority low code-points 101;
        }
        forwarding-class Control {
            loss-priority low code-points 110;
        }
        forwarding-class VOIP {
            loss-priority low code-points 111;
        }
    }
}
forwarding-classes {
    class BE queue-num 0;
    class SIG queue-num 1;
    class VOIP queue-num 7;
    class AF queue-num 2;
    class Bronze-Class queue-num 3;
    class Silver-Class queue-num 4;
    class Gold-Class queue-num 5;
```

```
        class Control queue-num 6;
    }
    interfaces {
        reth0 {
            shaping-rate 1g;
            unit 0 {
                scheduler-map test;
            }
        }
        reth0 {
            shaping-rate 1g;
            unit 0 {
                classifiers {
                    ieee-802.1 c802;
                }
                rewrite-rules {
                    ieee-802.1 rw802;
                }
            }
        }
    }
    rewrite-rules {
        ieee-802.1 rw802 {
            forwarding-class BE {
                loss-priority low code-point 000;
            }
            forwarding-class SIG {
                loss-priority low code-point 001;
            }
            forwarding-class AF {
                loss-priority low code-point 010;
            }
            forwarding-class Bronze-Class {
                loss-priority low code-point 011;
            }
            forwarding-class Silver-Class {
                loss-priority low code-point 100;
            }
            forwarding-class Gold-Class {
                loss-priority low code-point 101;
            }
            forwarding-class Control {
                loss-priority low code-point 110;
        }
        forwarding-class VOIP {
            loss-priority low code-point 111;
        }
    }
}
scheduler-maps {
    test {
        forwarding-class BE scheduler s-be;
        forwarding-class VOIP scheduler s-voip;
        forwarding-class Gold-Class scheduler s-gold;
        forwarding-class SIG scheduler s-sig;
        forwarding-class AF scheduler s-af;
        forwarding-class Bronze-Class scheduler s-bronze;
        forwarding-class Silver-Class scheduler s-silver;
        forwarding-class Control scheduler s-nc;
    }
}
schedulers {
    s-be {
        transmit-rate percent 15;
        priority low;
    }
    s-nc {
        transmit-rate percent 5;
        priority high;
    }
    s-gold {
        transmit-rate percent 10;
        priority medium-high;
    }
    s-sig {
        transmit-rate percent 15;
        priority low;
    }
    s-af {
        transmit-rate percent 20;
        priority medium-low;
    }
    s-bronze {
        transmit-rate percent 20;
        priority medium-low;
    }
```

```
        s-silver {
            transmit-rate percent 10;
            priority medium-high;
        }
        s-voip {
            transmit-rate percent 5;
            priority high;
        }
    }
```

If you are done configuring the device, enter `commit` from configuration mode.

Verification

IN THIS SECTION

- Verifying Class-of-Service Configuration | 338
- Verifying the Number of Dedicated Queues Configured on MPC Interfaces | 339

Confirm that the configuration is working properly.

Verifying Class-of-Service Configuration

Purpose

Verify that CoS is configured.

Action

From operational mode, enter the `show class-of-service classifier` command.

```
user@host> show class-of-service classifier type ieee-802.1

Forwarding class           ID    Queue Restricted queue  Fabric priority  Policing priority
SPU priority
  BE                       0       0          0            low                normal
low
  SIG                      1       1          1            low                normal
```

```
low
  AF                        2       2          2             low                 normal
low
  Bronze-Class              3       3          3             low                 normal
low
  Silver-Class              4       4          0             low                 normal
low
  Gold-Class                5       5          1             low                 normal
low
  Control                   6       6          2             low                 normal
low
  VOIP                      7       7          3             low                 normal
low
```

Verifying the Number of Dedicated Queues Configured on MPC Interfaces

Purpose

Display the number of dedicated queue resources that are configured for the interfaces on a port.

Action

From operational mode, enter the `show class-of-service interface` command.

```
user@host> show class-of-service interface reth0

Physical interface: reth0, Index: 129
Queues supported: 8, Queues in use: 4
  Scheduler map: <default>, Index: 2
  Congestion-notification: Disabled

  Logical interface: reth0.0, Index: 71
    Object                  Name                   Type                     Index
    Classifier              dscp-ipv6-compatibility dscp-ipv6                   9
    Classifier              ipprec-compatibility   ip                          13

  Logical interface: reth1.32767, Index: 70
```

RELATED DOCUMENTATION

Understanding IOC Hardware Properties | **315**

CoS Support on the SRX5000 Module Port Concentrator Overview | **326**

6

PART

Configuration Statements and Operational Commands

- Junos CLI Reference Overview | **342**

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