JUNIPER NETWORKS

Junos® OS

Class of Service Configuration Guide

Release

10.4

Published: 2010-10-08

Juniper Networks, Inc.
1194 North Mathilda Avenue
Sunnyvale, California 94089
USA
408-745-2000
www.juniper.net

Junos® OS Class of Service Configuration Guide
Release 10.4

Writing: Walter Goralski
Editing: Joanne McClintock
Illustration: Nathaniel Woodward
Cover Design: Edmonds Design

Revision History
October 2010—R1 Junos OS 10.4

The information in this document is current as of the date listed in the revision history.

YEAR 2000 NOTICE

Juniper Networks hardware and software products are Year 2000 compliant. The 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

READ THIS END USER LICENSE AGREEMENT ("AGREEMENT") BEFORE DOWNLOADING, INSTALLING, OR USING THE SOFTWARE. BY DOWNLOADING, INSTALLING, OR USING THE SOFTWARE OR OTHERWISE EXPRESSING YOUR AGREEMENT TO THE TERMS CONTAINED HEREIN, YOU (AS CUSTOMER OR IF YOU ARE NOT THE CUSTOMER, AS A REPRESENTATIVE/AGENT AUTHORIZED TO BIND THE CUSTOMER) CONSENT TO BE BOUND BY THIS AGREEMENT. IF YOU DO NOT OR CANNOT AGREE TO THE TERMS CONTAINED HEREIN, THEN (A) DO NOT DOWNLOAD, INSTALL, OR USE THE SOFTWARE, AND (B) YOU MAY CONTACT JUNIPER NETWORKS REGARDING LICENSE TERMS.

1. **The Parties.** The parties to this Agreement are (i) Juniper Networks, Inc. (if the Customer's principal office is located in the Americas) or Juniper Networks (Cayman) Limited (if the Customer's principal office is located outside the Americas) (such applicable entity being referred to herein as "Juniper"), and (ii) the person or organization that originally purchased from Juniper or an authorized Juniper reseller the applicable license(s) for use of the Software ("Customer") (collectively, the "Parties").

2. **The Software.** In this Agreement, "Software" means the program modules and features of the Juniper or Juniper-supplied software, for which Customer has paid the applicable license or support fees to Juniper or an authorized Juniper reseller, or which was embedded by Juniper in equipment which Customer purchased from Juniper or an authorized Juniper reseller. "Software" also includes updates, upgrades and new releases of such software. "Embedded Software" means Software which Juniper has embedded in or loaded onto the Juniper equipment and any updates, upgrades, additions or replacements which are subsequently embedded in or loaded onto the equipment.

3. **License Grant.** Subject to payment of the applicable fees and the limitations and restrictions set forth herein, Juniper grants to Customer a non-exclusive and non-transferable license, without right to sublicense, to use the Software, in executable form only, subject to the following use restrictions:

a. Customer shall use Embedded Software solely as embedded in, and for execution on, Juniper equipment originally purchased by Customer from Juniper or an authorized Juniper reseller.

b. Customer shall use the Software on a single hardware chassis having a single processing unit, or as many chassis or processing units for which Customer has paid the applicable license fees; provided, however, with respect to the Steel-Belted Radius or Odyssey Access Client software only, Customer shall use such Software on a single computer containing a single physical random access memory space and containing any number of processors. Use of the Steel-Belted Radius or IMS AAA software on multiple computers or virtual machines (e.g., Solaris zones) requires multiple licenses, regardless of whether such computers or virtualizations are physically contained on a single chassis.

c. Product purchase documents, paper or electronic user documentation, and/or the particular licenses purchased by Customer may specify limits to Customer's use of the Software. Such limits may restrict use to a maximum number of seats, registered endpoints, concurrent users, sessions, calls, connections, subscribers, clusters, nodes, realms, devices, links, ports or transactions, or require the purchase of separate licenses to use particular features, functionalities, services, applications, operations, or capabilities, or provide throughput, performance, configuration, bandwidth, interface, processing, temporal, or geographical limits. In addition, such limits may restrict the use of the Software to managing certain kinds of networks or require the Software to be used only in conjunction with other specific Software. Customer's use of the Software shall be subject to all such limitations and purchase of all applicable licenses.

d. For any trial copy of the Software, Customer's right to use the Software expires 30 days after download, installation or use of the Software. Customer may operate the Software after the 30-day trial period only if Customer pays for a license to do so. Customer may not extend or create an additional trial period by re-installing the Software after the 30-day trial period.

e. The Global Enterprise Edition of the Steel-Belted Radius software may be used by Customer only to manage access to Customer's enterprise network. Specifically, service provider customers are expressly prohibited from using the Global Enterprise Edition of the Steel-Belted Radius software to support any commercial network access services.

The foregoing license is not transferable or assignable by Customer. No license is granted herein to any user who did not originally purchase the applicable license(s) for the Software from Juniper or an authorized Juniper reseller.

4. **Use Prohibitions.** Notwithstanding the foregoing, the license provided herein does not permit the Customer to, and Customer agrees not to and shall not: (a) modify, unbundle, reverse engineer, or create derivative works based on the Software; (b) make unauthorized copies of the Software (except as necessary for backup purposes); (c) rent, sell, transfer, or grant any rights in and to any copy of the Software, in any form, to any third party; (d) remove any proprietary notices, labels, or marks on or in any copy of the Software or any product in which the Software is embedded; (e) distribute any copy of the Software to any third party, including as may be embedded in Juniper equipment sold in the secondhand market; (f) use any 'locked' or key-restricted feature, function, service, application, operation, or capability without first purchasing the applicable license(s) and obtaining a valid key from Juniper, even if such feature, function, service, application, operation, or capability is enabled without a key; (g) distribute any key for the Software provided by Juniper to any third party; (h) use the

Software in any manner that extends or is broader than the uses purchased by Customer from Juniper or an authorized Juniper reseller; (i) use Embedded Software on non-Juniper equipment; (j) use Embedded Software (or make it available for use) on Juniper equipment that the Customer did not originally purchase from Juniper or an authorized Juniper reseller; (k) disclose the results of testing or benchmarking of the Software to any third party without the prior written consent of Juniper; or (l) use the Software in any manner other than as expressly provided herein.

5. **Audit.** Customer shall maintain accurate records as necessary to verify compliance with this Agreement. Upon request by Juniper, Customer shall furnish such records to Juniper and certify its compliance with this Agreement.

6. **Confidentiality.** The Parties agree that aspects of the Software and associated documentation are the confidential property of Juniper. As such, Customer shall exercise all reasonable commercial efforts to maintain the Software and associated documentation in confidence, which at a minimum includes restricting access to the Software to Customer employees and contractors having a need to use the Software for Customer's internal business purposes.

7. **Ownership.** Juniper and Juniper's licensors, respectively, retain ownership of all right, title, and interest (including copyright) in and to the Software, associated documentation, and all copies of the Software. Nothing in this Agreement constitutes a transfer or conveyance of any right, title, or interest in the Software or associated documentation, or a sale of the Software, associated documentation, or copies of the Software.

8. **Warranty, Limitation of Liability, Disclaimer of Warranty.** The warranty applicable to the Software shall be as set forth in the warranty statement that accompanies the Software (the "Warranty Statement"). Nothing in this Agreement shall give rise to any obligation to support the Software. Support services may be purchased separately. Any such support shall be governed by a separate, written support services agreement. TO THE MAXIMUM EXTENT PERMITTED BY LAW, JUNIPER SHALL NOT BE LIABLE FOR ANY LOST PROFITS, LOSS OF DATA, OR COSTS OR PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES, OR FOR ANY SPECIAL, INDIRECT, OR CONSEQUENTIAL DAMAGES ARISING OUT OF THIS AGREEMENT, THE SOFTWARE, OR ANY JUNIPER OR JUNIPER-SUPPLIED SOFTWARE. IN NO EVENT SHALL JUNIPER BE LIABLE FOR DAMAGES ARISING FROM UNAUTHORIZED OR IMPROPER USE OF ANY JUNIPER OR JUNIPER-SUPPLIED SOFTWARE. EXCEPT AS EXPRESSLY PROVIDED IN THE WARRANTY STATEMENT TO THE EXTENT PERMITTED BY LAW, JUNIPER DISCLAIMS ANY AND ALL WARRANTIES IN AND TO THE SOFTWARE (WHETHER EXPRESS, IMPLIED, STATUTORY, OR OTHERWISE), INCLUDING ANY IMPLIED WARRANTY OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE, OR NONINFRINGEMENT. IN NO EVENT DOES JUNIPER WARRANT THAT THE SOFTWARE, OR ANY EQUIPMENT OR NETWORK RUNNING THE SOFTWARE, WILL OPERATE WITHOUT ERROR OR INTERRUPTION, OR WILL BE FREE OF VULNERABILITY TO INTRUSION OR ATTACK. In no event shall Juniper's or its suppliers' or licensors' liability to Customer, whether in contract, tort (including negligence), breach of warranty, or otherwise, exceed the price paid by Customer for the Software that gave rise to the claim, or if the Software is embedded in another Juniper product, the price paid by Customer for such other product. Customer acknowledges and agrees that Juniper has set its prices and entered into this Agreement in reliance upon the disclaimers of warranty and the limitations of liability set forth herein, that the same reflect an allocation of risk between the Parties (including the risk that a contract remedy may fail of its essential purpose and cause consequential loss), and that the same form an essential basis of the bargain between the Parties.

9. **Termination.** Any breach of this Agreement or failure by Customer to pay any applicable fees due shall result in automatic termination of the license granted herein. Upon such termination, Customer shall destroy or return to Juniper all copies of the Software and related documentation in Customer's possession or control.

10. **Taxes.** All license fees payable under this agreement are exclusive of tax. Customer shall be responsible for paying Taxes arising from the purchase of the license, or importation or use of the Software. If applicable, valid exemption documentation for each taxing jurisdiction shall be provided to Juniper prior to invoicing, and Customer shall promptly notify Juniper if their exemption is revoked or modified. All payments made by Customer shall be net of any applicable withholding tax. Customer will provide reasonable assistance to Juniper in connection with such withholding taxes by promptly: providing Juniper with valid tax receipts and other required documentation showing Customer's payment of any withholding taxes; completing appropriate applications that would reduce the amount of withholding tax to be paid; and notifying and assisting Juniper in any audit or tax proceeding related to transactions hereunder. Customer shall comply with all applicable tax laws and regulations, and Customer will promptly pay or reimburse Juniper for all costs and damages related to any liability incurred by Juniper as a result of Customer's non-compliance or delay with its responsibilities herein. Customer's obligations under this Section shall survive termination or expiration of this Agreement.

11. **Export.** Customer agrees to comply with all applicable export laws and restrictions and regulations of any United States and any applicable foreign agency or authority, and not to export or re-export the Software or any direct product thereof in violation of any such restrictions, laws or regulations, or without all necessary approvals. Customer shall be liable for any such violations. The version of the Software supplied to Customer may contain encryption or other capabilities restricting Customer's ability to export the Software without an export license.

12. **Commercial Computer Software.** The Software is "commercial computer software" and is provided with restricted rights. Use, duplication, or disclosure by the United States government is subject to restrictions set forth in this Agreement and as provided in DFARS 227.7201 through 227.7202-4, FAR 12.212, FAR 27.405(b)(2), FAR 52.227-19, or FAR 52.227-14(ALT III) as applicable.

13. **Interface Information.** To the extent required by applicable law, and at Customer's written request, Juniper shall provide Customer with the interface information needed to achieve interoperability between the Software and another independently created program, on payment of applicable fee, if any. Customer shall observe strict obligations of confidentiality with respect to such information and shall use such information in compliance with any applicable terms and conditions upon which Juniper makes such information available.

14. **Third Party Software.** Any licensor of Juniper whose software is embedded in the Software and any supplier of Juniper whose products or technology are embedded in (or services are accessed by) the Software shall be a third party beneficiary with respect to this Agreement, and such licensor or vendor shall have the right to enforce this Agreement in its own name as if it were Juniper. In addition, certain third party software may be provided with the Software and is subject to the accompanying license(s), if any, of its respective owner(s). To the extent portions of the Software are distributed under and subject to open source licenses obligating Juniper to make the source code for such portions publicly available (such as the GNU General Public License ("GPL") or the GNU Library General Public License ("LGPL")), Juniper will make such source code portions (including Juniper modifications, as appropriate) available upon request for a period of up to three years from the date of distribution. Such request can be made in writing to Juniper Networks, Inc., 1194 N. Mathilda Ave., Sunnyvale, CA 94089, ATTN: General Counsel. You may obtain a copy of the GPL at http://www.gnu.org/licenses/gpl.html, and a copy of the LGPL at http://www.gnu.org/licenses/lgpl.html .

15. **Miscellaneous.** This Agreement shall be governed by the laws of the State of California without reference to its conflicts of laws principles. The provisions of the U.N. Convention for the International Sale of Goods shall not apply to this Agreement. For any disputes arising under this Agreement, the Parties hereby consent to the personal and exclusive jurisdiction of, and venue in, the state and federal courts within Santa Clara County, California. This Agreement constitutes the entire and sole agreement between Juniper and the Customer with respect to the Software, and supersedes all prior and contemporaneous agreements relating to the Software, whether oral or written (including any inconsistent terms contained in a purchase order), except that the terms of a separate written agreement executed by an authorized Juniper representative and Customer shall govern to the extent such terms are inconsistent or conflict with terms contained herein. No modification to this Agreement nor any waiver of any rights hereunder shall be effective unless expressly assented to in writing by the party to be charged. If any portion of this Agreement is held invalid, the Parties agree that such invalidity shall not affect the validity of the remainder of this Agreement. This Agreement and associated documentation has been written in the English language, and the Parties agree that the English version will govern. (For Canada: Les parties aux présentés confirment leur volonté que cette convention de même que tous les documents y compris tout avis qui s'y rattaché, soient redigés en langue anglaise. (Translation: The parties confirm that this Agreement and all related documentation is and will be in the English language)).

Abbreviated Table of Contents

Table of Contents

Part 6 Index

List of Figures

List of Tables

About This Guide

This preface provides the following guidelines for using the *Junos® OS Class of Service Configuration Guide*:

Junos Documentation and Release Notes

For a list of related Junos documentation, see http://www.juniper.net/techpubs/software/junos/ .

If the information in the latest release notes differs from the information in the documentation, follow the *Junos Release Notes.*

To obtain the most current version of all Juniper Networks® technical documentation, see the product documentation page on the Juniper Networks website at http://www.juniper.net/techpubs/.

Juniper Networks supports a technical book program to publish books by Juniper Networks engineers and subject matter experts with book publishers around the world. These books go beyond the technical documentation to explore the nuances of network architecture, deployment, and administration using the Junos operating system (Junos OS) and Juniper Networks devices. In addition, the Juniper Networks Technical Library, published in conjunction with O'Reilly Media, explores improving network security, reliability, and availability using Junos OS configuration techniques. All the books are for sale at technical bookstores and book outlets around the world. The current list can be viewed at http://www.juniper.net/books .

Objectives

This guide provides an overview of the class-of-service features of the Junos OS and describes how to configure these properties on the routing platform.

NOTE: For additional information about Junos OS—either corrections to or information that might have been omitted from this guide—see the software release notes at http://www.juniper.net/.

Audience

This guide is designed for network administrators who are configuring and monitoring a Juniper Networks M Series, MX Series, T Series, EX Series, or J Series router or switch.

To use this guide, you need a broad understanding of networks in general, the Internet in particular, networking principles, and network configuration. You must also be familiar with one or more of the following Internet routing protocols:

- Border Gateway Protocol (BGP)
- Distance Vector Multicast Routing Protocol (DVMRP)
- Intermediate System-to-Intermediate System (IS-IS)
- Internet Control Message Protocol (ICMP) router discovery
- Internet Group Management Protocol (IGMP)
- Multiprotocol Label Switching (MPLS)
- Open Shortest Path First (OSPF)
- Protocol-Independent Multicast (PIM)
- Resource Reservation Protocol (RSVP)
- Routing Information Protocol (RIP)
- Simple Network Management Protocol (SNMP)

Personnel operating the equipment must be trained and competent; must not conduct themselves in a careless, willfully negligent, or hostile manner; and must abide by the instructions provided by the documentation.

Supported Platforms

For the features described in this manual, Junos OS currently supports the following platforms:

- J Series
- M Series

- MX Series
- T Series
- EX Series

Using the Indexes

This reference contains two indexes: a complete index that includes topic entries, and an index of statements and commands only.

In the index of statements and commands, an entry refers to a statement summary section only. In the complete index, the entry for a configuration statement or command contains at least two parts:

- The primary entry refers to the statement summary section.
- The secondary entry, *usage guidelines,* refers to the section in a configuration guidelines chapter that describes how to use the statement or command.

Using the Examples in This Manual

If you want to use the examples in this manual, you can use the **load merge** or the **load merge relative** command. These commands cause the software to merge the incoming configuration into the current candidate configuration. If the example configuration contains the top level of the hierarchy (or multiple hierarchies), the example is a *full example*. In this case, use the **load merge** command.

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

Merging a Full Example

To merge a full example, follow these steps:

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

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

   ```
   system {
     scripts {
       commit {
         file ex-script.xsl;
       }
     }
   }
   interfaces {
     fxp0 {
       disable;
       unit 0 {
   ```



```
            family inet {
               address 10.0.0.1/24;
            }
         }
      }
   }
```

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

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

Merging a Snippet

To merge a snippet, follow these steps:

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

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

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

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

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

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

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

For more information about the **load** command, see the *Junos OS CLI User Guide*.

Documentation Conventions

Table 1 on page xxxiii defines notice icons used in this guide.

Table 1: Notice Icons

Icon	Meaning	Description
	Informational note	Indicates important features or instructions.
	Caution	Indicates a situation that might result in loss of data or hardware damage.
	Warning	Alerts you to the risk of personal injury or death.
	Laser warning	Alerts you to the risk of personal injury from a laser.

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

Table 2: Text and Syntax Conventions

Convention	Description	Examples
Bold text like this	Represents text that you type.	To enter configuration mode, type the **configure** command: user@host> **configure**
`Fixed-width text like this`	Represents output that appears on the terminal screen.	`user@host>` **show chassis alarms** `No alarms currently active`
Italic text like this	• Introduces important new terms. • Identifies book names. • Identifies RFC and Internet draft titles.	• A policy *term* is a named structure that defines match conditions and actions. • *Junos System Basics Configuration Guide* • RFC 1997, *BGP Communities Attribute*
Italic text like this	Represents variables (options for which you substitute a value) in commands or configuration statements.	Configure the machine's domain name: [edit] root@# **set system domain-name *domain-name***
Text like this	Represents names of configuration statements, commands, files, and directories; IP addresses; configuration hierarchy levels; or labels on routing platform components.	• To configure a stub area, include the **stub** statement at the **[edit protocols ospf area area-id]** hierarchy level. • The console port is labeled **CONSOLE**.
< > (angle brackets)	Enclose optional keywords or variables.	**stub <default-metric *metric*>;**

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

Convention	Description	Examples
| (pipe symbol)	Indicates a choice between the mutually exclusive keywords or variables on either side of the symbol. The set of choices is often enclosed in parentheses for clarity.	**broadcast | multicast** **(*string1* | *string2* | *string3*)**
# (pound sign)	Indicates a comment specified on the same line as the configuration statement to which it applies.	**rsvp { # Required for dynamic MPLS only**
[ ] (square brackets)	Enclose a variable for which you can substitute one or more values.	**community name members [*community-ids*]**

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

; (semicolon)	Identifies a leaf statement at a configuration hierarchy level.	
J-Web GUI Conventions		
Bold text like this	Represents J-Web graphical user interface (GUI) items you click or select.	• In the Logical Interfaces box, select **All Interfaces**. • To cancel the configuration, click **Cancel**.
> (bold right angle bracket)	Separates levels in a hierarchy of J-Web selections.	In the configuration editor hierarchy, select **Protocols>Ospf**.

Documentation Feedback

We encourage you to provide feedback, comments, and suggestions so that we can improve the documentation. You can send your comments to **techpubs-comments@juniper.net**, or fill out the documentation feedback form at https://www.juniper.net/cgi-bin/docbugreport/. If you are using e-mail, be sure to include the following information with your comments:

- Document or topic name
- URL or page number
- Software release version (if applicable)

Requesting Technical Support

Technical product support is available through the Juniper Networks Technical Assistance Center (JTAC). If you are a customer with an active J-Care or JNASC support contract, or are covered under warranty, and need postsales technical support, you can access our tools and resources online or open a case with JTAC.

- JTAC policies—For a complete understanding of our JTAC procedures and policies, review the JTAC User Guide located at http://www.juniper.net/us/en/local/pdf/resource-guides/7100059-en.pdf .
- Product warranties—For product warranty information, visit http://www.juniper.net/support/warranty/ .
- JTAC Hours of Operation —The JTAC centers have resources available 24 hours a day, 7 days a week, 365 days a year.

Self-Help Online Tools and Resources

For quick and easy problem resolution, Juniper Networks has designed an online self-service portal called the Customer Support Center (CSC) that provides you with the following features:

- Find CSC offerings: http://www.juniper.net/customers/support/
- Find product documentation: http://www.juniper.net/techpubs/
- Find solutions and answer questions using our Knowledge Base: http://kb.juniper.net/
- Download the latest versions of software and review release notes: http://www.juniper.net/customers/csc/software/
- Search technical bulletins for relevant hardware and software notifications: https://www.juniper.net/alerts/
- Join and participate in the Juniper Networks Community Forum: http://www.juniper.net/company/communities/
- Open a case online in the CSC Case Management tool: http://www.juniper.net/cm/

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

Opening a Case with JTAC

You can open a case with JTAC on the Web or by telephone.

- Use the Case Management tool in the CSC at http://www.juniper.net/cm/ .
- Call 1-888-314-JTAC (1-888-314-5822 toll-free in the USA, Canada, and Mexico).

For international or direct-dial options in countries without toll-free numbers, visit us at http://www.juniper.net/support/requesting-support.html

PART 1

CoS Overview

CHAPTER 1

CoS Overview

This chapter discusses the following topics:

CoS Overview

When a network experiences congestion and delay, some packets must be dropped. Junos 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 rules that you configure.

For interfaces that carry IPv4, IPv6, and MPLS traffic, you can configure Junos CoS features to provide multiple classes of service for different applications. On the router, 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.

The Junos CoS features provide a set of mechanisms that you can use to provide differentiated services when best-effort traffic delivery is insufficient. In designing CoS applications, you must give careful consideration to your service needs, and you must thoroughly plan and design your CoS configuration to ensure consistency across all routers in a CoS domain. You must also consider all the routers and other networking equipment in the CoS domain to ensure interoperability among all equipment.

Because Juniper Networks routers implement CoS in hardware rather than in software, you can experiment with and deploy CoS features without adversely affecting packet forwarding and routing performance.

Related Documentation

- Hardware Capabilities and Limitations on page 259

CoS Standards

The standards for Junos class of service (CoS) capabilities are defined in the following RFCs:

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

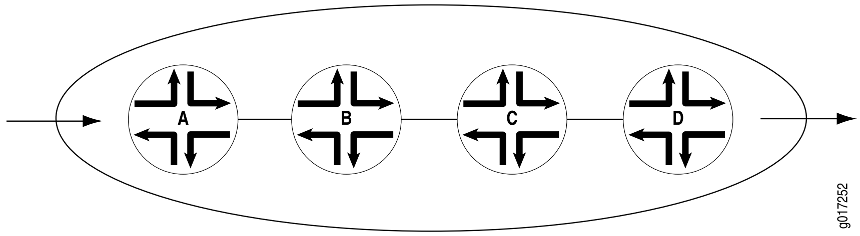
Understanding Packet Flow Across a Network

CoS works by examining traffic entering at the edge of your network. The edge routers classify traffic into defined service groups, to provide the special treatment of traffic across the network. For example, voice traffic can be sent across certain links, and data traffic can use other links. In addition, the data traffic streams can be serviced differently along the network path 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 router in the network. Generally, each router examines the packets that enter it to determine their CoS settings. These settings then dictate which packets are first transmitted to the next downstream router. In addition, the routers at the edges of the network might be required to alter the CoS settings of the packets that enter the network from the customer or peer networks.

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

Figure 1: Packet Flow Across the Network



Junos CoS Components

Junos CoS consists of many components that you can combine and tune to provide the level of services required by customers.

The Junos CoS components include:

- 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.
- Classifiers—*Packet classification* refers to the examination of an incoming packet. This function associates the packet with a particular CoS servicing level. In the Junos OS, classifiers associate incoming packets with a forwarding class and loss priority and, based on the associated forwarding class, assign packets to output queues. Two general types of classifiers are supported:
  - Behavior aggregate or CoS value traffic classifiers—A *behavior aggregate* (BA) is a method of classification that operates on a packet as it enters the router. The CoS value in the packet header is examined, and this single field determines the CoS settings applied to the packet. BA classifiers allow you to set the forwarding class and loss priority of a packet based on the Differentiated Services code point (DSCP) value, DSCP IPv6 value, IP precedence value, MPLS EXP bits, and IEEE 802.1p value. The default classifier is based on the IP precedence value.
  - Multifield traffic classifiers—A *multifield* classifier is a second method for classifying traffic flows. Unlike a behavior aggregate, a multifield classifier can examine multiple fields in the packet. Examples of some fields that a multifield classifier can examine include the source and destination address of the packet as well as the source and destination port numbers of the packet. With multifield classifiers, you set the forwarding class and loss priority of a packet based on firewall filter rules.
- Forwarding classes—The *forwarding classes* affect the forwarding, scheduling, and marking policies applied to packets as they transit a router. The forwarding class plus the loss priority define the per-hop behavior. Four categories of forwarding classes are supported: best effort, assured forwarding, expedited forwarding, and network control. For Juniper Networks M Series Multiservice Edge Routers, four forwarding classes are supported; you can configure up to one each of the four types of forwarding classes. For M120 and M320 Multiservice Edge Routers, MX Series Ethernet Services Routers, and T Series Core Routers, 16 forwarding classes are supported, so you can classify packets more granularly. For example, you can configure multiple classes of expedited forwarding (EF) traffic: EF, EF1, and EF2.

- Loss priorities—*Loss priorities* allow you to set the priority of dropping a packet. Loss priority affects the scheduling of a packet without affecting the packet's relative ordering. You can use the packet loss priority (PLP) bit as part of a congestion control strategy. You can use the loss priority setting to identify packets that have experienced congestion. Typically you mark packets exceeding some service level with a high loss priority. You set loss priority by configuring a classifier or a policer. The loss priority is used later in the work flow to select one of the drop profiles used by RED.
- Forwarding policy options—These options allow you to associate forwarding classes with next hops. Forwarding policy also allows you to create classification overrides, which assign forwarding classes to sets of prefixes.
- Transmission scheduling and rate control—These parameters provide you with a variety of tools to manage traffic flows:
  - Queuing—After a packet is sent to the outgoing interface on a router, it is queued for transmission on the physical media. The amount of time a packet is queued on the router is determined by the availability of the outgoing physical media as well as the amount of traffic using the interface.
  - Schedulers—An individual router interface has multiple queues assigned to store packets. The router determines which queue to service based on a particular method of scheduling. This process often involves a determination of which type of packet should be transmitted before another. Junos 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.
  - Fabric schedulers—For M320 and T Series routers only, fabric schedulers allow you to identify a packet as high or low priority based on its forwarding class, and to associate schedulers with the fabric priorities.
  - 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 filters that can be associated with input or output interfaces.
- Rewrite rules—A *rewrite rule* sets the appropriate CoS bits in the outgoing packet. This allows the next downstream router to classify the packet into the appropriate service group. Rewriting, or marking, outbound packets is useful when the router is at the border of a network and must alter the CoS values to meet the policies of the targeted peer.

Default CoS Settings

If you do not configure any CoS settings on your router, 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 issuing the **show class-of-service** operational mode command. This section includes sample output displaying the default CoS settings. The sample output is truncated for brevity.

show class-of-service

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

Default Forwarding Classes

```
Forwarding class           Queue
  best-effort                          0
  expedited-forwarding                 1
  assured-forwarding                   2
  network-control                      3
```

Default Code-Point Aliases

```
Code point type: dscp
  Alias             Bit pattern
  af11              001010
  af12              001100
...
Code point type: dscp-ipv6
...
Code point type: exp
...
Code point type: ieee-802.1
...
Code point type: inet-precedence
...
```

Default Classifiers

```
Classifier: dscp-default, Code point type: dscp, Index: 7
...

Classifier: dscp-ipv6-default, Code point type: dscp-ipv6, Index: 8
...

Classifier: exp-default, Code point type: exp, Index: 9
...

Classifier: ieee8021p-default, Code point type: ieee-802.1, Index: 10
...

Classifier: ipprec-default, Code point type: inet-precedence, Index: 11
...

Classifier: ipprec-compatibility, Code point type: inet-precedence, Index: 12
...
```

Default Frame Relay Loss Priority Map

```
Loss-priority-map: frame-relay-de-default, Code point type: frame-relay-de, Index:
 13
  Code point         Loss priority
```



```
  0                  low
  1                  high
```

Default Rewrite Rules

```
Rewrite rule: dscp-default, Code point type: dscp, Index: 24
  Forwarding class                    Loss priority       Code point
  best-effort                         low                 000000
  best-effort                         high                000000
...

Rewrite rule: dscp-ipv6-default, Code point type: dscp-ipv6, Index: 25
...

Rewrite rule: exp-default, Code point type: exp, Index: 26
...

Rewrite rule: ieee8021p-default, Code point type: ieee-802.1, Index: 27
...

Rewrite rule: ipprec-default, Code point type: inet-precedence, Index: 28
...
```

Default Drop Profile

```
Drop profile: <default-drop-profile>, Type: discrete, Index: 1
  Fill level    Drop probability
         100                 100
```

Default Schedulers

```
Scheduler map: <default>, Index: 2

  Scheduler: <default-be>, Forwarding class: best-effort, Index: 17
    Transmit rate: 95 percent, Rate Limit: none, Buffer size: 95 percent, Priority:
 low
    Drop profiles:
      Loss priority    Protocol    Index    Name
      Low              Any             1    <default-drop-profile>
      High             Any             1    <default-drop-profile>
...
```

Related Documentation

- Default Forwarding Classes on page 112
- Default Behavior Aggregate Classification Overview on page 43
- Default Drop Profile on page 233
- Default Schedulers on page 147
- Default Fabric Priority Queuing on page 200

CoS Input and Output Configuration

This topic includes the following:

- CoS Inputs and Outputs Overview on page 9
- CoS Inputs and Outputs Examples on page 9

CoS Inputs and Outputs Overview

Some CoS components map one set of values to another set of values. Each mapping contains one or more inputs and one or more outputs.

Some CoS components map one set of values to another set of values. Each mapping contains one or more inputs and one or more outputs. When you configure a mapping, you set the outputs for a given set of inputs, as shown in Table 3 on page 9.

Table 3: CoS Mappings—Inputs and Outputs

CoS Mappings	Inputs	Outputs	Comments
classifiers	**code-points**	**forwarding-class** **loss-priority**	The map sets the forwarding class and PLP for a specific set of code points.
drop-profile-map	**loss-priority protocol**	**drop-profile**	The map sets the drop profile for a specific PLP and protocol type.
rewrite-rules	**loss-priority**	**code-points**	The map sets the code points for a specific forwarding class and PLP.

Related Documentation

- Default Behavior Aggregate Classification Overview on page 43
- Configuring Drop Profile Maps for Schedulers on page 159
- Applying Default Rewrite Rules on page 240
- CoS Inputs and Outputs Examples on page 9

CoS Inputs and Outputs Examples

Here are examples of configurations for classifiers, drop-profile maps, and rewrite rules.

In the following classifier example, packets with EXP bits **000** are assigned to the **data-queue** forwarding class with a **low** loss priority, and packets with EXP bits **001** are assigned to the **data-queue** forwarding class with a **high** loss priority.

```
[edit class-of-service]
classifiers {
  exp exp_classifier {
    forwarding-class data-queue {
      loss-priority low code-points 000;
      loss-priority high code-points 001;
    }
  }
}
```

In the following drop-profile map example, the scheduler includes two drop-profile maps, which specify that packets are evaluated by the **low-drop** drop profile if they have a **low** loss priority and are from any protocol. Packets are evaluated by the **high-drop** drop profile if they have a **high** loss priority and are from any protocol.

```
[edit class-of-service]
schedulers {
```



```
    best-effort {
        drop-profile-map loss-priority low protocol any drop-profile low-drop;
        drop-profile-map loss-priority high protocol any drop-profile high-drop;
    }
}
```

In the following rewrite rule example, packets in the **be** forwarding class with **low** loss priority are assigned the EXP bits **000**, and packets in the **be** forwarding class with **high** loss priority are assigned the EXP bits **001**.

```
[edit class-of-service]
rewrite-rules {
    exp exp-rw {
        forwarding-class be {
            loss-priority low code-point 000;
            loss-priority high code-point 001;
        }
    }
}
```

Related Documentation

- CoS Inputs and Outputs Overview on page 9

Packet Flow Within Routers

Packet flow differs by router type. This topic discusses the following:

Packet Flow Within Routers Overview

Although the architecture of Juniper Networks routers different in detail, the overall flow of a packet within the router remains consistent.

When a packet enters a Juniper Networks router, the PIC or other interface type receiving the packet retrieves it from the network and verifies that the link-layer information is valid. The packet is then passed to the concentrator device such as a Flexible PIC Concentrator (FPC), where the data link and network layer information is verified. In addition, the FPC is responsible for segmenting the packet into 64-byte units called J-cells. These cells are then written into packet storage memory while a notification cell is sent to the route lookup engine. The destination address listed in the notification cell is located in the forwarding table, and the next hop of the packet is written into the result cell. This result cell is queued on the appropriate outbound FPC until the outgoing interface is ready to transmit the packet. The FPC then reads the J-cells out of memory, re-forms the original packet, and sends the packet to the outgoing PIC, where it is transmitted back into the network.

Related Documentation

- Packet Flow on Juniper Networks J Series Services Routers on page 11
- Packet Flow on Juniper Networks M Series Multiservice Edge Routers on page 11
- Packet Flow on MX Series Ethernet Services Routers on page 13
- Packet Flow on Juniper Networks T Series Core Routers on page 17

Packet Flow on Juniper Networks J Series Services Routers

On J Series Services Routers, some of the hardware components associated with larger routers are virtualized.

These virtualized components include Packet Forwarding Engines, Routing Engines, and their associated ASICs. For this reason, packet flow on J Series routers cannot be described in terms of discrete hardware components.

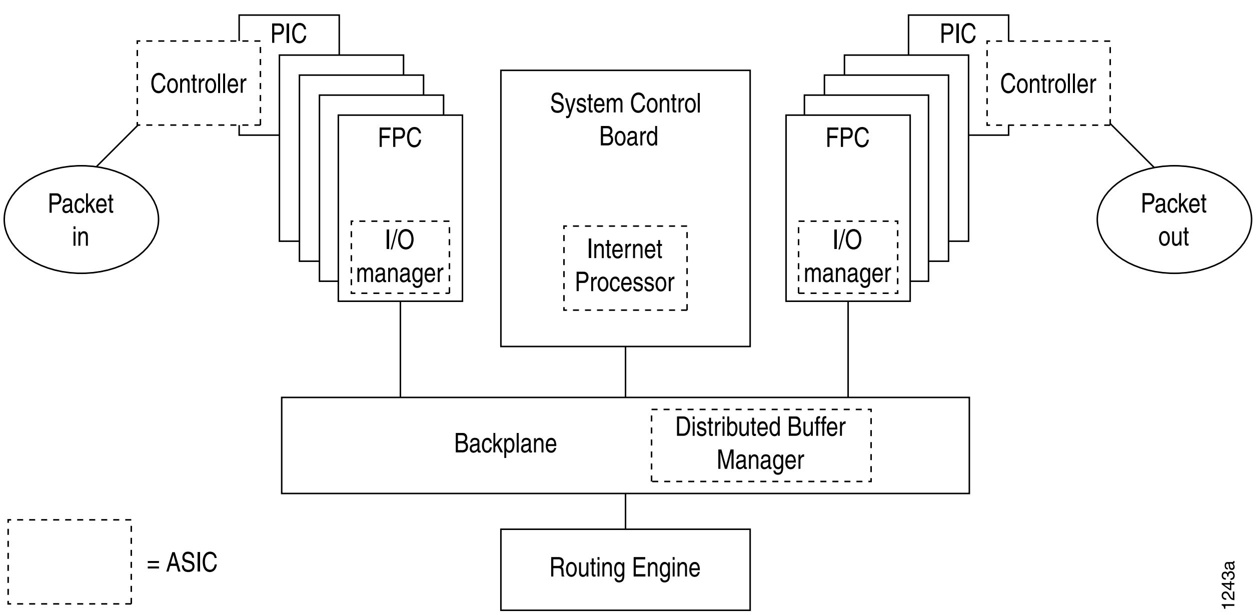
Related Documentation

- Packet Flow Within Routers Overview on page 10
- Packet Flow on Juniper Networks M Series Multiservice Edge Routers on page 11
- Packet Flow on MX Series Ethernet Services Routers on page 13
- Packet Flow on Juniper Networks T Series Core Routers on page 17

Packet Flow on Juniper Networks M Series Multiservice Edge Routers

On M Series Multiservice Edge Routers, CoS actions are performed in several locations in a Juniper Networks router: the incoming I/O Manager ASIC, the Internet Processor II ASIC, and the outgoing I/O Manager ASIC. These locations are shown in Figure 2 on page 11.

Figure 2: M Series Routers Packet Forwarding Engine Components and Data Flow



This topic describes the packet flow through the following components in more detail:

- Incoming I/O Manager ASIC on page 12
- Internet Processor ASIC on page 12
- Outgoing I/O Manager ASIC on page 12
- Enhanced CFEB and CoS on M7i and M10i Multiservice Edge Routers on page 12

Incoming I/O Manager ASIC

When a data packet is passed from the receiving interface to its connected FPC, it is received by the I/O Manager ASIC on that specific FPC. During the processing of the packet by this ASIC, the information in the packet's header is examined by a behavior aggregate (BA) classifier. This classification action associates the packet with a particular forwarding class. In addition, the value of the packet's loss priority bit is set by this classifier. Both the forwarding class and loss priority information are placed into the notification cell, which is then transmitted to the Internet Processor II ASIC.

Internet Processor ASIC

The Internet Processor II ASIC receives notification cells representing inbound data packets and performs route lookups in the forwarding table. This lookup determines the outgoing interface on the router and the next-hop IP address for the data packet. While the packet is being processed by the Internet Processor II ASIC, it might also be evaluated by a firewall filter, which is configured on either the incoming or outgoing interface. This filter can perform the functions of a multifield classifier by matching on multiple elements within the packet and overwriting the forwarding class, loss priority settings, or both within the notification cell. Once the route lookup and filter evaluations are complete, the notification cell, now called the result cell, is passed to the I/O Manager ASIC on the FPC associated with the outgoing interface.

Outgoing I/O Manager ASIC

When the result cell is received by the I/O Manager ASIC, it is placed into a queue to await transmission on the physical media. The specific queue used by the ASIC is determined by the forwarding class associated with the data packet. The configuration of the queue itself helps determine the service the packet receives while in this queued state. This functionality guarantees that certain packets are serviced and transmitted before other packets. In addition, the queue settings and the packet's loss priority setting determine which packets might be dropped from the network during periods of congestion.

In addition to queuing the packet, the outgoing I/O Manager ASIC is responsible for ensuring that CoS bits in the packet's header are correctly set before it is transmitted. This rewrite function helps the next downstream router perform its CoS function in the network.

Enhanced CFEB and CoS on M7i and M10i Multiservice Edge Routers

The Enhanced Compact Forwarding Engine Board (CFEB-E) for the M7i and M10i Multiservice Edge Routers provides additional hardware performance, scaling, and functions, as well as enhanced CoS software capabilities.

The enhanced CoS functions available with the CFEB-E on M7i and M10i routers include:

- Support for 16 forwarding classes and 8 queues
- Support for four loss priorities (medium-high and medium-low in addition to high and low)
- Support for hierarchical policing with tricolor marking, both single-rate tricolor marking (TCM) and two-rate TCM (trTCM)

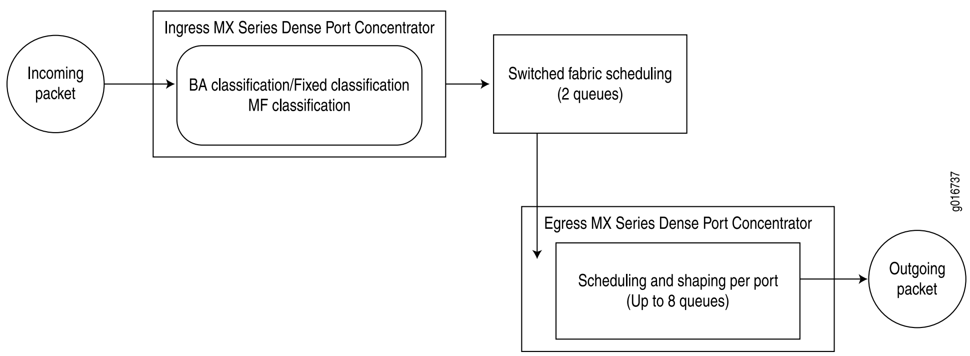
Related Documentation

- Packet Flow Within Routers Overview on page 10
- Packet Flow on Juniper Networks J Series Services Routers on page 11
- Packet Flow on MX Series Ethernet Services Routers on page 13
- Packet Flow on Juniper Networks T Series Core Routers on page 17

Packet Flow on MX Series Ethernet Services Routers

The CoS architecture for MX Series Ethernet Services Routers, such as the MX960 router, is in concept similar to, but in particulars different from, other routers. The general architecture for MX Series routers is shown in Figure 3 on page 13.Figure 3 on page 13 illustrates packet flow through a Dense Port Concentrator (DPC).

Figure 3: MX Series Router Packet Forwarding and Data Flow



NOTE: All Layer 3 Junos OS CoS functions are supported on the MX Series routers. In addition, Layer 3 CoS capabilities, with the exception of traffic shaping, are supported on virtual LANs (VLANs) that span multiple ports.

The MX Series router can be equipped with Dense Port Concentrators (DPCs), Flexible PIC Concentrators (FPCs) and associated Physical Interface Cards (PICs), or Trio Modular Port Concentrators (MPCs) and associated Modular Interface Cards (MICs). In all cases, the command-line interface (CLI) configuration syntax refers to FPCs, PICs, and ports (***type-fpc/pic/port***).

NOTE: The MX80 router is a single-board router with a built-in Routing Engine and one Packet Forwarding Engine, which can have up to four Modular Interface Cards (MICs) attached to it. The Packet Forwarding Engine has two "pseudo" Flexible PIC Concentrators (FPC 0 and FPC1). Because there is no switching fabric, the single Packet Forwarding Engine takes care of both ingress and egress packet forwarding.

.Fixed classification places all packets in the same forwarding class, or the usual multifield or behavior aggregate (BA) classifications can be used to treat packets differently. BA classification with firewall filters can be used for classification based on IP precedence, DSCP, IEEE, or other bits in the frame or packet header.

However, the MX Series routers can also employ multiple BA classifiers on the same logical interface. The logical interfaces do not have to employ the same type of BA classifier. For example, a single logical interface can use classifiers based on IP precedence as well as IEEE 802.1p. If the CoS bits of interest are on the inner VLAN tag of a dual-tagged VLAN interface, the classifier can examine either the inner or outer bits. (By default, the classification is done based on the outer VLAN tag.)

Internal fabric scheduling is based on only two queues: high and low priority. Strict-high priority queuing is also supported in the high-priority category.

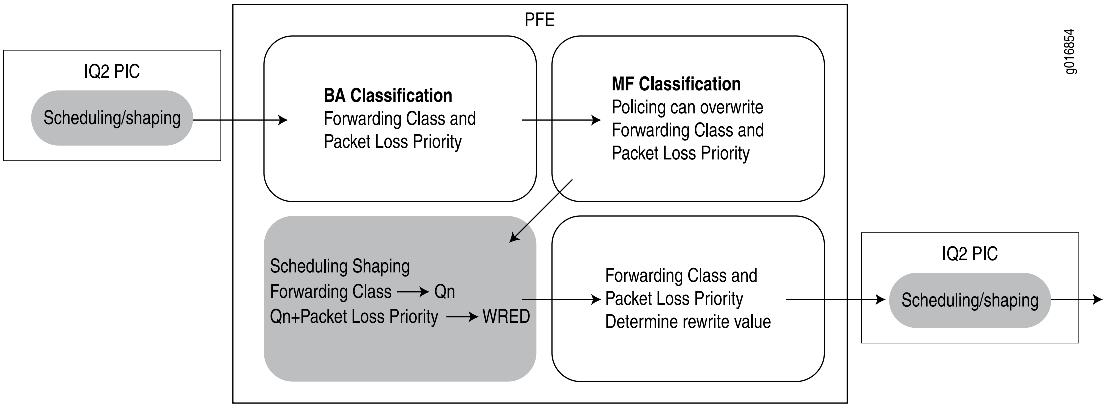
Egress port scheduling supports up to eight queues per port using a form of round-robin queue servicing. The supported priority levels are strict-high, high, medium-high, medium-low, and low. The MX Series router architecture supports both early discard and tail drop on the queues.

All CoS features are supported at line rate.

The fundamental flow of a packet subjected to CoS is different in the MX Series router with integrated chips than it is in the M Series Multiservice Edge Router and T Series Core Router, which have a different packet-handling architecture.

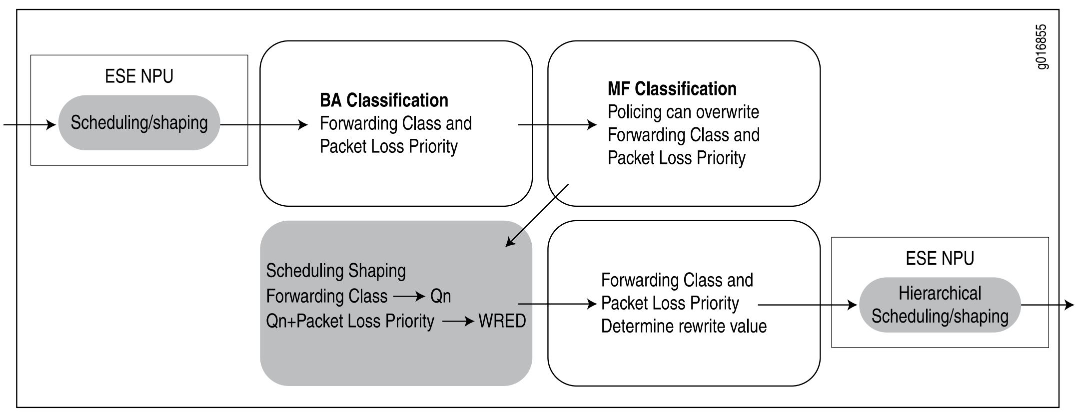
The way that a packet makes its way through an M Series or T Series router with Intelligent Queuing 2 (IQ2) PICs is shown in Figure 4 on page 15. Note that the per-VLAN scheduling and shaping are done on the PIC whereas all other CoS functions at the port level are performed on the Packet Forwarding Engine.

Figure 4: Packet Handling on the M Series and T Series Routers



The way that a packet makes its way through an MX Series router is shown in Figure 5 on page 15. Note that the scheduling and shaping are done with an integrated architecture along with all other CoS functions. In particular, scheduling and shaping are done on the Ethernet services engine network processing unit (ESE NPU). Hierarchical scheduling is supported on the output side as well as the input side.

Figure 5: Packet Handling on the MX Series Routers



Related Documentation

- Packet Flow Within Routers Overview on page 10
- Packet Flow on Juniper Networks J Series Services Routers on page 11
- Packet Flow on Juniper Networks M Series Multiservice Edge Routers on page 11
- Example of Packet Flow on MX Series Ethernet Services Routers on page 15
- Packet Flow on Juniper Networks T Series Core Routers on page 17

Example of Packet Flow on MX Series Ethernet Services Routers

MX Series routers, especially the MX960 Ethernet Services Router, have several features that differ from the usual CoS features in the Junos OS.

The MX960 router allows fixed classification of traffic. All packets on a logical interface can be put into the same forwarding class:

```
[edit class-of-service interfaces ge-1/0/0 unit 0]
forwarding-class af;
```

As on other routers, the MX Series routers allow BA classification, the classifying of packets into different forwarding classes (up to eight) based on a value in the packet header. However, MX Series routers allow a mixture of BA classifiers (IEEE 802.1p and others) for logical interfaces on the same port, as shown in the following example:

```
[edit class-of-service interfaces ge-0/0/0 unit 0]
classifiers {
  inet-precedence IPPRCE-BA-1;
  ieee-802.1 DOT1P-BA-1;
}
```

In this case, the IEEE classifier is applied to Layer 2 traffic and the Internet precedence classifier is applied to Layer 3 (IP) traffic. The IEEE classifier can also perform BA classification based on the bits of either the inner or outer VLAN tag on a dual-tagged logical interface, as shown in the following example:

```
[edit class-of-service interfaces ge-0/0/0]
unit 0 {
  classifiers {
    ieee-802.1 DOT1-BA-1 {
      vlan-tag inner;
    }
  }
}
unit 1 {
  classifiers {
    ieee-802.1 DOT1-BA-1 {
      vlan-tag outer;
    }
  }
}
```

The default action is based on the outer VLAN tag's IEEE precedence bits.

As on other routers, the BA classification can be overridden with a multifield classifier in the action part of a firewall filter. Rewrites are handled as on other routers, but MX Series routers support classifications and rewrites for aggregated Ethernet (**ae-**) logical interfaces.

On MX Series routers, the 64 classifier limit is a theoretical upper limit. In practice, you cannot configure 64 classifiers. Three values are used internally by the default IP precedence, IPv6, and EXP classifiers. Two other classifiers are used for forwarding class and queue operations. This leaves 59 classifiers for configuration purposes. If you configure Differentiated Services code point (DSCP) rewrites for MPLS, the maximum number of classifiers you can configure is less than 59.

On MX Series routers, IEEE 802.1 classifier bit rewrites are determined by forwarding class and packet priority, not by queue number and packet priority as on other routers.

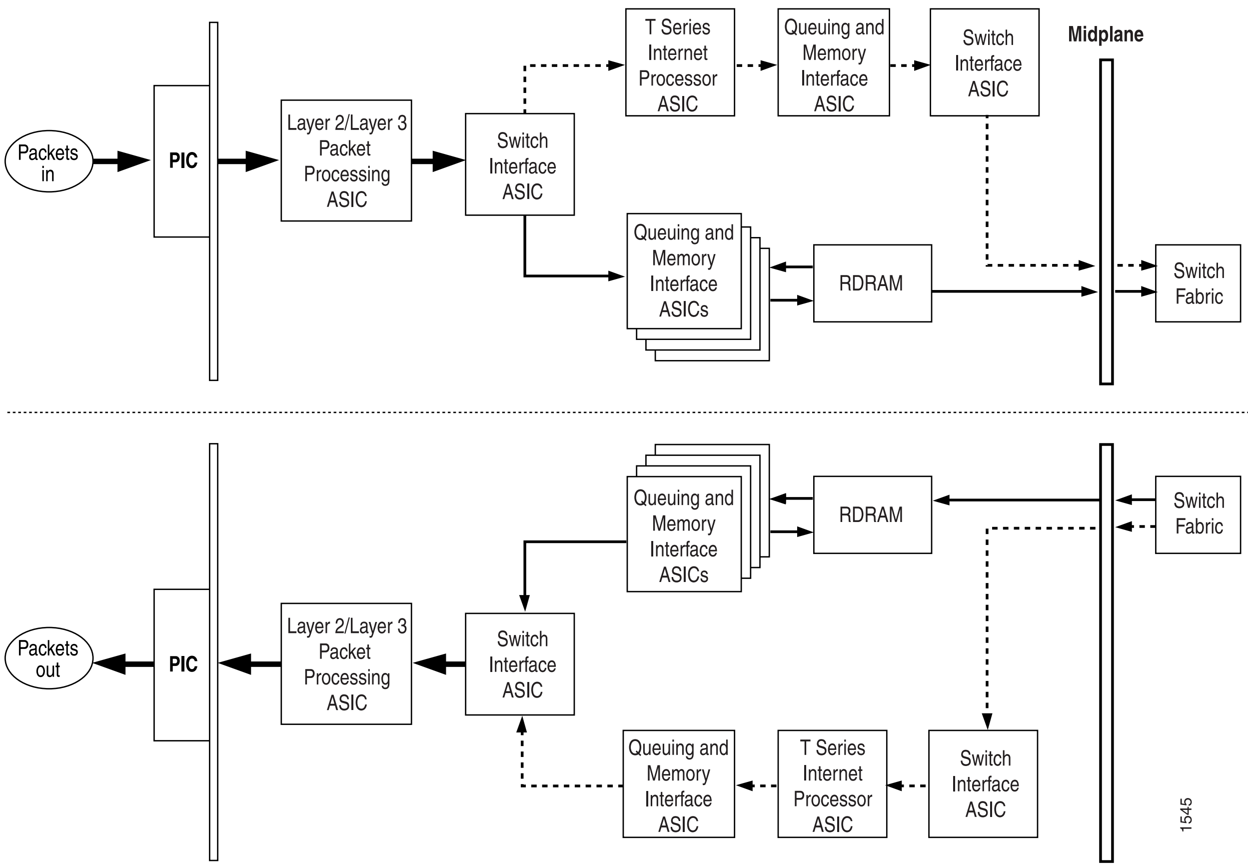
Related Documentation

- Packet Flow on Juniper Networks M Series Multiservice Edge Routers on page 11
- Packet Flow on MX Series Ethernet Services Routers on page 13

Packet Flow on Juniper Networks T Series Core Routers

On T Series Core Routers, CoS actions are performed in several locations: the incoming and outgoing Switch Interface ASICs, the T Series router Internet Processor ASIC, and the Queuing and Memory Interface ASICs. These locations are shown in Figure 6 on page 17.

Figure 6: T Series Router Packet Forwarding Engine Components and Data Flow



This topic describes the packet flow through the following components in more detail:

- Incoming Switch Interface ASICs on page 17
- T Series Routers Internet Processor ASIC on page 18
- Queuing and Memory Interface ASICs on page 18
- Outgoing Switch Interface ASICs on page 18

Incoming Switch Interface ASICs

When a data packet is passed from the receiving interface to its connected FPC, it is received by the incoming Switch Interface ASIC on that specific FPC. During the processing of the packet by this ASIC, the information in the packet's header is examined by a BA classifier. This classification action associates the packet with a particular forwarding class. In addition, the value of the packet's loss priority bit is set by this classifier. Both the forwarding class and loss priority information are placed into the notification cell, which is then transmitted to the T Series router Internet Processor ASIC.

T Series Routers Internet Processor ASIC

The T Series router Internet Processor ASIC receives notification cells representing inbound data packets and performs route lookups in the forwarding table. This lookup determines the outgoing interface on the router and the next-hop IP address for the data packet. While the packet is being processed by the T Series router Internet Processor ASIC, it might also be evaluated by a firewall filter, which is configured on either the incoming or outgoing interface. This filter can perform the functions of a multifield classifier by matching on multiple elements within the packet and overwriting the forwarding class settings, loss priority settings, or both within the notification cell. Once the route lookup and filter evaluations are complete, the notification cell, now called the result cell, is passed to the Queuing and Memory Interface ASICs.

Queuing and Memory Interface ASICs

The Queuing and Memory Interface ASICs pass the data cells to memory for buffering. The data cells are placed into a queue to await transmission on the physical media. The specific queue used by the ASICs is determined by the forwarding class associated with the data packet. The configuration of the queue itself helps determine the service the packet receives while in this queued state. This functionality guarantees that certain packets are serviced and transmitted before other packets. In addition, the queue settings and the packet's loss priority setting determine which packets might be dropped from the network during periods of congestion.

In addition to queuing the packet, the outgoing I/O Manager ASIC is responsible for ensuring that CoS bits in the packet's header are correctly set before it is transmitted. This rewrite function helps the next downstream router perform its CoS function in the network.

The Queuing and Memory Interface ASIC sends the notification to the Switch Interface ASIC facing the switch fabric, unless the destination is on the same Packet Forwarding Engine. In this case, the notification is sent back to the Switch Interface ASIC facing the outgoing ports, and the packets are sent to the outgoing port without passing through the switch fabric. The default behavior is for fabric priority queuing on egress interfaces to match the scheduling priority you assign. High-priority egress traffic is automatically assigned to high-priority fabric queues.

The Queuing and Memory Interface ASIC forwards the notification, including next-hop information, to the outgoing Switch Interface ASIC.

Outgoing Switch Interface ASICs

The destination Switch Interface ASIC sends bandwidth grants through the switch fabric to the originating Switch Interface ASIC. The Queuing and Memory Interface ASIC forwards the notification, including next-hop information, to the Switch Interface ASIC. The Switch Interface ASIC sends read requests to the Queuing and Memory Interface ASIC to read the data cells out of memory, and passes the cells to the Layer 2 or Layer 3 Packet Processing ASIC. The Layer 2 or Layer 3 Packet Processing ASIC reassembles the data cells into packets, adds Layer 2 encapsulation, and sends the packets to the outgoing PIC interface. The outgoing PIC sends the packets out into the network.

Related Documentation

- Packet Flow Within Routers Overview on page 10
- Packet Flow on Juniper Networks J Series Services Routers on page 11
- Packet Flow on Juniper Networks M Series Multiservice Edge Routers on page 11
- Packet Flow on MX Series Ethernet Services Routers on page 13

Packet Flow Through the CoS Process

This topic consists of the following:

- Packet Flow Through the CoS Process Overview on page 19
- Packet Flow Through the CoS Process Configuration Example on page 21

Packet Flow Through the CoS Process Overview

Perhaps the best way to understand Junos CoS is to examine how a packet is treated on its way through the CoS process. This topic includes a description of each step and figures illustrating the process.

The following steps describe the CoS process:

1. A logical interface has one or more classifiers of different types applied to it (at the **[edit class-of-service interfaces]** hierarchy level). The types of classifiers are based on which part of the incoming packet the classifier examines (for example, EXP bits, IEEE 802.1p bits, or DSCP bits). You can use a translation table to rewrite the values of these bits on ingress.

   NOTE: You can only rewrite the values of these bits on ingress on the Juniper Networks M40e, M120, M320 Multiservice Edge Routers, and T Series Core Routers with IQE PICs. For more information about rewriting the values of these bits on ingress, see "Configuring ToS Translation Tables" on page 291.

2. The classifier assigns the packet to a forwarding class and a loss priority (at the **[edit class-of-service classifiers]** hierarchy level).
3. Each forwarding class is assigned to a queue (at the **[edit class-of-service forwarding-classes]** hierarchy level).
4. Input (and output) policers meter traffic and might change the forwarding class and loss priority if a traffic flow exceeds its service level.
5. The physical or logical interface has a scheduler map applied to it (at the **[edit class-of-service interfaces]** hierarchy level).

At the **[edit class-of-service interfaces]** hierarchy level, the **scheduler-map** and **rewrite-rules** statements affect the outgoing packets, and the **classifiers** statement affects the incoming packets.

6. The scheduler defines how traffic is treated in the output queue—for example, the transmit rate, buffer size, priority, and drop profile (at the **[edit class-of-service schedulers]** hierarchy level).
7. The scheduler map assigns a scheduler to each forwarding class (at the **[edit class-of-service scheduler-maps]** hierarchy level).
8. The drop-profile defines how aggressively to drop packets that are using a particular scheduler (at the **[edit class-of-service drop-profiles]** hierarchy level).
9. The rewrite rule takes effect as the packet leaves a logical interface that has a rewrite rule configured (at the **[edit class-of-service rewrite-rules]** hierarchy level). 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.

Figure 7 on page 20 and Figure 8 on page 21 show the components of the Junos CoS features, illustrating the sequence in which they interact.

Figure 7: CoS Classifier, Queues, and Scheduler

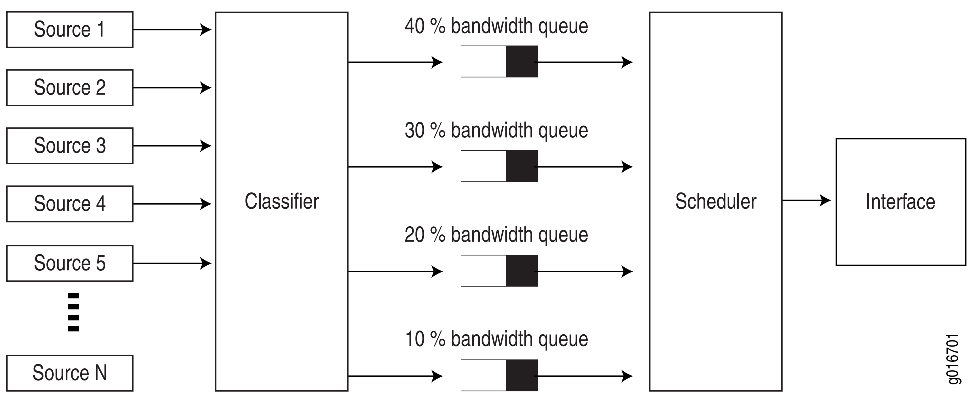
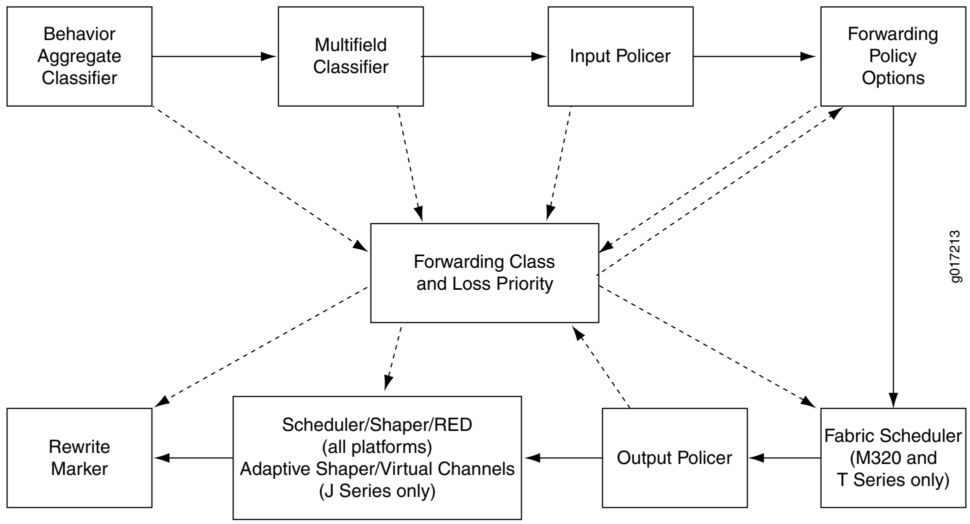


Figure 8: Packet Flow Through CoS Configurable Components



Each outer box in Figure 8 on page 21 represents a process component. The components in the upper row apply to inbound packets, and the components in the lower row apply to outbound packets. The arrows with the solid lines point in the direction of packet flow.

The middle box (forwarding class and loss priority) represents two data values that can either be inputs to or outputs of the process components. The arrows with the dotted lines indicate inputs and outputs (or settings and actions based on settings). For example, the multifield classifier sets the forwarding class and loss priority of incoming packets. This means that the forwarding class and loss priority are outputs of the classifier; thus, the arrow points away from the classifier. The scheduler receives the forwarding class and loss priority settings, and queues the outgoing packet based on those settings. This means that the forwarding class and loss priority are inputs to the scheduler; thus, the arrow points to the scheduler.

Typically, only a combination of some components (not all) is used to define a CoS service offering.

Related Documentation

- Packet Flow Through the CoS Process Configuration Example on page 21

Packet Flow Through the CoS Process Configuration Example

The following configuration demonstrates the packet flow through the CoS process:

```
[edit class-of-service]
interfaces { # Step 1: Define CoS interfaces.
  so-* {
    scheduler-map sched1;
    unit 0 {
      classifiers {
        exp exp_classifier;
      }
    }
  }
```



```
  t3-* {
    scheduler-map sched1;
    unit 0 {
      classifiers {
        exp exp_classifier;
      }
    }
  }
}
classifiers { # Step 2: Define classifiers.
exp exp_classifier {
  forwarding-class data-queue {
    loss-priority low code-points 000;
    loss-priority high code-points 001;
  }
  forwarding-class video-queue {
    loss-priority low code-points 010;
    loss-priority high code-points 011;
  }
  forwarding-class voice-queue {
    loss-priority low code-points 100;
    loss-priority high code-points 101;
  }
  forwarding-class nc-queue {
    loss-priority high code-points 111;
    loss-priority low code-points 110;
  }
}
drop-profiles { # Step 3: Define drop profiles.
  be-red {
    fill-level 50 drop-probability 100;
  }
}
forwarding-classes { # Step 4: Define queues.
  queue 0 data-queue;
  queue 1 video-queue;
  queue 2 voice-queue;
  queue 3 nc-queue;
}
schedulers { # Step 5: Define schedulers.
  data-scheduler {
    transmit-rate percent 50;
    buffer-size percent 50;
    priority low;
    drop-profile-map loss-priority high protocol any drop-profile be-red;
  }
  video-scheduler {
    transmit-rate percent 25;
    buffer-size percent 25;
    priority strict-high;
  }
  voice-scheduler {
    transmit-rate percent 20;
    buffer-size percent 20;
    priority high;
  }
```



```
  nc-scheduler {
    transmit-rate percent 5;
    buffer-size percent 5;
    priority high;
  }
}
scheduler-maps { # Step 6: Define scheduler maps.
  sched1 {
    forwarding-class data-queue scheduler data-scheduler;
    forwarding-class video-queue scheduler video-scheduler;
    forwarding-class voice-queue scheduler voice-scheduler;
    forwarding-class nc-queue scheduler nc-scheduler;
  }
}
```

Related Documentation

- Packet Flow Through the CoS Process Overview on page 19

CoS Applications Overview

You can configure CoS features to meet your application needs. Because the components are generic, you can use a single CoS configuration syntax across multiple routers. CoS mechanisms are useful for two broad classes of applications. These applications can be referred to as *in the box* and *across the network*.

In-the-box applications use CoS mechanisms to provide special treatment for packets passing through a single node on the network. You can monitor the incoming traffic on each interface, using CoS to provide preferred service to some interfaces (that is, to some customers) while limiting the service provided to other interfaces. You can also filter outgoing traffic by the packet's destination, thus providing preferred service to some destinations.

Across-the-network applications use CoS mechanisms to provide differentiated treatment to different classes of packets across a set of nodes in a network. In these types of applications, you typically control the ingress and egress routers to a routing domain and all the routers within the domain. You can use Junos CoS features to modify packets traveling through the domain to indicate the packet's priority across the domain.

Specifically, you modify the CoS code points in packet headers, remapping these bits to values that correspond to levels of service. When all routers in the domain are configured to associate the precedence bits with specific service levels, packets traveling across the domain receive the same level of service from the ingress point to the egress point. For CoS to work in this case, the mapping between the precedence bits and service levels must be identical across all routers in the domain.

Junos CoS applications support the following range of mechanisms:

- Differentiated Services (DiffServ)—The CoS application supports DiffServ, which uses 6-bit IPv4 and IPv6 header type-of-service (ToS) byte settings. The configuration uses CoS values in the IP and IPv6 ToS fields to determine the forwarding class associated with each packet.

- Layer 2 to Layer 3 CoS mapping—The CoS application supports mapping of Layer 2 (IEEE 802.1p) packet headers to router forwarding class and loss-priority values.

Layer 2 to Layer 3 CoS mapping involves setting the forwarding class and loss priority based on information in the Layer 2 header. Output involves mapping the forwarding class and loss priority to a Layer 2-specific marking. You can mark the Layer 2 and Layer 3 headers simultaneously.

- MPLS EXP—Supports configuration of mapping of MPLS experimental (EXP) bit settings to router forwarding classes and vice versa.
- VPN outer-label marking—Supports setting of outer-label EXP bits, also known as CoS bits, based on MPLS EXP mapping.

Interface Types That Do Not Support CoS

For original Channelized OC12 PICs, limited CoS functionality is supported. For more information, contact Juniper Networks customer support.

The standard Junos CoS hierarchy is not supported on ATM interfaces. ATM has traffic-shaping capabilities that would override CoS, because ATM traffic shaping is performed at the ATM layer and CoS is performed at the IP layer. For more information about ATM traffic shaping and ATM CoS components, see the *Junos OS Network Interfaces Configuration Guide*.

NOTE: Transmission scheduling is not supported on 8-port, 12-port, and 48-port Fast Ethernet PICs.

You can configure CoS on all interfaces, except the following:

- **cau4**—Channelized STM1 IQ interface (configured on the Channelized STM1 IQ PIC).
- **coc1**—Channelized OC1 IQ interface (configured on the Channelized OC12 IQ PIC).
- **coc12**—Channelized OC12 IQ interface (configured on the Channelized OC12 IQ PIC).
- **cstm-1**—Channelized STM1 IQ interface (configured on the Channelized STM1 IQ PIC).
- **ct1**—Channelized T1 IQ interface (configured on the Channelized DS3 IQ PIC or Channelized OC12 IQ PIC).
- **ct3**—Channelized T3 IQ interface (configured on the Channelized DS3 IQ PIC or Channelized OC12 IQ PIC).
- **ce1**—Channelized E1 IQ interface (configured on the Channelized E1 IQ PIC or Channelized STM1 IQ PIC).
- **dsc**—Discard interface.
- **fxp**—Management and internal Ethernet interfaces.
- **lo**—Loopback interface. This interface is internally generated.

- **pe**—Encapsulates packets destined for the rendezvous point router. This interface is present on the first-hop router.
- **pd**—De-encapsulates packets at the rendezvous point. This interface is present on the rendezvous point.
- **vt**—Virtual loopback tunnel interface.

NOTE: For channelized interfaces, you can configure CoS on channels, but not at the controller level. For a complex configuration example, see the *Junos OS Feature Guide*.

Related Documentation

- CoS on ATM Interfaces Overview on page 411

VPLS and Default CoS Classification

A VPLS routing instance with the **no-tunnel-services** option configured has a default classifier applied to the label-switched interface for all VPLS packets coming from the remote VPLS PE. This default classifier is modifiable only on MX Series routers.

For example, on routers with eight queues (Juniper Networks M120 and M320 Multiservice Edge Routers, MX Series Ethernet Services Routers, and T Series Core Routers), the default classification applied to **no-tunnel-services** VPLS packets are shown in Table 4 on page 25.

Table 4: Default VPLS Classifiers

MPLS Label EXP Bits	Forwarding Class/Queue
000	0
001	1
010	2
011	3
100	4
101	5
110	6
111	7

NOTE: Forwarding class to queue number mapping is not always one-to-one. Forwarding classes and queues are only the same when default forwarding-class-to-queue mapping is in effect. For more information about configuring forwarding class and queues, see “Configuring Forwarding Classes” on page 115.

On MX Series routers, VPLS filters and policers act on a Layer 2 frame that includes the media access control (MAC) header (after any VLAN rewrite or other rules are applied), but does not include the cyclical redundancy check (CRC) field.

NOTE: On MX Series routers, if you apply a counter in a firewall for egress MPLS or VPLS packets with the EXP bits set to 0, the counter will not tally these packets.

CHAPTER 2

Class of Service Configuration Statements

This topic shows the complete configuration statement hierarchy for class of service (CoS), listing all possible configuration statements and showing their level in the configuration hierarchy. When you are configuring the Junos OS, your current hierarchy level is shown in the banner on the line preceding the **user@host#** prompt.

For a complete list of the Junos configuration statements, see the *Junos OS Hierarchy and RFC Reference*.

This topic is organized as follows:

[edit chassis] Hierarchy Level

This topic shows the complete configuration for class of service (CoS) statements for the **[edit chassis]** hierarchy level, listing all possible configuration statements and showing their level in the configuration hierarchy. When you are configuring the Junos OS, your current hierarchy level is shown in the banner on the line preceding the **user@host#** prompt.

For a complete list of the Junos configuration statements, see the *Junos OS Hierarchy and RFC Reference*.

This is not a comprehensive list of statements available at the **[edit chassis]** hierarchy level. Only the statements that are also documented in this manual are listed here. For more information about chassis configuration, see the *Junos OS System Basics Configuration Guide*.

```
[edit chassis]
fpc slot-number {
  pic pic-number {
    max-queues-per-interface (4 | 8);
    q-pic-large-buffer {
      [ large-scale | small-scale ];
```

```
        }
        red-buffer-occupancy {
          weighted-averaged [ instant-usage-weight-exponent weight-value ];
        }
        traffic-manager {
          egress-shaping-overhead number;
          ingress-shaping-overhead number;
          mode session-shaping;
        }
      }
    }
```

[edit class-of-service] Hierarchy Level

This topic shows the complete configuration for class of service (CoS) statements for the **[edit class-of-servies]** hierarchy level, listing all possible configuration statements and showing their level in the configuration hierarchy. When you are configuring the Junos OS, your current hierarchy level is shown in the banner on the line preceding the **user@host#** prompt.

For a complete list of the Junos configuration statements, see the *Junos OS Hierarchy and RFC Reference*.

```
[edit class-of-service]
classifiers {
  (dscp | dscp-ipv6 | exp | ieee-802.1 | inet-precedence) classifier-name {
    import (classifier-name | default);
    forwarding-class class-name {
      loss-priority level code-points [ aliases ] [ bit-patterns ];
    }
  }
}
code-point-aliases {
  (dscp | dscp-ipv6 | exp | ieee-802.1 | inet-precedence) {
    alias-name bits;
  }
}
copy-plp-all;
drop-profiles {
  profile-name {
    fill-level percentage drop-probability percentage;
    interpolate {
      drop-probability [ values ];
      fill-level [ values ];
    }
  }
}
fabric {
  scheduler-map {
    priority (high | low) scheduler scheduler-name;
  }
}
forwarding-classes {
  class class-name queue-num queue-number priority (high | low);
  queue queue-number class-name priority (high | low) [ policing-priority (high | low) ];
```



```
}
forwarding-classes-interface-specific  forwarding-class-map-name {
  class class-name queue-num queue-number [ restricted-queue queue-number ];
}
forwarding-policy {
  next-hop-map map-name {
    forwarding-class class-name {
      next-hop [ next-hop-name ];
      lsp-next-hop [ lsp-regular-expression ];
      non-lsp-next-hop;
      discard;
    }
  }
  class class-name {
    classification-override {
      forwarding-class class-name;
    }
  }
}
fragmentation-maps {
  map-name {
    forwarding-class class-name {
      drop-timeout milliseconds;
      fragment-threshold bytes;
      multilink-class number;
      no-fragmentation;
    }
  }
}
host-outbound-traffic {
  forwarding-class class-name;
  dscp-code-point value;
}
interfaces {
  interface-name {
    input-scheduler-map map-name;
    input-shaping-rate rate;
    irb {
      unit logical-unit-number {
        classifiers {
          type (classifier-name | default) family (mpls | all);
        }
        rewrite-rules {
          dscp (rewrite-name | default);
          dscp-ipv6 (rewrite-name | default);
          exp (rewrite-name | default)protocol protocol-types;
          ieee-802.1 (rewrite-name | default) vlan-tag (outer | outer-and-inner);
          inet-precedence (rewrite-name | default);
        }
      }
    }
    output-forwarding-class-map forwarding-class-map-name;
    member-link-scheduler (replicate | scale);
    scheduler-map map-name;
    scheduler-map-chassis map-name;
    shaping-rate rate;
```

```
        unit logical-unit-number {
            classifiers {
                (dscp | dscp-ipv6 | exp | ieee-802.1 | inet-precedence) (classifier-name | default)
                    family (mpls | inet);
            }
            forwarding-class class-name;
            fragmentation-map map-name;
            input-scheduler-map map-name;
            input-shaping-rate (percent percentage | rate);
            input-traffic-control-profile profile-name shared-instance instance-name;
            output-traffic-control-profile profile-name shared-instance instance-name;
            per-session-scheduler;
            rewrite-rules {
                dscp (rewrite-name | default)protocol protocol-types;
                dscp-ipv6 (rewrite-name | default);
                exp (rewrite-name | default)protocol protocol-types;
                exp-push-push-push default;
                exp-swap-push-push default;
                ieee-802.1 (rewrite-name | default) vlan-tag (outer | outer-and-inner);
                inet-precedence (rewrite-name | default)protocol protocol-types;
            }
            scheduler-map map-name;
            shaping-rate rate;
            translation-table (to-dscp-from-dscp | to-dscp-ipv6-from-dscp-ipv6 |
                to-exp-from-exp | to-inet-precedence-from-inet-precedence) table-name;
        }
    }
}
restricted-queues {
    forwarding-class class-name queue queue-number;
}
rewrite-rules {
    (dscp | dscp-ipv6 | exp | ieee-802.1 | inet-precedence) rewrite-name {
        import (rewrite-name | default);
        forwarding-class class-name {
            loss-priority level code-point (alias | bits);
        }
    }
}
routing-instances routing-instance-name {
    classifiers {
        exp (classifier-name | default);
        dscp (classifier-name | default);
        dscp-ipv6 (classifier-name | default);
    }
}
scheduler-maps {
    map-name {
        forwarding-class class-name scheduler scheduler-name;
    }
}
schedulers {
    scheduler-name {
        buffer-size (percent percentage | remainder | temporal microseconds);
        drop-profile-map loss-priority (any | low | medium-low | medium-high | high)protocol
            (any | non-tcp | tcp) drop-profile profile-name;
```



```
    excess-priority (low | high);
    excess-rate percent percentage;
    priority priority-level;
    transmit-rate (rate | percent percentage | remainder) <exact | rate-limit>;
  }
}
traffic-control-profiles profile-name {
  delay-buffer-rate (percent percentage | rate);
  excess-rate (percent percentage | proportion value);
  guaranteed-rate (percent percentage | rate);
  overhead-accounting (frame-mode | cell-mode) <bytes byte-value>;
  scheduler-map map-name;
  shaping-rate (percent percentage | rate);
}
translation-table {
  (to-dscp-from-dscp | to-dscp-ipv6-from-dscp-ipv6 | to-exp-from-exp |
    to-inet-precedence-from-inet-precedence) table-name {
    to-code-point value from-code-points (* | [ values ]);
  }
}
tri-color;
```

On a Juniper Networks MX Series Ethernet Services Routers with Enhanced Queuing DPCs, you can configure the following CoS statements at the **[edit class-of-service interfaces]** hierarchy level:

```
interface-set interface-set-name {
  excess-bandwith-share (proportional value | equal);
  internal-node;
  traffic-control-profiles profile-name;
  output-traffic-control-profile-remaining profile-name;
}
```

[edit firewall] Hierarchy Level

The following CoS statements can be configured at the **[edit firewall]** hierarchy level. This is not a comprehensive list of statements available at the **[edit firewall]** hierarchy level. Only the statements documented in this manual are listed here. For more information about firewall configuration, see the *Junos OS Policy Framework Configuration Guide*.

```
[edit firewall]
family family-name {
  filter filter-name {
    term term-name {
      from {
        match-conditions;
      }
      then {
        dscp 0;
        forwarding-class class-name;
        loss-priority (high | low);
        three-color-policer {
          (single-rate | two-rate) policer-name;
        }
```

```
            }
          }
        }
        simple-filter filter-name {
          term term-name {
            from {
              match-conditions;
            }
            then {
              forwarding-class class-name;
              loss-priority (high | low | medium);
            }
          }
        }
      }
      policer policer-name {
        logical-bandwidth-policer;
        if-exceeding {
          bandwidth-limit rate;
          bandwidth-percent number;
          burst-size-limit bytes;
        }
        then {
          policer-action;
        }
      }
      three-color-policer policer-name {
        action {
          loss-priority high then discard;
        }
        logical-interface-policer;
        single-rate {
          (color-aware | color-blind);
          committed-information-rate bps;
          committed-burst-size bytes;
          excess-burst-size bytes;
        }
        two-rate {
          (color-aware | color-blind);
          committed-information-rate bps;
          committed-burst-size bytes;
          peak-information-rate bps;
          peak-burst-size bytes;
        }
      }
```

[edit interfaces] Hierarchy Level

The following CoS statements can be configured at the **[edit interfaces]** hierarchy level. This is not a comprehensive list of statements available at the **[edit interfaces]** hierarchy level. Only the statements that are also documented in this manual are listed here. For more information about interface configuration, see the *Junos OS Network Interfaces Configuration Guide*.

```
[edit interfaces]
```



```
interface-name {
  atm-options {
    linear-red-profiles profile-name {
      high-plp-max-threshold percent;
      low-plp-max-threshold percent;
      queue-depth cells high-plp-threshold percent low-plp-threshold percent;
    }
    plp-to-clp;
    scheduler-maps map-name {
      forwarding-class class-name {
        epd-threshold cells plp1 cells;
        linear-red-profile profile-name;
        priority (high | low);
        transmit-weight (cells number | percent number);
      }
      vc-cos-mode (alternate | strict);
    }
  }
  per-unit-scheduler;
  shared-scheduler;
  schedulers number;
  unit logical-unit-number {
    atm-scheduler-map (map-name | default);
    copy-tos-to-outer-ip-header;
    family family {
      address address {
        destination address;
      }
      filter {
        input filter-name;
        output filter-name;
      }
      policer {
        input policer-name;
        output policer-name;
      }
      simple-filter {
        input filter-name;
      }
    }
    layer2-policer {
      input-policer policer-name;
      input-three-color policer-name;
      output-policer policer-name;
      output-three-color policer-name;
    }
    plp-to-clp;
    shaping {
      (cbr rate | rtvbr peak rate sustained rate burst length | vbr peak rate sustained rate
        burst length);
    }
    vci vpi-identifier.vci-identifier;
  }
}
```

On the Juniper Networks MX Series Ethernet Services Routers with Enhanced Queuing DPCs and on M Series and T Series routers with IQ2E PIC, you can configure the following CoS statements at the [edit interfaces] hierarchy level:

```
hierarchical-scheduler;
interface-set interface-set-name {
  ethernet-interface-name {
    [interface-parameters];
  }
}
```

[edit services cos] Hierarchy Level

The following CoS statements can be configured at the **[edit services cos]** hierarchy level. This is not a comprehensive list of statements available at the **[edit services cos]** hierarchy level. Only the statements documented in this manual are listed here. For more information about services configuration, see the *Junos OS Services Interfaces Configuration Guide*.

```
[edit services cos]
application-profile profile-name {
  ftp {
    data {
      dscp (alias | bits);
      forwarding-class class-name;
    }
  }
  sip {
    video {
      dscp (alias | bits);
      forwarding-class class-name;
    }
    voice {
      dscp (alias | bits);
      forwarding-class class-name;
    }
  }
}
rule rule-name {
  match-direction (input | output | input-output);
  term term-name {
    from {
      applications [ application-names ];
      application-sets [ set-names ];
      destination-address address;
      source-address address;
    }
    then {
      application-profile profile-name;
      dscp (alias | bits);
      forwarding-class class-name;
      syslog;
      (reflexive | reverse) {
        application-profile profile-name;
        dscp (alias | bits);
```

```
            forwarding-class class-name;
            syslog;
        }
    }
  }
}
rule-set rule-set-name {
  [ rule rule-names ];
}
```

PART 2

CoS Configuration Components

CHAPTER 3

Classifying Packets by Behavior Aggregate

This topic discusses the following:

BA Classifier Overview

The behavior aggregate (BA) classifier maps a class-of-service (CoS) value to a forwarding class and loss priority. The forwarding class determines the output queue. The loss priority is used by schedulers in conjunction with the random early discard (RED) algorithm to control packet discard during periods of congestion.

The types of BA classifiers are based on which part of the incoming packet the classifier examines:

- Differentiated Services code point (DSCP) for IP DiffServ
- DSCP for IPv6 DiffServ

- IP precedence bits
- MPLS EXP bits
- IEEE 802.1p CoS bits
- IEEE 802.1ad drop eligible indicator (DEI) bit

Unlike multifield classifiers (which are discussed in "Multifield Classifier Overview" on page 71), BA classifiers are based on fixed-length fields, which makes them computationally more efficient than multifield classifiers. For this reason, core devices are normally configured to perform BA classification, because of the higher traffic volumes they handle.

In most cases, you need to rewrite a given marker (IP precedence, DSCP, IEEE 802.1p, IEEE 802.1ad, or MPLS EXP settings) at the ingress node to accommodate BA classification by core and egress devices. For more information about rewrite markers, see "Rewriting Packet Header Information Overview" on page 239.

For Juniper Networks M Series Multiservice Edge Routers, four classes can forward traffic independently. For M320 Multiservice Edge Routers and T Series Core Routers, eight classes can forward traffic independently. Therefore, you must configure additional classes to be aggregated into one of these classes. You use the BA classifier to configure class aggregation.

For MX Series Ethernet Services Routers and Intelligent Queuing 2 (IQ2) PICs, the following restrictions apply:

- You can only use multifield classifiers for IPv4 DSCP bits for virtual private LAN service (VPLS).
- You cannot use BA classifiers for IPv4 DSCP bits for Layer 2 VPNs.
- You cannot use BA classifiers for IPv6 DSCP bits for VPLS.
- You cannot use BA classifiers for IPv6 DSCP bits for Layer 2 VPNs.

For the 10-port 10-Gigabit Oversubscribed Ethernet (OSE) PICs, the following restrictions on BA classifiers apply:

- Multiple classifiers can be configured to a single logical interface. However, there are some restrictions on which the classifiers can coexist.

  For example, the DSCP and IP precedence classifiers cannot be configured on the same logical interface. The DSCP and IP precedence classifiers can coexist with the DSCP IPv6 classifier on the same logical interface. An IEEE 802.1 classifier can coexist with other classifiers and is applicable only if a packet does not match any of the configured classifiers. For information about the supported combinations, see "Applying Classifiers to Logical Interfaces" on page 50.

- If the classifiers are not defined explicitly, then the default classifiers are applied as follows:
  - All MPLS packets are classified using the MPLS (EXP) classifier. If there is no explicit MPLS (EXP) classifier, then the default MPLS (EXP) classifier is applied.
  - All IPv4 packets are classified using the IP precedence and DSCP classifiers. If there is no explicit IP precedence and DSCP classifiers, then the default IP precedence classifier is applied.
  - All IPv6 packets are classified using DSCP IPv6 classifier. If there is no explicit DSCP IPv6 classifier, then the default DSCP IPv6 classifier is applied.
  - If the IEEE 802.1p classifier is configured and a packet does not match any explicitly configured classifier, then the IEEE 802.1p classifier is applied.

NOTE: For a specified interface, you can configure both a multifield classifier and a BA classifier without conflicts. Because the classifiers are always applied in sequential order, the BA classifier followed by the multifield classifier, any BA classification result is overridden by an multifield classifier if they conflict. For more information about multifield classifiers, see "Multifield Classifier Overview" on page 71.

For MX Series routers and IQ2 PICs, the following restrictions on BA classifiers apply:

- IPv4 DSCP markings for VPLS are not supported (use multifield classifiers instead)
- IPv4 DSCP markings for Layer2 VPNs are not supported
- IPv6 DSCP markings for VPLS are not supported
- IPv6 DSCP markings for Layer2 VPNs are not supported

BA Classifier Configuration Hierarchy

To configure BA classifiers, include the following statements at the **[edit class-of-service]** hierarchy level:

```
[edit class-of-service]
classifiers {
  (dscp | dscp-ipv6 | exp | ieee-802.1 | ieee-802.1ad | inet-precedence) classifier-name {
    import (classifier-name | default);
    forwarding-class class-name {
      loss-priority level code-points [ aliases ] [ bit-patterns ];
    }
  }
}
interfaces {
  interface-name {
    unit logical-unit-number {
      classifiers {
```



```
            (dscp | dscp-ipv6 | exp | ieee-802.1 | ieee-802.1ad | inet-precedence)
              (classifier-name | default);
          }
        }
      }
    }
    routing-instances routing-instance-name {
      classifiers {
        exp (classifier-name | default);
      }
    }
```

Overview of BA Classifier Types

The idea behind class of service (CoS) is that packets are not treated identically by the routers on the network. In order to selectively apply service classes to specific packets, the packets of interest must be classified in some fashion.

The simplest way to classify a packet is to use behavior aggregate classification. The DSCP, DSCP IPv6, or IP precedence bits of the IP header convey the behavior aggregate class information. The information might also be found in the MPLS EXP bits, IEEE 802.1ad, or or IEEE 802.1p CoS bits.

You can configure the following classifier types:

- DSCP, DSCP IPv6, or IP precedence—IP packet classification (Layer 3 headers)
- MPLS EXP—MPLS packet classification (Layer 2 headers)
- IEEE 802.1p—Packet classification (Layer 2 headers)
- IEEE 802.1ad—Packet classification for IEEE 802.1ad formats (including DEI bit)

If you apply an IEEE 802.1 classifier to a logical interface, this classifier takes precedence and is not compatible with any other classifier type. On Juniper Networks MX Series Ethernet Services Routers using IEEE 802.1ad frame formats, you can apply classification on the basis of the IEEE 802.1p bits (three bits in either the inner virtual LAN (VLAN) tag or the outer VLAN tag) and the drop eligible indicator (DEI) bit. On routers with IQ2 PICs using IEEE 802.1ad frame format, you can apply classification based on the IEEE 802.1p bits and the DEI bit. Classifiers for IP (DSCP or IP precedence) and MPLS (EXP) can coexist on a logical interface if the hardware requirements are met. (See “Applying Classifiers to Logical Interfaces” on page 50.)

The Enhanced Queuing DPC (EQ DPC) does not support BA classification for packets received from a Layer 3 routing interface or a virtual routing and forwarding (VRF) interface and routed to an integrated routing and bridging interface (IRB) (either a local Layer 2 interface or a pseudowire connection). The EQ DPC also does not support BA classification for Layer 2 frames received from a Virtual Private LAN Service (VPLS) or bridging domain and routed to an IRB interface.

Default Behavior Aggregate Classification Overview

The software automatically assigns an implicit default IP precedence classifier to all logical interfaces.

NOTE: Only the IEEE 802.1p classifier is supported in Layer 2 interfaces. You must explicitly apply this classifier to the interface as shown in "Default IEEE 802.1p Classifier" on page 46.

If you enable the MPLS protocol family on a logical interface, a default MPLS EXP classifier is automatically applied to that logical interface.

Other default classifiers (such as those for IEEE 802.1p bits and DSCP) require that you explicitly associate a default classification table with a logical interface. When you explicitly associate a default classifier with a logical interface, you are in effect overriding the implicit default classifier with an explicit default classifier.

NOTE: Although several code points 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 **af1x** are mapped to best-effort, because RFC 2597, *Assured Forwarding PHB Group*, prohibits a node from aggregating classes.

You can apply IEEE 802.1p classifiers to interfaces that are part of VPLS routing instances.

Related Documentation

- Default IP Precedence Classifier (ipprec-compatibility) on page 44
- Default MPLS EXP Classifier on page 44
- Default DSCP and DSCP IPv6 Classifier on page 45
- Default IEEE 802.1p Classifier on page 46
- Default IEEE 802.1ad Classifier on page 47
- Default IP Precedence Classifier (ipprec-default) on page 48

BA Classifier Default Values

Here are the values for these default BA classifiers:

- Default IP Precedence Classifier (ipprec-compatibility) on page 44
- Default MPLS EXP Classifier on page 44
- Default DSCP and DSCP IPv6 Classifier on page 45
- Default IEEE 802.1p Classifier on page 46

- Default IEEE 802.1ad Classifier on page 47
- Default IP Precedence Classifier (ipprec-default) on page 48

Default IP Precedence Classifier (ipprec-compatibility)

By default, all logical interfaces are automatically assigned an implicit IP precedence classifier called **ipprec-compatibility**. The **ipprec-compatibility** IP precedence classifier maps IP precedence bits to forwarding classes and loss priorities, as shown in Table 5 on page 44.

Table 5: Default IP Precedence Classifier

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

Default MPLS EXP Classifier

For all PICs except PICs mounted on Juniper Networks M Series Multiservice Edge Router standard (nonenhanced) FPCs, if you enable the MPLS protocol family on a logical interface, the default MPLS EXP classifier is automatically applied to that logical interface. The default MPLS classifier maps EXP bits to forwarding classes and loss priorities, as shown in Table 6 on page 44.

Table 6: Default MPLS Classifier

Code Point	Forwarding Class	Loss Priority
000	best-effort	low
001	best-effort	high
010	expedited-forwarding	low
011	expedited-forwarding	high
100	assured-forwarding	low

Table 6: Default MPLS Classifier *(continued)*

Code Point	Forwarding Class	Loss Priority
101	assured-forwarding	high
110	network-control	low
111	network-control	high

Default DSCP and DSCP IPv6 Classifier

Table 7 on page 45 shows the forwarding class and packet loss priority (PLP) that are assigned to each well-known DSCP when you apply the explicit default DSCP or DSCP IPv6 classifier. To do this, include the **default** statement at the **[edit class-of-service interfaces** ***interface-name*** **unit** ***logical-unit-number*** **classifiers (dscp | dscp-ipv6)]** hierarchy level:

```
[edit class-of-service interfaces interface-name unit logical-unit-number classifiers (dscp
    | dscp-ipv6)]
default;
```

NOTE: If you deactivate or delete dscp-ipv6 statement from the configuration, the default IPv6 classifier is not activated on the on the M5, M10, M7i, M10i, M20, M40, M40e, and M160 routing platforms. As a workaround, explicitly specify the default option to the dscp-ipv6 statement.

Table 7: Default DSCP Classifier

DSCP and DSCP IPv6	Forwarding Class	PLP
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
af32	best-effort	low
af33	best-effort	low

Table 7: Default DSCP Classifier *(continued)*

DSCP and DSCP IPv6	Forwarding Class	PLP
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

Default IEEE 802.1p Classifier

Table 8 on page 46 shows the forwarding class and PLP that are assigned to the IEEE 802.1p CoS bits when you apply the explicit default IEEE 802.1p classifier. To do this, include the **default** statement at the **[edit class-of-service interfaces *interface-name* unit *logical-unit-number* classifiers ieee-802.1]** hierarchy level:

NOTE: Only the IEEE 802.1p classifier is supported in Layer 2 interfaces. You must explicitly apply this classifier as shown.

[edit class-of-service interfaces *interface-name* unit *logical-unit-number* classifiers
  ieee-802.1]
default;

Table 8: Default IEEE 802.1p Classifier

Code Point	Forwarding Class	PLP
000	best-effort	low
001	best-effort	high

Table 8: Default IEEE 802.1p Classifier *(continued)*

Code Point	Forwarding Class	PLP
010	expedited-forwarding	low
011	expedited-forwarding	high
100	assured-forwarding	low
101	assured-forwarding	high
110	network-control	low
111	network-control	high

Default IEEE 802.1ad Classifier

Table 9 on page 47 shows the code point, forwarding class alias, and PLP that are assigned to the IEEE 802.1ad bits when you apply the explicit default IEEE 802.1ad classifier. The table is very similar to the IEEE 802.1p default table, but the loss priority is determined by the DEI bit. To apply the default table, include the **default** statement at the **[edit class-of-service interfaces *interface-name* unit *logical-unit-number* classifiers ieee-802.1]** hierarchy level:

```
[edit class-of-service interfaces interface-name unit logical-unit-number classifiers
  ieee-802.1ad]
default;
```

Table 9: Default IEEE 802.1ad Classifier

IEEE 802.1ad Code Point	Forwarding Class Alias	PLP
0000	be	low
0010	be1	low
0100	ef	low
0110	ef1	low
1000	af11	low
1010	af12	low
1100	nc1	low
1110	nc2	low
0001	be-dei	high
0011	be1-dei	high

Table 9: Default IEEE 802.1ad Classifier *(continued)*

IEEE 802.1ad Code Point	Forwarding Class Alias	PLP
0101	ef-dei	high
0111	ef1-dei	high
1001	af11-dei	high
1011	af12-dei	high
1101	nc1-dei	high
1111	nc2-dei	high

Default IP Precedence Classifier (ipprec-default)

There are two separate tables for default IP precedence classification. All logical interfaces are implicitly assigned the **ipprec-compatibility** classifier by default, as described in Table 5 on page 44.

The other default IP precedence classifier (called **ipprec-default**) overrides the **ipprec-compatibility** classifier when you explicitly associate it with a logical interface. To do this, include the **default** statement at the **[edit class-of-service interfaces *interface-name* unit *logical-unit-number* classifiers inet-precedence]** hierarchy level:

```
[edit class-of-service interfaces interface-name unit logical-unit-number classifiers
  inet-precedence]
default;
```

Table 10 on page 48 shows the forwarding class and PLP that are assigned to the IP precedence CoS bits when you apply the default IP precedence classifier.

Table 10: Default IP Precedence (ipprec-default) Classifier

Code Point	Forwarding Class	PLP
000	best-effort	low
001	assured-forwarding	low
010	best-effort	low
011	best-effort	low
100	best-effort	low
101	expedited-forwarding	low
110	network-control	low

Table 10: Default IP Precedence (ipprec-default) Classifier *(continued)*

Code Point	Forwarding Class	PLP
111	**network-control**	**high**

Defining Classifiers

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:

```
[edit class-of-service]
classifiers {
  (dscp | dscp-ipv6 | exp | ieee-802.1 | inet-precedence) classifier-name {
    import [classifier-name | default];
    forwarding-class class-name {
      loss-priority level code-points [ aliases ] [ bit-patterns ];
    }
  }
}
```

The map sets the forwarding class and PLP for a specific set of code-point aliases and bit patterns. The inputs of the map are code-point aliases and bit patterns. The outputs of the map are the forwarding class and the PLP. For more information about how CoS maps work, see “CoS Inputs and Outputs Overview” on page 9.

The classifiers work as follows:

- **dscp**—Handles incoming IPv4 packets.
- **dscp-ipv6**—Handles incoming IPv6 packets. For more information, see “Applying DSCP IPv6 Classifiers” on page 55.
- **exp**—Handles MPLS packets using Layer 2 headers.
- **ieee-802.1**—Handles Layer 2 CoS.
- **inet-precedence**—Handles incoming IPv4 packets. IP precedence mapping requires only the upper three bits of the DSCP field.

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

The code-point aliases and bit patterns are the input for the map. The loss priority and forwarding class are outputs of the map. In other words, the map sets the PLP and forwarding class for a given set of code-point aliases and bit patterns.

NOTE: On M Series, MX Series, and T Series routers that do not have tricolor marking enabled, the loss priority can be configured only by setting the PLP within a multifield classifier. This setting can then be used by the appropriate drop profile map and rewrite rule. For more information, see "Setting Packet Loss Priority" on page 60.

Importing a Classifier

You can use any table, including the default, in the definition of a new classifier by including the **import** statement. The imported classifier is used as a template and is not modified. Whenever you commit a configuration that assigns a new ***class-name*** and **loss-priority** value to a code-point alias or set of bits, it replaces that entry in the imported classifier template. As a result, you must explicitly specify every CoS value in every designation that requires modification.

To do this, include the **import default** statement at the **[edit class-of-service classifiers *type classifier-name*]** hierarchy level:

```
[edit class-of-service classifiers type classifier-name]
import default;
```

For instance, to import the default DSCP classifier, include the **dscp default** statement at the **[edit class-of-service classifiers dscp *classifier-name*]** hierarchy level:

```
[edit class-of-service classifiers dscp classifier-name]
import default;
```

Applying Classifiers to Logical Interfaces

You can apply the classification map to a logical interface by including the **classifiers** 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]
classifiers (dscp | dscp-ipv6 | exp | ieee-802.1 | inet-precedence) (classifier-name | default);
```

You can use interface wildcards for ***interface-name*** and ***logical-unit-number***.

For most PICs, if you apply an IEEE 802.1p classifier to a logical interface, you cannot apply non-IEEE classifiers to other logical interfaces on the same physical interface. This restriction does not apply to Gigabit Ethernet IQ2 PICs.

There are some restrictions on applying multiple BA classifiers to a single logical interface. Table 11 on page 51 shows the supported combinations. In this table, the OSE PICs refer to the 10-port 10-Gigabit OSE PICs.

Table 11: Logical Interface Classifier Combinations

Classifier Combinations	Gigabit Ethernet IQ2 PICs	OSE PICs	Other PICs on M320, MX Series, and T Series	Other M Series with Regular FPCs	Other M Series with Enhanced FPCs
dscp and **inet-precedence**	No	No	No	No	No
dscp-ipv6 and **(dscp | inet-precedence)**	Yes	Yes	Yes	No	No
exp and **ieee 802.1**	Yes	Yes	No	No	No
ieee 802.1 and **(dscp | dscp-ipv6 | exp | inet-precedence)**	Yes	Yes	No	No	Yes
exp and **(dscp | dscp-ipv6 | inet-precedence)**	Yes	Yes	Yes	No	Yes

For Gigabit Ethernet IQ2 and 10-port 10-Gigabit Oversubscribed Ethernet (OSE) interfaces, family-specific classifiers take precedence over IEEE 802.1p BA classifiers. For example, if you configure a logical interface to use both an MPLS EXP and an IEEE 802.1p classifier, the EXP classifier takes precedence. MPLS-labeled packets are evaluated by the EXP classifier, and all other packets are evaluated by the IEEE 802.1p classifier. The same is true about other classifiers when combined with IEEE 802.1p classifiers on the same logical interface.

In Junos OS Releases 9.6 and later, the DSCP and IPv6 DSCP classifiers are not compatible with older formats. You cannot directly replace the old classifier with the new one. You must first delete the old classifier and then apply the new one, although both steps can be done in one configuration session. Otherwise, the commit will fail.

NOTE: If an interface is mounted on an M Series router FPC, you can apply only the default **exp** classifier. If an interface is mounted on an enhanced FPC, you can create a new **exp** classifier and apply it to an interface.

On MX, M120, and M320 routers with Enhanced Type III FPCs only, you can configure user-defined DSCP-based BA classification for MPLS interfaces (this feature is not available for IQE PICs or on MX Series routers when ingress queuing is used) or VPLS/L3VPN routing instances (LSI interfaces). The DSCP-based classification for MPLS packets for Layer 2 VPNs is not supported. To classify MPLS packets on the routing instance at the egress PE, include the **dscp** or **dscp-ipv6** statements at the [**edit class-of-service routing-instances** ***routing-instance-name*** **classifiers**] hierarchy level. To classify MPLS packets at the core-facing interface, apply the classifier at the [**edit class-of-service interface** ***interface-name*** **unit** ***unit-name*** **classifiers (dscp | dscp-ipv6)** ***classifier-name*** **family mpls**] hierarchy level.

NOTE: If you do not apply a DSCP classifier, the default EXP classifier is applied to MPLS traffic.

You can apply DSCP classification for MPLS traffic in four distinct usage scenarios:

- In a Layer 3 VPN (L3VPN) using an LSI routing instance.
  - Supported on the M120, M320, and MX routers.
  - DSCP classifier configured under **[edit class-of-service routing-instances]** on the egress PE router.
- In VPLS using an LSI routing instance.
  - Supported on the M120, M320, and MX routers.
  - DSCP classifier configured under **[edit class-of-service routing-instances]** on the egress PE router.
- In a Layer 3 VPN (L3VPN) using a VT routing instance.
  - Supported on the M120, M320, and MX routers.
  - DSCP classifier configured under **[edit class-of-service interfaces]** on the core-facing interface on the egress PE router.
- MPLS forwarding.
  - Supported on the M120, M320, and MX routers (not supported on IQE and MX when ingress queueing is enabled).
  - DSCP classifier configured under **[edit class-of-service interfaces]** on the ingress core-facing interface on the P or egress PE router.

The following configuration scenarios are not supported:

- VPLS using the VT routing instance.
- MPLS forwarding when the label stacking is greater than 2.

The following example configures a DSCP classifier for IPv4 named **dscp-ipv4-classifier** for the **fc-af11-class** forwarding class and a corresponding IPv6 DSCP classifier:

```
class-of-service {
  routing-instances routing-instance-one {
    classifiers {
      dscp dscp-ipv4-classifier {
        loss-priority low code-points 000100;
      }
      dscp dscp-ipv6-classifier {
        forwarding-class fc-af11-class {
          loss-priority low {
            code-points af11;
          }
```



```
        }
      }
    }
  }
}
```

NOTE: This is not a complete configuration.

This example applies the IPv4 classifier to MPLS traffic and the IPv6 classifier to Internet traffic on interface **ge-2/0/3.0**:

```
class-of-service {
  interfaces ge-2/0/3 {
    unit 0 {
      classifiers {
        dscp dscp-ipv4-classifier {
          family mpls;
        }
        dscp-ipv6 dscp-ipv6-classifier {
          family inet; # This is the default if not present.
        }
      }
    }
  }
}
```

NOTE: This is not a complete configuration.

This example applies the same classifier to both MPLS and IP traffic on interface **ge-2/2/0**.

```
[edit class-of-services interface ge-2/2/0]
unit 0 {
  classifiers {
    dscp dscp-mpls {
      family [ mpls inet ];
    }
  }
}
```

NOTE: This is not a complete configuration.

NOTE: You can apply DSCP and DSCP IPv6 classifiers to explicit null MPLS packets. The **family mpls** statement works the same on both explicit null and non-null MPLS labels.

Configuring BA Classifiers for Bridged Ethernet

On M120 and M320 routers equipped with IQ2 PICs, you can configure BA classification based on the IEEE 802.1 bits for bridged Ethernet over Asynchronous Transfer Mode (ATM), Point-to-Point Protocol (PPP), and frame relay for VPLS applications. The BA classification is applied to the first (outer) tag when tagged frames are received. Untagged frames are bypassed and a value of 000 for the classification IEEE 802.1p bits is assumed. There is no support for circuit cross-connect (CCC), and only port-mode VPLS is supported (in port-mode VPLS, only VLANs on a single physical port are included in the VPLS instance). There is no support for multilink PPP bonding with VPLS. For bridging over frame relay, only frames that do not preserve the frame check sequence (FCS) field are supported. Frames that preserve the FCS field are silently discarded.

The bridging over PPP function is restricted:

- There is no support for “tinygram” compression and expansion.
- Frames received with preserved FCS bits are silently discarded.
- Bridge control frames are also classified based on header bit values.
- Both tagged and untagged frames are classified and forwarded. The peer must discard frame types that are not supported.

This example applies an IEEE 802.1p classifier named **ppp-ether-vpls-classifier** to an interface (**so-1/2/3**) with Ethernet VPLS over PPP encapsulation. Note that the interface and CoS configuration must be consistent to support the feature. You must also configure the classifier and other CoS parameters such as forwarding classes.

```
[edit class-of-service]
interfaces {
  so-1/2/3 {
    unit 0 {
      classifiers {
        ieee-802.1 ppp-ether-vpls-classifier;
      }
    }
  }
}

[edit interfaces]
s0-1/2/3 {
  encapsulation ether-vpls-over-ppp;
  unit 0 {
    family vpls;
  }
}
```

On routers with IQ2 or IQ2E PICs, you can perform BA classification based on the value of the inner VLAN tag in an Ethernet frame. To configure BA classification based on the inner VLAN tag value, include the **inner** option at the [**edit class-of-service interfaces** ***interface-name*** **unit** ***logical-unit-number*** **classifiers ieee-802.1** ***classifier-name*** **vlan-tag**] hierarchy level. You must configure the inner VLAN tag for the logical interface with the

inner option at the [**edit interfaces *interface-name* unit *logical-interface-name* vlan-tag**] hierarchy level.

```
[edit class-of-service interfaces ge-2/2/2 unit 0]
classifiers ieee-802.1 inner-vlan-tag-ba-classifier {
  vlan-tag inner;
}
```

Tunneling and BA Classifiers

BA classifiers can be used with GRE and IP-IP tunnels on the following routers:

- M7i and M10i routers
- M Series routers with E-FPC or EP-FPC
- M120 routers
- M320 routers
- T Series routers

When a GRE or IP-IP tunnel is configured on an incoming (core-facing) interface, the queue number and PLP information are carried through the tunnel. At the egress (customer-facing) interface, the packet is queued and the CoS bits rewritten based on the information carried through the tunnel.

If no BA classifier is configured in the incoming interface, the default classifier is applied. If no rewrite rule is configured, the default rewrite rule is applied.

Applying DSCP IPv6 Classifiers

For M320 and T Series routers, you can apply separate classifiers for IPv4 and IPv6 packets per logical interface by including the **classifiers** statement at the **[edit class-of-service interfaces *interface-name* unit *logical-unit-number*]** hierarchy level and specifying the **dscp** and **dscp-ipv6** classifier types:

```
[edit class-of-service interfaces interface-name unit logical-unit-number]
classifiers dscp (classifier-name | default) family (mpls | inet);
classifiers dscp-ipv6 (classifier-name | default) family (mpls | inet));
```

For M Series router enhanced FPCs, you cannot apply separate classifiers for IPv4 and IPv6 packets on a single logical interface. Instead, classifier assignment works as follows:

- If you assign a DSCP classifier only, IPv4 and IPv6 packets are classified using the DSCP classifier.
- If you assign an IP precedence classifier only, IPv4 and IPv6 packets are classified using the IP precedence classifier. In this case, the lower three bits of the DSCP field are ignored because IP precedence mapping requires the upper three bits only.
- If you assign either the DSCP or the IP precedence classifier in conjunction with the DSCP IPv6 classifier, the commit fails.
- If you assign a DSCP IPv6 classifier only, IPv4 and IPv6 packets are classified using the DSCP IPv6 classifier, but the commit displays a warning message.

For more information, see “Applying Classifiers to Logical Interfaces” on page 50. For a complex configuration example, see the *Junos OS Feature Guide*.

Applying 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, the default MPLS EXP classifier is applied to the routing instance. For detailed information about VRF table labels, see the *Junos OS VPNs Configuration Guide*.

The default MPLS EXP classification table contents are shown in Table 12 on page 56.

Table 12: Default MPLS EXP Classification Table

Forwarding Class	Loss Priority	CoS Value
best-effort	low	000
best-effort	high	001
expedited-forwarding	low	010
expedited-forwarding	high	011
assured-forwarding	low	100
assured-forwarding	high	101
network-control	low	110
network-control	high	111

For PICs that are installed on enhanced FPCs, you can override the default MPLS EXP classifier and apply a custom classifier to the routing instance. To do this, perform the following configuration tasks:

1. Filter traffic based on the IP header by including the **vrf-table-label** statement at the **[edit routing-instances *routing-instance-name*]** hierarchy level:

```
[edit routing-instances routing-instance-name]
vrf-table-label;
```

2. Configure a custom MPLS EXP classifier by including the following statements at the **[edit class-of-service]** hierarchy level:

```
[edit class-of-service]
classifiers {
  exp classifier-name {
    import (classifier-name | default);
    forwarding-class class-name {
      loss-priority level code-points [ aliases ] [ bit-patterns ];
    }
  }
```



```
}
forwarding-classes {
  queue queue-number class-name priority (high | low);
}
```

3. Configure the routing instance to use the custom MPLS EXP classifier by including the **exp** statement at the **[edit class-of-service routing-instances** *routing-instance-name* **classifiers]** hierarchy level:

```
[edit class-of-service routing-instances routing-instance-name classifiers]
exp classifier-name;
```

To display the MPLS EXP classifiers associated with all routing instances, issue the **show class-of-service routing-instances** command.

NOTE: The following caveats apply to custom MPLS EXP classifiers for routing instances:

- An enhanced FPC is required.
- Logical systems are not supported.

For more details, see the following sections:

- Configuring Global Classifiers and Wildcard Routing Instances on page 57
- Examples: Applying MPLS EXP Classifiers to Routing Instances on page 58

Configuring Global Classifiers and Wildcard Routing Instances

To configure a global routing instance classifier, include the **all** statement at the **[edit class-of-service routing-instances]** hierarchy level:

```
[edit class-of-service routing-instances]
all {
  classifiers {
    exp classifier-name;
  }
}
```

For routing instances associated with specific classifiers, the global configuration is ignored.

To use a wildcard in the routing instance classifier configuration, include an asterisk (*) in the name of the routing instance:

```
[edit class-of-service routing-instances]
routing-instance-name* {
  classifiers {
    exp classifier-name;
  }
}
```

The wildcard configuration follows the longest match. If there is a specific configuration, it is given precedence over the wildcard configuration.

NOTE: Wildcards and the **all** keyword are supported 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***, the Junos OS 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, the Junos OS treats the **vpn*** routing instance name as a wildcard, and all the 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.

Examples: Applying MPLS EXP Classifiers to Routing Instances

Configure a global classifier for all routing instances and override the global classifier for a specific routing instance. In this example, there are 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**.

Configuring a Global Classifier

```
[edit routing-instances]
vpn1 {
  vrf-table-label;
}
vpn2 {
  vrf-table-label;
}
vpn3 {
  vrf-table-label;
}

[edit class-of-service routing-instances]
all {
  classifiers {
    exp exp-classifier-global;
  }
}
vpn3 {
  classifiers {
    exp exp-classifier-3;
  }
}
```

Configure a wildcard routing instance and override the wildcard with a specific routing instance. In this example, there are 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**.

Configuring a Wildcard Routing Instance

```
[edit routing-instances]
vpn-red {
  vrf-table-label;
```



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

[edit class-of-service routing-instances]
vpn* {
  classifiers {
    exp exp-class-wildcard;
  }
}
vpn-red {
  classifiers {
    exp exp-class-red;
  }
}
```

Display the MPLS EXP classifiers associated with two routing instances:

Monitoring a Configuration

```
[edit class-of-service routing-instances]
vpn1 {
  classifiers {
    exp default;
  }
}
vpn2 {
  classifiers {
    exp class2;
  }
}
```

```
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
```

Applying MPLS EXP Classifiers for Explicit-Null Labels

When you configure MPLS explicit-null labels, label 0 is advertised to the egress router of an LSP. When label 0 is advertised, the egress router (instead of the penultimate router) removes the label. Ultimate-hop popping ensures that any packets traversing an MPLS network include a label. For more information about explicit-null labels and ultimate-hop popping, see the *Junos OS MPLS Applications Configuration Guide*.

On M320 and T Series routers, when you configure MPLS explicit-null labels with an MPLS EXP classifier, the MPLS EXP classifier can be different from an IPv4 or IPv6 classifier configured on the same logical interface. In other words, you can apply separate classifiers for MPLS EXP, IPv4, and IPv6 packets per logical interface. To combine an

EXP classifier with a distinct IPv6 classifier, the PIC must be mounted on an Enhanced FPC.

NOTE: For J Series routers and other M Series routers, MPLS explicit-null labels with MPLS EXP classification are supported if you set the same classifier for EXP and IPv4 traffic, or EXP and IPv6 traffic.

For more information about how IPv4 and IPv6 packet classification is handled, see "Applying DSCP IPv6 Classifiers" on page 55.

To configure an MPLS EXP classifiers for explicit-null labels, include the **exp** statement at the **[edit class-of-service classifiers]** and **[edit class-of-service interfaces *interface-name* unit *logical-unit-number* classifiers]** hierarchy levels:

```
[edit class-of-service classifiers]
exp classifier-name {
  import (classifier-name | default);
  forwarding-class class-name {
    loss-priority level code-points [ aliases ] [ bit-patterns ];
  }
}
[edit class-of-service interfaces interface-name unit logical-unit-number classifiers]
exp (classifier-name | default);
```

Setting Packet Loss Priority

By default, the least significant bit of the CoS value sets the packet loss priority (PLP) value. For example, CoS value 000 is associated with PLP **low**, and CoS value 001 is associated with PLP **high**. In general, you can change the PLP by configuring a behavior aggregate (BA) or multifield classifier, as discussed in "Overview of BA Classifier Types" on page 42 and "Multifield Classifier Overview" on page 71.

However, on Juniper Networks M320 Multiservice Edge Routers, MX Series Ethernet Services Routers, and T Series Core Routers that do not have tricolor marking enabled, the loss priority can be configured only by setting the PLP within a multifield classifier. This setting can then be used by the appropriate drop profile map and rewrite rule.

On M320 routers and T Series routers with Enhanced II Flexible PIC Concentrators (FPCs) and tricolor marking enabled, you can set the PLP with a BA or multifield classifier, as described in "Using BA Classifiers to Set PLP" on page 102 and "Using Multifield Classifiers to Set PLP" on page 103.

On T Series routers with different Packet Forwarding Engines (non-Enhanced Scaling and Enhanced Scaling FPCs), you can configure PLP bit copying for ingress and egress unicast and multicast traffic. To configure, include the **copy-plp-all** statement at the [**edit class-of-service**] hierarchy level.

Example: Overriding the Default PLP on M320 Routers

The following example shows a two-step procedure to override the default PLP settings on M320 routers:

1. The following example specifies that while the DSCP code points are 110, the loss priority is set to **high**; however, on M320 routers, overriding the default PLP this way has no effect.

```
class-of-service {
  classifiers {
    dscp ba-classifier {
      forwarding-class expedited-forwarding {
        loss-priority high code-points 110;
      }
    }
  }
}
```

2. For M320 routers, this multifield classifier sets the PLP.

```
firewall {
  filter ef-filter {
    term ef-multifield {
      from {
        precedence 6;
      }
      then {
        loss-priority high;
        forwarding-class expedited-forwarding;
      }
    }
  }
}
```

Configuring and Applying IEEE 802.1ad Classifiers

For Juniper Network MX Series Ethernet Services Router interfaces or IQ2 PICs with IEEE 802.1ad frame formats, you can set the forwarding class and loss priority for traffic on the basis of the three IEEE 802.1p bits and the DEI bit. You can apply the default map or customize one or more of the default values.

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

Defining Custom IEEE 802.1ad Maps

You can customize the default IEEE 802.1ad map by defining values for IEEE 802.1ad code points.

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


Applying Custom IEEE 802.1ad Maps

You then apply the classifier map to the logical interface:

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

Verifying Custom IEEE 802.1ad Map Configuration

To verify your configuration, you can issue the following operational mode commands:

- **show class-of-service forwarding-table loss-priority-map**
- **show class-of-service forwarding-table loss-priority-map mapping**
- **show chassis forwarding**
- **show pfe fwdd**

Understanding DSCP Classification for VPLS

You can perform Differentiated Services Code Point (DSCP) classification for IPv4 packets on Ethernet interfaces that are part of a virtual private LAN service (VPLS) routing instance on the ingress provider edge (PE) router. This is supported on the M320 router with Enhanced type III FPC and the M120 router. On the ATM II IQ PIC, the **ether-vpls-over-atm-llc** encapsulation statement is required. On the Intelligent Queuing 2 (IQ2) or Intelligent Queuing 2 Enhanced (IQ2E) PICs, the **vlan-vpls** encapsulation statement is required. DSCP for IPv6 and Internet precedence for IPv6 are not supported.

In order to perform DSCP classification for IPv4 packets on Ethernet interfaces that are part of a VPLS routing instance on the ingress PE router, you must make sure of the following:

- The correct encapsulation statement based on PIC type is configured for the interface.
- The DSCP classifier is defined (default is allowed) at the **[edit class-of-service classifiers]** hierarchy level.
- The defined DSCP classifier is applied to the interface.
- The interface is included in the VPLS routing instance on the ingress of the PE router.

Related Documentation

- BA Classifier Overview on page 39

Example: Configuring DSCP Classification for VPLS

The following example configures DSCP classifier **dscp_vpls** on ATM interface **at-4/1/1** with **ether-vpls-over-atm-llc** encapsulation. The classifier **dscp_vpls** is applied to the interface and the interface is listed in the VPLS routing instance **vpls1** on the ingress PE router.

1. Configure the ATM interface **at-4/1/1.0** and the encapsulation as **ether-vpls-over-atm-llc**:

```
[edit]
interfaces {
   at-4/1/1 {
      mtu 9192;
      atm-options {
         vpi 10;
      }
      unit 0 {
         encapsulation ether-vpls-over-atm-llc;
         vci 10.128;
         family vpls;
      }
   }
}
```

2. Configure the DSCP classifer **dscp_vpls**:

```
[edit]
class-of-service {
   classifiers {
      dscp dscp_vpls {
         forwarding-class expedited-forwarding {
            loss-priority low code-points 000010;
         }
      }
   }
}
```

3. Apply the classifier **dscp_vpls** to the ATM interface **at-4/1/1.0**:

```
[edit]
interfaces {
   at-4/1/1 {
      unit 0 {
         classifiers {
            dscp dscp_vpls;
         }
      }
   }
}
```

4. Include the ATM interface virtual circuit **at-4/1/1.0** as part of the routing instance **vpls1** configuration:

```
[edit]
   routing-instances {
```

```
        vpls1 {
            instance-type vpls;
            interface at-4/1/1.0;
            route-distinguisher 10.255.245.51:1;
            vrf-target target:1234:1;
            protocols {
                vpls {
                    site-range 10;
                    no-tunnel-services;
                    site vpls-1-site-1 {
                        site-identifier 1;
                    }
                }
            }
        }
    }
```

Related Documentation

- Understanding DSCP Classification for VPLS on page 62

BA Classifiers and ToS Translation Tables

On some PICs, the behavior aggregate (BA) translation tables are included for every logical interface (unit) protocol family configured on the logical interface. The proper default translation table is active even if you do not include any explicit translation tables. You can display the current translation table values with the **show class-of-service classifiers** command.

On Juniper Networks M40e, M120, M320 Multiservice Edge Routers, and T Series Core Routers with Enhanced IQ (IQE) PICs, or on any router with IQ2 or Enhanced IQ2 (IQ2E) PICs, you can replace the type-of-service (ToS) bit value on the incoming packet header on a logical interface with a user-defined value. The new ToS value is used for all class-of-service processing and is applied before any other class-of-service or firewall treatment of the packet. The PIC uses the **translation-table** statement to determine the new ToS bit values.

You can configure a physical interface (port) or logical interface (unit) with up to three translation tables. For example, you can configure a port or unit with BA classification for IPv4 DSCP, IPv6 DSCP, and MPLS EXP. The number of frame relay data-link connection identifiers (DLCIs) (units) that you can configure on each PIC varies based on the number and type of BA classification tables configured on the interfaces.

For more information on configuring ToS translation tables, along with examples, see “Configuring ToS Translation Tables” on page 291.

CHAPTER 4

Defining Code-Point Aliases

This topic discusses the following:

Default Code-Point Alias Overview

Behavior aggregate (BA) classifiers use class-of-service (CoS) values such as Differentiated Services code points (DSCPs), DSCP IPv6, IP precedence, IEEE 802.1 and MPLS experimental (EXP) bits to associate incoming packets with a particular CoS servicing level. On a Services Router, you can assign a meaningful name or alias to the CoS values and use this alias instead of bits when configuring CoS components. These aliases are not part of the specifications but are well known through usage. For example, the alias for DSCP 101110 is widely accepted as **ef** (expedited forwarding).

NOTE: The code point aliases must begin with a letter and can be up to 64 characters long.

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.

To configure class-of-service (CoS) code point aliases, include the **code-point-aliases** statement at the **[edit class-of-service]** hierarchy level:

```
[edit class-of-service]
code-point-aliases {
  (dscp | dscp-ipv6 | exp | ieee-802.1 | inet-precedence) {
    alias-name bits;
  }
}
```

Related Documentation

- **code-point-aliases on page 457**

Default CoS Values

Table 13 on page 66 shows the default mappings between the bit values and standard aliases. For example, it is widely accepted that the alias for DSCP 101110 is **ef** (expedited forwarding).

Table 13: Default CoS Values

CoS Value Types	Mapping
DSCP and DSCP IPv6 CoS Values	
ef	101110
af11	001010
af12	001100
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
cs5	101000

Table 13: Default CoS Values *(continued)*

CoS Value Types	Mapping
nc1/cs6	110000
nc2/cs7	111000
MPLS EXP CoS Values	
be	000
be1	001
ef	010
ef1	011
af11	100
af12	101
nc1/cs6	110
nc2/cs7	111
IEEE 802.1 CoS Values	
be	000
be1	001
ef	010
ef1	011
af11	100
af12	101
nc1/cs6	110
nc2/cs7	111
Legacy IP Precedence CoS Values	
be	000
be1	001
ef	010
ef1	011

Table 13: Default CoS Values *(continued)*

CoS Value Types	Mapping
af11	100
af12	101
nc1/cs6	110
nc2/cs7	111

Defining Code Point Aliases for Bit Patterns

To define a code-point alias, include the **code-point-aliases** statement at the **[edit class-of-service]** hierarchy level:

```
[edit class-of-service]
code-point-aliases {
  (dscp | dscp-ipv6 | exp | ieee-802.1 | inet-precedence) {
    alias-name bit-pattern;
  }
}
```

The CoS marker types are as follows:

- **dscp**—Handles incoming IPv4 packets.
- **dscp-ipv6**—Handles incoming IPv6 packets. For more information, see “Applying DSCP IPv6 Classifiers” on page 55.
- **exp**—Handles MPLS packets using Layer 2 headers.
- **ieee-802.1**—Handles Layer 2 CoS.
- **inet-precedence**—Handles incoming IPv4 packets. IP precedence mapping requires only the upper three bits of the DSCP field.

For example, you might configure the following aliases:

```
[edit class-of-service]
code-point-aliases {
  dscp {
    my1 110001;
    my2 101110;
    be 000001;
    cs7 110000;
  }
}
```

This configuration produces the following mapping:

```
user@host> show class-of-service code-point-aliases dscp
Code point type: dscp

Alias              Bit pattern
```

ef/my2	101110
af11	001010
af12	001100
af13	001110
af21	010010
af22	010100
af23	010110
af31	011010
af32	011100
af33	011110
af41	100010
af42	100100
af43	100110
be	000001
cs1	001000
cs2	010000
cs3	011000
cs4	100000
cs5	101000
nc1/cs6/cs7	110000
nc2	111000
my1	110001

The following notes explain certain results in the mapping:

- **my1 110001**:
  - 110001 was not mapped to anything before, and **my1** is a new alias.
  - Nothing in the default mapping table is changed by this statement.
- **my2 101110**:
  - 101110 is now mapped to **my2** as well as **ef**.
- **be 000001**:
  - **be** is now mapped to 000001.
  - The old value of **be**, 000000, is not associated with any alias. Packets with this DSCP value are now mapped to the default forwarding class.
- **cs7 110000**:
  - **cs7** is now mapped to 110000, as well as **nc1** and **cs6**.
  - The old value of **cs7**, 111000, is still mapped to **nc2**.

CHAPTER 5

Classifying Packets Based on Various Packet Header Fields

This topic discusses the following:

Multifield Classifier Overview

A multifield classifier is a method of classifying traffic flows. Devices that sit at the edge of a network usually classify packets according to codings that are located in multiple packet header fields. Multifield classification is normally performed at the network edge because of the general lack of DiffServ code point (DSCP) or IP precedence support in end-user applications.

In an edge router, a multifield classifier provides the filtering functionality that scans through a variety of packet fields to determine the forwarding class for a packet. Typically, a classifier performs matching operations on the selected fields against a configured value.

Unlike a behavior aggregate (BA), which classifies packets based on class-of-service (CoS) bits in the packet header, a multifield classifier can examine multiple fields in the packet header—for example, the source and destination address of the packet, and the source and destination port numbers of the packet. A multifield classifier typically matches one or more of the six packet header fields: destination address, source address, IP protocol, source port, destination port, and DSCP. multifield classifiers are used when a simple BA classifier is insufficient to classify a packet.

In the Junos OS, you configure an multifield classifier with a firewall filter and its associated match conditions. This enables you to use any filter match criteria to locate packets that require classification. From a CoS perspective, multifield classifiers (or firewall filter rules) provide the following services:

- Classify packets to a forwarding class and loss priority. The forwarding class determines the output queue. The loss priority is used by schedulers in conjunction with the random early discard (RED) algorithm to control packet discard during periods of congestion.
- Police traffic to a specific bandwidth and burst size. Packets exceeding the policer limits can be discarded, or can be assigned to a different forwarding class, to a different loss priority, or to both.

NOTE: You *police* traffic on input to conform to established CoS parameters, setting loss handling and forwarding class assignments as needed. You *shape* traffic on output to make sure router resources, especially bandwidth, are distributed fairly. However, input policing and output shaping are two different CoS processes, each with their own configuration statements.

Configuring Multifield Classifiers

If you configure both a behavior aggregate (BA) classifier and a multifield classifier, BA classification is performed first; then multifield classification is performed. If they conflict, any BA classification result is overridden by the multifield classifier.

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

To activate a multifield classifier, you must configure it on a logical interface. There is no restriction on the number of multifield classifiers you can configure.

NOTE: For MX Series routers, if you configure a firewall filter with a DSCP action or traffic-class action on a DPC, the commit does not fail, but a warning displays and an entry is made in the syslog.

To configure multifield classifiers, include the following statements at the **[edit firewall]** hierarchy level:

```
[edit firewall]
```



```
family family-name {
    filter filter-name {
        term term-name {
            from {
                match-conditions;
            }
            then {
                dscp 0;
                forwarding-class class-name;
                loss-priority (high | low);
            }
        }
    }
    simple-filter filter-name {
        term term-name {
            from {
                match-conditions;
            }
            then {
                forwarding-class class-name;
                loss-priority (high | low | medium);
            }
        }
    }
}
```

The **[edit firewall]** configuration statements are discussed in detail in the *Junos OS Policy Framework Configuration Guide*.

Example: Classifying Packets Based on Their Destination Address

Configure a multifield classifier that ensures that all IPv4 packets destined for the **10.10.10.0/24** network are placed into the **platinum** forwarding class. This assignment occurs regardless of the received CoS bit values in the packet. Apply this filter to the inbound interface **so-1/2/2.0**.

To verify that your configuration is attached to the correct interface, issue the **show interfaces filters** command.

```
firewall {
    family inet {
        filter set-FC-to-platinum {
            term match-a-single-route {
                from {
                    destination-address {
                        10.10.10.0/24;
                    }
                }
                then {
                    forwarding-class platinum;
                    accept;
                }
            }
            term accept-all {
                then accept;
```



```
          }
        }
      }
    }
    interfaces {
      so-1/2/2 {
        unit 0 {
          family inet {
            filter {
              input set-FC-to-platinum;
            }
          }
        }
      }
    }
```

Example: Configuring and Verifying a Complex Multifield Filter

In this example, SIP signaling (VoIP) messages use TCP/UDP, port 5060, and RTP media channels use UDP with port assignments from 16,384 through 32,767. See the following sections:

- Configuring a Complex Multifield Filter on page 74
- Verifying a Complex Multifield Filter on page 76

Configuring a Complex Multifield Filter

To configure the multifield filter, perform the following actions:

- Classify SIP signaling messages (VoIP network control traffic) as NC with a firewall filter.
- Classify VoIP traffic as EF with the same firewall filter.
- Police all remaining traffic with IP precedence **0** and make it BE.
- Police BE traffic to 1 Mbps with excess data marked with PLP high.
- Apply the firewall filter with policer to the interface.

The firewall filter called **classify** matches on the transport protocol and ports identified in the incoming packets and classifies packets into the forwarding classes specified by your criteria.

The first term, **sip**, classifies SIP signaling messages as network control messages. The **port** statement matches any source port or destination port (or both) that is coded to 5060.

Classifying SIP Signaling Messages

```
firewall {
  family inet {
    filter classify {
      interface-specific;
      term sip {
        from {
```



```
                protocol [ udp tcp ];
                port 5060;
            }
            then {
                forwarding-class network-control;
                accept;
            }
        }
    }
  }
}
```

The second term, **rtp**, classifies VoIP media channels that use UDP-based transport.

Classifying VoIP Channels That Use UDP

```
term rtp {
    from {
        protocol udp;
        port 16384-32767;
    }
    then {
        forwarding-class expedited-forwarding;
        accept;
    }
}
```

The policer's burst tolerance is set to the recommended value for a low-speed interface, which is ten times the interface MTU. For a high-speed interface, the recommended burst size is the transmit rate of the interface times 3 to 5 milliseconds.

Configuring the Policer

```
policer be-policer {
    if-exceeding {
        bandwidth-limit 1m;
        burst-size-limit 15k;
    }
    then loss-priority high;
}
```

The third term, **be**, ensures that all remaining traffic is policed according to a bandwidth restriction.

Policing All Remaining Traffic

```
term be {
    then policer be-policer;
}
```

The **be** term does not include a **forwarding-class** action modifier. Furthermore, there is no explicit treatment of network control (NC) traffic provided in the **classify** filter. You can configure explicit classification of NC traffic and all remaining IP traffic, but you do not need to, because the default IP precedence classifier correctly classifies the remaining traffic.

Apply the **classify** classifier to the **fe-0/0/2** interface:

Applying the Classifier

```
interfaces {
  fe-0/0/2 {
    unit 0 {
      family inet {
        filter {
          input classify;
        }
        address 10.12.0.13/30;
      }
    }
  }
}
```

Verifying a Complex Multifield Filter

Before the configuration is committed, display the default classifiers in effect on the interface using the **show class-of-service interface *interface-name*** command. The display confirms that the **ipprec-compatibility** classifier is in effect by default.

Verifying Default Classification

```
user@host> show class-of-service fe-0/0/2
Physical interface: fe-0/0/2, Index: 135
Queues supported: 8, Queues in use: 4
  Scheduler map: <default>, Index: 2032638653

  Logical interface: fe-0/0/2.0, Index: 68
    Shaping rate: 32000
    Object         Name                  Type                       Index
    Scheduler-map  <default>                                           27
    Rewrite        exp-default           exp                           21
    Classifier     exp-default           exp                            5
    Classifier     ipprec-compatibility  ip                             8
```

To view the default classifier mappings, use the **show class-of-service classifier name *name*** command. The highlighted output confirms that traffic with IP precedence setting of 0 is correctly classified as BE, and NC traffic, with precedence values of 6 or 7, is properly classified as NC.

Displaying Default Classifier Mappings

```
user@host> show class-of-service classifier name ipprec-compatibility
Classifier: ipprec-compatibility, Code point type: inet-precedence, Index: 12
  Code point           Forwarding class                     Loss priority
  000                  best-effort                          low
  001                  best-effort                          high
  010                  best-effort                          low
  011                  best-effort                          high
  100                  best-effort                          low
  101                  best-effort                          high
  110                  network-control                      low
  111                  network-control                      high
```

After your configuration is committed, verify that your multifield classifier is working correctly. You can monitor the queue counters for the router's ***egress*** interface used when forwarding traffic received from the peer. Displaying the queue counters for the ingress interface (**fe-0/0/2**) does not allow you to check your ingress classification, because queuing generally occurs only at egress in the Junos OS. (Ingress queuing is supported on Gigabit Ethernet IQ2 PICs and Enhanced IQ2 PICs only.)

To verify the operation of the multifield filter:

1. To determine which egress interface is used for the traffic, use the **traceroute** command.
2. After you identify the egress interface, clear its associated queue counters by issuing the **clear interfaces statistics *interface-name*** command.
3. Confirm the default forwarding class-to-queue number assignment. This allows you to predict which queues are used by the VoIP, NC, and other traffic. To do this, issue the **show class-of-service forwarding-class** command.
4. Display the queue counts on the interface by issuing the **show interfaces queue** command.

Example: Writing Different DSCP and EXP Values in MPLS-Tagged IP Packets

On Juniper Networks M320 Multiservice Edge Routers and T Series Core Routers, you can selectively set the DSCP field of MPLS-tagged IPv4 and IPv6 packets to **000000**. In the same packets, you can set the MPLS EXP field according to a configured rewrite table, which is based on the forwarding classes that you set in incoming packets using a BA or multifield classifier.

Queue selection is based on the forwarding classes you assign in scheduler maps. This means that you can direct traffic to a single output queue, regardless of whether the DSCP field is unchanged or rewritten to **000000**. To do this, you must configure a multifield classifier that matches selected packets and modifies them with the **dscp 0** action.

Selective marking of DSCP fields to **0**, without affecting output queue assignment, can be useful. For example, suppose you need to use the MPLS EXP value to configure CoS applications for core provider routers. At the penultimate egress provider edge (PE) router where the MPLS labels are removed, the CoS bits need to be provided by another value, such as DSCP code points. This case illustrates why it is useful to mark both the DSCP and MPLS EXP fields in the packet. Furthermore, it is useful to be able to mark the two fields differently, because the CoS rules of the core provider router might differ from the CoS rules of the egress penultimate router. At egress, as always, you can use a rewrite table to rewrite the MPLS EXP values corresponding to the forwarding classes that you need to set.

For IPv4 traffic, the **dscp 0** action modifier at the **[edit firewall family inet filter *filter-name* term *term-name* then]** hierarchy level is valid. However, for IPv6 traffic, you configure this feature by including the **traffic-class 0** action modifier at the **[edit firewall family inet6 filter *filter-name* term *term-name* then]** hierarchy level.

In the following IPv4 example, term **1** of the multifield classifier matches packets with DSCP **001100** code points coming from a certain VRF, rewrites the bits to DSCP **000000**, and sets the forwarding class to **best-effort**. In term **2**, the classifier matches packets with DSCP **010110** code points and sets the forwarding class to **best-effort**. Because term **2** does not include the **dscp 0** action modifier, the DSCP **010110** bits remain unchanged. Because the classifier sets the forwarding class for both code points to **best-effort**, both traffic types are directed to the same output queue.

NOTE: If you configure a bit string in a DSCP match condition in a firewall filter, then you must include the letter “b” in front of the string, or the match rule creation fails on commit.

```
firewall {
  family inet {
    filter vrf-rewrite {
      term 1 {
        from {
          dscp b001100;
        }
        then {
          dscp 0;
          forwarding-class best-effort;
        }
      }
      term 2 {
        from {
          dscp b010110;
        }
        then {
          forwarding-class best-effort;
        }
      }
    }
  }
}
```

Applying the Multifield Classifier

Apply the filter to an input interface corresponding to the VRF:

```
interfaces {
  so-0/1/0 {
    unit 0 {
      family inet {
        filter input vrf-rewrite;
      }
    }
  }
}
```


NOTE: The **dscp 0** action is supported in both input and output filters. You can use this action for non-MPLS packets as well as for IPv4 and IPv6 packets entering an MPLS network. All IPv4 and IPv6 firewall filter match conditions are supported with the **dscp 0** action.

The following limitations apply:

- You can use a multifield classifier to rewrite DSCP fields to value 0 only. Other values are not supported.
- If a packet matches a filter that has the **dscp 0** action, then the outgoing DSCP value of the packet is **0**, even if the packet matches a rewrite rule, and the rewrite rule is configured to mark the packet to a non-zero value. The **dscp 0** action overrides any other rewrite rule actions configured on the router.
- Although you can use the **dscp 0** action on an input filter, the output filter and other classifiers do not see the packet as being marked **dscp 0**. Instead, they classify the packet based on its original incoming DSCP value. The DSCP value of the packet is set to **0** after all other classification actions have completed on the packet.

Overview of Simple Filters

Simple filters are recommended for metropolitan Ethernet applications. They are supported on Gigabit Ethernet intelligent queuing 2 (IQ2) and Enhanced Queuing Dense Port Concentrator (DPC) interfaces only.

Unlike normal filters, simple filters are for IPv4 traffic only and have the following restrictions:

- The **next term** action is not supported.
- Qualifiers, such as the **except** and **protocol-except** statements, are not supported.
- Noncontiguous masks are not supported.
- Multiple source addresses and destination addresses in a single term are not supported. If you configure multiple addresses, only the last one is used.
- Ranges are only valid as source or destination ports. For example, **source-port 400-500** or **destination-port 600-700**.
- Output filters are not supported. You can apply a simple filter to ingress traffic only.
- Simple filters are not supported for interfaces in an aggregated-Ethernet bundle.
- Explicitly configurable terminating actions, such as **accept**, **reject**, and **discard**, are not supported. Simple filters always accept packets.

NOTE: On the MX Series routers with the Enhanced Queuing DPC, the forwarding class is not supported as a from match condition.

Example: Configuring a Simple Filter

This simple filter sets the loss priority to low for TCP traffic with source address **1.1.1.1**, sets the loss priority to high for HTTP (web) traffic with source addresses in the **4.0.0.0/8** range, and sets the loss priority to low for all traffic with destination address **6.6.6.6**. The simple filter is applied as an input filter (arriving packets are checking for destination address **6.6.6.6**, not queued output packets) on interface **ge-0/0/1.0**.

```
firewall {
  family inet {
    simple-filter filter1 {
      term 1 {
        from {
          source-address {
            1.1.1.1/32;
          }
          protocol {
            tcp;
          }
        }
        then loss-priority low;
      }
      term 2 {
        from {
          source-address {
            4.0.0.0/8;
          }
          source-port {
            http;
          }
        }
        then loss-priority high;
      }
      term 3 {
        from {
          destination-address {
            6.6.6.6/32;
          }
        }
        then {
          loss-priority low;
          forwarding-class best-effort;
        }
      }
    }
  }
}
interfaces {
```

```
    ge-0/0/1 {
      unit 0 {
        family inet {
          simple-filter {
            input filter1;
          }
          address 10.1.2.3/30;
        }
      }
    }
}
```

Configuring Logical Bandwidth Policers

When you configure a policer as a percentage (using the **bandwidth-percent** statement), the bandwidth is calculated as a percentage of either the physical interface media rate or the logical interface shaping rate. To specify that the bandwidth be calculated based on the logical interface shaping rate and not the physical interface media rate, include the **logical-bandwidth-policer** statement. If a shaping rate is not configured for the logical interface, the physical interface media rate is used, even if you include the **logical-bandwidth-policer** You can configure the shaping rate on the logical interface using class-of-service statements.

[edit firewall policer *policer-name*]
logical-bandwidth-policer;

Example: Configuring a Logical Bandwidth Policer

This example applies a logical bandwidth policer rate to two logical interfaces on interface **ge-0/2/7**. The policed rate on **unit 0** is 2 Mbps (50 percent of 4 Mbps) and the policed rate on **unit 1** is 1 Mbps (50 percent of 2 Mbps).

```
[edit firewall]
policer Logical_Policer {
  logical-bandwidth-policer; # This applies the policer to logical interfaces
  if-exceeding {
    bandwidth-percent 50; # This applies 50 percent to the shaping-rate
    burst-size-limit 125k;
  }
  then discard;
}

[edit class-of-service]
interfaces {
  ge-0/2/7 {
    unit 0 {
      shaping-rate 4m # This establishes the rate to be policed on unit 0
    }
    unit 1 {
      shaping-rate 2m # This establishes the rate to be policed on unit 1
    }
  }
}
[edit interfaces ge-0/2/7]
```

```
per-unit-scheduler;
vlan-tagging;
unit 0 {
  vlan-id 100;
  family inet {
    policer {
      input Logical_Policer;
      output Logical_Policer;
    }
    address 172.1.1.1/30;
  }
}
unit 1 {
  vlan-id 200;
  family inet {
    policer {
      input Logical_Policer;
      output Logical_Policer;
    }
    address 172.2.1.1/30;
  }
}
```

Two-Color Policers and Shaping Rate Changes

When you configure a change in shaping rate, it is important to consider the effect on the bandwidth limit. Whenever the shaping rate changes, the bandwidth limit is adjusted based on whether a logical interface (unit) or bandwidth percentage policer is configured.

When a logical interface bandwidth policer is configured, the order of priority for the shaping rate (if configured at that level) is:

- The shaping rate applied to the logical interface (unit).
- The shaping rate applied to the physical interface (port).
- The physical interface speed.

When a bandwidth percentage policer is configured, the order of priority for the shaping rate (if configured at that level) is:

- The shaping rate applied to the physical interface (port).
- The physical interface speed.

These guidelines must be kept in mind when calculating the logical link speed and link speed from the configured shaping rate, which determines the rate-limited bandwidth after the policer is applied.

Related Documentation

- Example: Two-Color Policers and Shaping Rate Changes on page 83

Example: Two-Color Policers and Shaping Rate Changes

In this example, the shaping rate has been configured for the logical interface, but a bandwidth percentage policer is also configured. Therefore policing is based on the physical interface speed of 1 Gbps.

If both a shaping rate and a bandwidth percentage policer is configured on the same logical interface, the policing is based on the physical interface speed. Here is the example configuration:

```
[edit interfaces]
ge-0/1/0 {
  per-unit-scheduler;
  vlan-tagging;
  unit 0 {
    vlan-id 1;
    family inet {
      policer {
        output policer_test;
      }
      address 10.0.7.1/24;
    }
  }
}

[edit firewall]
policer policer_test {
  if-exceeding {
    bandwidth-percent 75;
    burst-size-limit 256k;
  }
  then discard;
}

[edit]
class-of-service {
  interfaces {
    ge-0/1/0 {
      unit 0 {
        shaping-rate 15m;
      }
    }
  }
}
```


CHAPTER 6

Configuring Tricolor Marking Policers

This topic discusses the following:

Policer Overview

Policing, or rate limiting, enables you to limit the amount of traffic that passes into or out of an interface. It is an essential component of firewall filters that is designed to thwart denial-of-service (DoS) attacks. Networks police traffic by limiting the input or output transmission rate of a class of traffic on the basis of user-defined criteria. Policing traffic allows you to control the maximum rate of traffic sent or received on an interface and to partition a network into multiple priority levels or classes of service.

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

Policing uses the *token-bucket algorithm*, which enforces a limit on average bandwidth while allowing bursts up to a specified maximum value. It offers more flexibility than the *leaky bucket algorithm* (see the *Junos OS Class of Service Configuration Guide*) in allowing a certain amount of bursty traffic before it starts discarding packets.

You can define specific classes of traffic on an interface and apply a set of rate limits to each. You can use a policer in one of two ways: as part of a filter configuration or as part of a logical interface (where the policer is applied to all traffic on that interface).

After you have defined and named a policer, it is stored as a template. You can later use the same policer name to provide the same policer configuration each time you wish to use it. This eliminates the need to define the same policer values more than once.

Juniper Networks routing platform architectures can support three types of policer:

- Single-rate two-color—A two-color policer (or "policer" when used without qualification) meters the traffic stream and classifies packets into two categories of packet loss priority (PLP) according to a configured bandwidth and burst-size limit. You can mark packets that exceed the bandwidth and burst-size limit in some way, or simply discard them. A policer is most useful for metering traffic at the port (physical interface) level.
- Single-rate three-color—This type of policer is defined in RFC 2697, *A Single Rate Three Color Marker*, as part of an assured forwarding (AF) per-hop-behavior (PHB) classification system for a Differentiated Services (DiffServ) environment. This type of policer meters traffic based on the configured committed information rate (CIR), committed burst size (CBS), and the excess burst size (EBS). Traffic is marked as belonging to one of three categories (green, yellow, or red) based on whether the packets arriving are below the CBS (green), exceed the CBS (yellow) but not the EBS, or exceed the EBS (red). A single-rate three-color policer is most useful when a service is structured according to packet length and not peak arrival rate.
- Two-rate three-color—This type of policer is defined in RFC 2698, *A Two Rate Three Color Marker*, as part of an assured forwarding (AF) per-hop-behavior (PHB) classification system for a Differentiated Services (DiffServ) environment. This type of policer meters traffic based on the configured CIR and peak information rate (PIR), along with their associated burst sizes, the CBS and *peak burst size* (PBS). Traffic is marked as belonging to one of three categories (green, yellow, or red) based on whether the packets arriving are below the CIR (green), exceed the CIR (yellow) but not the PIR, or exceed the PIR (red). A two-rate three-color policer is most useful when a service is structured according to arrival rates and not necessarily packet length.

Policer actions are implicit or explicit and vary by policer type. The term Implicit means that Junos assigns the loss-priority automatically. Table 14 on page 87 describes the policer actions.

Table 14: Policer Actions

Policer	Marking	Implicit Action	Configurable Action
Single-rate two-color	Green (Conforming)	Assign low loss priority	None
	Red (Nonconforming)	None	Assign low or high loss priority, assign a forwarding class, or discard On some platforms, you can assign medium-low or medium-high loss priority
Single-rate three-color	Green (Conforming)	Assign low loss priority	None
	Yellow (Above the CIR and CBS)	Assign medium-high loss priority	None
	Red (Above the EBS)	Assign high loss priority	Discard
Two-rate three-color	Green (Conforming)	Assign low loss priority	None
	Yellow (Above the CIR and CBS)	Assign medium-high loss priority	None
	Red (Above the PIR and PBS)	Assign high loss priority	Discard

You can configure policers at the queue, logical interface, or Layer 2 (MAC) level. Only a single policer is applied to a packet at the egress queue, and the search for policers occurs in this order:

- Queue level
- Logical interface level
- Layer 2 (MAC) level

Three-color policers are not bound by a green-yellow-red coloring convention. Packets are marked with low, medium-high, or high PLP bit configurations based on color, so both three-color policer schemes extend the functionality of class-of-service (CoS) traffic policing by providing three levels of drop precedence (loss priority) instead of the two normally available in port-level policers. Both single-rate and two-rate three-color policer schemes can operate in two modes:

- Color-blind—In color-blind mode, the three-color policer assumes that all packets examined have not been previously marked or metered. In other words, the three-color policer is “blind” to any previous coloring a packet might have had.
- Color-aware—In color-aware mode, the three-color policer assumes that all packets examined have been previously marked or metered. In other words, the three-color policer is “aware” of the previous coloring a packet might have had. In color-aware mode, the three-color policer can increase the PLP of a packet, but never decrease it. For example, if a color-aware three-color policer meters a packet with a medium PLP marking, it can raise the PLP level to high, but cannot reduce the PLP level to low.

NOTE: We recommend you use the naming convention *policertypeTCM#-color type* when configuring three-color policers and *policer#* when configuring two-color policers. TCM stands for three-color marker. Because policers can be numerous and must be applied correctly to work, a simple naming convention makes it easier to apply the policers properly.

For example, the first single-rate, color-aware three-color policer configured would be named **srTCM1-ca**. The second two-rate, color-blind three-color configured would be named **trTCM2-cb**.

Platform Support for Tricolor Marking

Tricolor marking is supported on the following Juniper Networks routers:

- M120 Multiservice Edge Routers
- M320 Multiservice Edge Routers and T Series Core Routers with Enhanced II Flexible PIC Concentrators (FPCs)
- MX Series Ethernet Services Routers
- T640 Core Routers with Enhanced Scaling FPC4

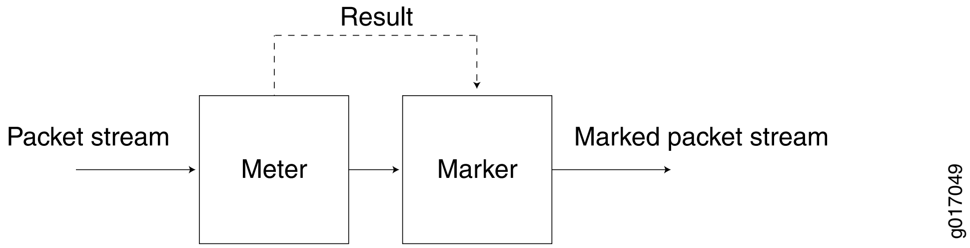
NOTE: On MX Series and M120 routers, you can apply three-color policers to aggregated interfaces.

Tricolor Marking Architecture

Policers provide two functions: metering and marking.

The policer meters each packet and passes the packet and the metering result to the marker, as shown in Figure 9 on page 89.

Figure 9: Flow of Tricolor Marking Policer Operation



The meter operates in two modes. In the color-blind mode, the meter treats the packet stream as uncolored. Any preset loss priorities are ignored. In the color-aware mode, the meter inspects the packet loss priority (PLP) field, which has been set by an upstream device as PLP high, medium-high, medium-low, or low; in other words, the PLP field has already been set by a behavior aggregate (BA) or multifield classifier. The marker changes the PLP of each incoming IP packet according to the results of the meter. For more information, see "Configuring Two-Rate Tricolor Marking" on page 94.

This chapter emphasizes configuration and use of TCM policers. For more information about configuring and using two-color policers ("policers"), see the *Junos OS Policy Framework Configuration Guide*.

Single-rate TCM is so called because traffic is policed according to one rate—the CBR—and two burst sizes: the CBS and EBS. The CIR specifies the average rate at which bits are admitted to the network. The CBS specifies the usual burst size in bytes and the EBS specifies the maximum burst size in bytes for packets that are admitted to the network. The EBS is greater than or equal to the CBS, and neither can be 0. As each packet enters the network, its bytes are counted. Packets that do not exceed the CBS are marked low PLP. Packets that exceed the CBS but are below the EBS are marked medium-high PLP. Packets that exceed the PIR are marked high PLP.

Two-rate TCM is so called because traffic is policed according to two rates: the CIR and the PIR. The PIR is greater than or equal to the CIR. The CIR specifies the average rate at which bits are admitted to the network and the PIR specifies the maximum rate at which bits are admitted to the network. As each packet enters the network, its bits are counted. Bits in packets that do not exceed the CIR have their packets marked low PLP. Bits in packets that exceed the CIR but are below the PIR have their packets marked medium-high PLP. Bits in packets that exceed the PIR have their packets marked high PLP.

For information about how to use marking policers with BA and multifield classifiers, see "Using BA Classifiers to Set PLP" on page 102 and "Using Multifield Classifiers to Set PLP" on page 103.

Configuring Tricolor Marking

You configure marking policers by defining the policer and multiple levels of PLP for classifiers, rewrite rules, random early detection (RED) drop profiles, and firewall filters. To configure marking policers, include the following statements at the **[edit class-of-service]** hierarchy level:

```
[edit class-of-service]
tri-color;
classifiers {
  (dscp | dscp-ipv6 | exp | ieee-802.1 | inet-precedence) classifier-name {
    import classifier-name | default);
    forwarding-class class-name {
      loss-priority (low | medium-low | medium-high | high) code-points [ aliases ]
        [ bit-patterns ];
    }
  }
}
rewrite-rules {
  (dscp | dscp-ipv6 | exp | ieee-802.1 | inet-precedence) rewrite-name {
    import (rewrite-name | default);
    forwarding-class class-name {
      loss-priority (low | medium-low | medium-high | high) code-point (aliases |
        bit-patterns;
    }
  }
}
schedulers {
  scheduler-name {
    drop-profile-map loss-priority (any | low | medium-low | medium-high | high)protocol
      any drop-profile profile-name;
  }
}

[edit firewall]
policer name {
  then loss-priority (low | medium-low | medium-high | high);
}
three-color-policer policer-name {
  action {
    loss-priority high then discard; # Only for IQ2 PICs
  }
  logical-interface-policer;
  single-rate {
    (color-aware | color-blind);
    committed-information-rate bps;
    committed-burst-size bytes;
    excess-burst-size bytes;
  }
  two-rate {
    (color-aware | color-blind);
    committed-information-rate bps;
    committed-burst-size bytes;
    peak-information-rate bps;
    peak-burst-size bytes;
  }
}
filter filter-name {
  <family family> {
    term rule-name {
      then {
        three-color-policer (single-rate | two-rate) policer-name;
      }
```



```
        }
      }
    }
```

Tricolor Marking Limitations

Tricolor Marking (TCM) has some limitations that must be kept in mind during configuration and operation.

The following limitations apply to TCM:

- When you enable TCM on a 10-port Gigabit Ethernet PIC or a 10-Gigabit Ethernet PIC, for queues 6 and 7 only, the output of the **show interfaces queue** ***interface-name*** command does not display the number of queued bytes and packets, or the number of bytes and packets dropped due to RED. If you do not configure tricolor marking on the interface, these statistics are available for all queues.
- When you enable TCM, Transmission Control Protocol (TCP)-based configurations for drop profiles are rejected. In other words, you cannot include the **protocol** statement at the **[edit class-of-service schedulers** ***scheduler-name*** **drop-profile-map]** hierarchy level. The result is that drop profiles are applied to packets with the specified PLP and any protocol type.
- On Gigabit Ethernet IQ PICs, for IEEE 802.1 rewrite rules, only two loss priorities are supported. Exiting packets with medium-high loss priority are treated as high, and packets with medium-low loss priority are treated as low. In other words rewrite rules corresponding to high and low apply instead of those corresponding to medium-high and medium-low. For IQ PICs, you can only configure one IEEE 802.1 rewrite rule on a physical port. All logical ports (units) on that physical port should apply the same IEEE 802.1 rewrite rule.
- When some PICs with Frame Relay encapsulation mark a packet with high loss priority, the packet is treated as having medium-high loss priority on M320 Multiservice Edge Routers and T Series Core Routers with Enhanced II FPCs and T640 Core Routers with Enhanced Scaling FPC4.
- TCM is not supported on aggregated Ethernet and aggregated SONET/SDH interfaces.
- In a single firewall filter term, you cannot configure both the **loss-priority** action modifier and the **three-color-policer** action modifier. These statements are mutually exclusive.

Configuring Single-Rate Tricolor Marking

With TCM, you can configure traffic policing according to two separate modes—color-blind and color-aware. In color-blind mode, the current PLP value is ignored. In color-aware mode, the current PLP values are considered by the policer and can only be increased.

- Configuring Color-Blind Mode for Single-Rate Tricolor Marking on page 92
- Configuring Color-Aware Mode for Single-Rate Tricolor Marking on page 92

Configuring Color-Blind Mode for Single-Rate Tricolor Marking

All packets are evaluated by the CBS. If a packet exceeds the CBS, it is evaluated by the EBS. In color-blind mode, the policer supports three loss priorities only: low, medium-high, and high.

In color-blind mode, packets that exceed the CBS but are below the EBS are marked yellow (medium-high). Packets that exceed the EBS are marked red (high), as shown in Table 15 on page 92.

Table 15: Color-Blind Mode TCM Color-to-PLP Mapping

Color	PLP	Meaning
Green	**low**	Packet does not exceed the CBS.
Yellow	**medium-high**	Packet exceeds the CBS but does not exceed the EBS.
Red	**high**	Packet exceeds the EBS.

If you are using color-blind mode and you wish to configure an output policer that marks packets to have medium-low loss priority, you must configure a policer at the **[edit firewall policer *policer-name*]** hierarchy level. For example:

```
firewall {
  policer 4PLP {
    if-exceeding {
      bandwidth-limit 40k;
      burst-size-limit 4k;
    }
    then loss-priority medium-low;
  }
}
```

Apply this policer at one or both of the following hierarchy levels:

- **[edit firewall family *family* filter *filter-name* term *rule-name* then policer *policer-name*]**
- **[edit interfaces *interface-name* unit *logical-unit-number* family *family* filter *filter-name*]**

Configuring Color-Aware Mode for Single-Rate Tricolor Marking

In color-aware mode, the metering treatment the packet receives depends on its classification. Metering can increase a packet's preassigned PLP, but cannot decrease it, as shown in Table 16 on page 93.

Table 16: Color-Aware Mode TCM PLP Mapping

Incoming PLP	Packet Metered Against	Possible Cases	Outgoing PLP
low	CBS and EBS	Packet does not exceed the CBS.	low
		Packet exceeds the CBS but not the EBS.	medium-high
		Packet exceeds the EBS.	high
medium-low	EBS only	Packet does not exceed the CBS.	medium-low
		Packet does not exceed the EBS.	medium-low
		Packet exceeds the EBS.	high
medium-high	EBS only	Packet does not exceed the CBS.	medium-high
		Packet does not exceed the EBS.	medium-high
		Packet exceeds the EBS.	high
high	Not metered by the policer.	All cases.	high

The following sections describe single-rate color-aware PLP mapping in more detail.

Effect on Low PLP of Single-Rate Policer

Packets belonging to the green class have already been marked by a classifier with low PLP. The marking policer can leave the packet's PLP unchanged or increase the PLP to medium-high or high. Therefore, these packets are metered against both the CBS and the EBS.

For example, if a BA or multifield classifier marks a packet with low PLP according to the type-of-service (ToS) bits in the IP header, and the two-rate TCM policer is in color-aware mode, the output loss priority is as follows:

- If the rate of traffic flow is less than the CBS, packets remain marked as low PLP.
- If the rate of traffic flow is greater than the CBS but less than the EBS, some of the packets are marked as medium-high PLP, and some of the packets remain marked as low PLP.
- If the rate of traffic flow is greater than the EBS, some of the packets are marked as high PLP, and some of the packets remain marked as low PLP.

Effect on Medium-Low PLP of Single-Rate Policer

Packets belonging to the yellow class have already been marked by a classifier with medium-low or medium-high PLP. The marking policer can leave the packet's PLP unchanged or increase the PLP to high. Therefore, these packets are metered against the EBS only.

For example, if a BA or multifield classifier marks a packet with medium-low PLP according to the ToS bits in the IP header, and the two-rate TCM policer is in color-aware mode, the output loss priority is as follows:

- If the rate of traffic flow is less than the CBS, packets remain marked as medium-low PLP.
- If the rate of traffic flow is greater than the CBS but less than the EBS, packets remain marked as medium-low PLP.
- If the rate of traffic flow is greater than the EBS, some of the packets are marked as high PLP, and some of the packets remain marked as medium-low PLP.

Effect on Medium-High PLP of Single-Rate Policer

Packets belonging to the yellow class have already been marked by a classifier with medium-low or medium-high PLP. The marking policer can leave the packet's PLP unchanged or increase the PLP to high. Therefore, these packets are metered against the EBS only.

For example, if a BA or multifield classifier marks a packet with medium-high PLP according to the ToS bits in the IP header, and the two-rate TCM policer is in color-aware mode, the output loss priority is as follows:

- If the rate of traffic flow is less than the CBS, packets remain marked as medium-high PLP.
- If the rate of traffic flow is greater than the CBS but less than the EBS, packets remain marked as medium-high PLP.
- If the rate of traffic flow is greater than the EBS, some of the packets are marked as high PLP, and some of the packets remain marked as medium-high PLP.

Effect on High PLP of Single-Rate Policer

Packets belonging to the red class have already been marked by a classifier with high PLP. The marking policer can only leave the packet's PLP unchanged. Therefore, these packets are not metered against the CBS or the EBS and all the packets remain marked as high PLP.

Configuring Two-Rate Tricolor Marking

With TCM, you can configure traffic policing according to two separate modes—color-blind and color-aware. In color-blind mode, the current PLP value is ignored. In color-aware mode, the current PLP values are considered by the policer and can only be increased.

- Configuring Color-Blind Mode for Two-Rate Tricolor Marking on page 95
- Configuring Color-Aware Mode for Two-Rate Tricolor Marking on page 95

Configuring Color-Blind Mode for Two-Rate Tricolor Marking

All packets are evaluated by the CIR. If a packet exceeds the CIR, it is evaluated by the PIR. In color-blind mode, the policer supports three loss priorities only: low, medium-high, and high.

In color-blind mode, packets that exceed the CIR but are below the PIR are marked yellow (medium-high). Packets that exceed the PIR are marked red (high), as shown in Table 17 on page 95.

Table 17: Color-Blind Mode TCM Color-to-PLP Mapping

Color	PLP	Meaning
Green	**low**	Packet does not exceed the CIR.
Yellow	**medium-high**	Packet exceeds the CIR but does not exceed the PIR.
Red	**high**	Packet exceeds the PIR.

If you are using color-blind mode and you wish to configure an output policer that marks packets to have medium-low loss priority, you must configure a policer at the **[edit firewall policer *policer-name*]** hierarchy level. For example:

```
firewall {
  policer 4PLP {
    if-exceeding {
      bandwidth-limit 40k;
      burst-size-limit 4k;
    }
    then loss-priority medium-low;
  }
}
```

Apply this policer at one or both of the following hierarchy levels:

- **[edit firewall family *family* filer *filter-name* term *rule-name* then policer *policer-name*]**
- **[edit interfaces *interface-name* unit *logical-unit-number* family *family* filter *filter-name*]**

Configuring Color-Aware Mode for Two-Rate Tricolor Marking

In color-aware mode, the metering treatment the packet receives depends on its classification. Metering can increase a packet's preassigned PLP, but cannot decrease it, as shown in Table 18 on page 96.

Table 18: Color-Aware Mode TCM Mapping

Incoming PLP	Packet Metered Against	Possible Cases	Outgoing PLP
low	CIR and PIR	Packet does not exceed the CIR.	low
		Packet exceeds the CIR but not the PIR.	medium-high
		Packet exceeds the PIR.	high
medium-low	PIR only	Packet does not exceed the CIR.	medium-low
		Packet does not exceed the PIR.	medium-low
		Packet exceeds the PIR.	high
medium-high	PIR only	Packet does not exceed the CIR.	medium-high
		Packet does not exceed the PIR.	medium-high
		Packet exceeds the PIR.	high
high	Not metered by the policer.	All cases.	high

The following sections describe color-aware two-rate PLP mapping in more detail.

Effect on Low PLP of Two-Rate Policer

Packets belonging to the green class have already been marked by a classifier with low PLP. The marking policer can leave the packet's PLP unchanged or increase the PLP to medium-high or high. Therefore, these packets are metered against both the CIR and the PIR.

For example, if a BA or multifield classifier marks a packet with low PLP according to the ToS bits in the IP header, and the two-rate TCM policer is in color-aware mode, the output loss priority is as follows:

- If the rate of traffic flow is less than the CIR, packets remain marked as low PLP.
- If the rate of traffic flow is greater than the CIR but less than the PIR, some of the packets are marked as medium-high PLP, and some of the packets remain marked as low PLP.
- If the rate of traffic flow is greater than the PIR, some of the packets are marked as high PLP, and some of the packets remain marked as low PLP.

Effect on Medium-Low PLP of Two-Rate Policer

Packets belonging to the yellow class have already been marked by a classifier with medium-low or medium-high PLP. The marking policer can leave the packet's PLP unchanged or increase the PLP to high. Therefore, these packets are metered against the PIR only.

For example, if a BA or multifield classifier marks a packet with medium-low PLP according to the ToS bits in the IP header, and the two-rate TCM policer is in color-aware mode, the output loss priority is as follows:

- If the rate of traffic flow is less than the CIR, packets remain marked as medium-low PLP.
- If the rate of traffic flow is greater than the CIR/CBS but less than the PIR, packets remain marked as medium-low PLP.
- If the rate of traffic flow is greater than the PIR, some of the packets are marked as high PLP, and some of the packets remain marked as medium-low PLP.

Effect on Medium-High PLP of Two-Rate Policer

Packets belonging to the yellow class have already been marked by a classifier with medium-low or medium-high PLP. The marking policer can leave the packet's PLP unchanged or increase the PLP to high. Therefore, these packets are metered against the PIR only.

For example, if a BA or multifield classifier marks a packet with medium-high PLP according to the ToS bits in the IP header, and the two-rate TCM policer is in color-aware mode, the output loss priority is as follows:

- If the rate of traffic flow is less than the CIR, packets remain marked as medium-high PLP.
- If the rate of traffic flow is greater than the CIR but less than the PIR, packets remain marked as medium-high PLP.
- If the rate of traffic flow is greater than the PIR, some of the packets are marked as high PLP, and some of the packets remain marked as medium-high PLP.

Effect on High PLP of Two-Rate Policer

Packets belonging to the red class have already been marked by a classifier with high PLP. The marking policer can only leave the packet's PLP unchanged. Therefore, these packets are not metered against the CIR or the PIR and all the packets remain marked as high PLP.

Enabling Tricolor Marking

By default, TCM is enabled on M120 and MX Series routers. To enable TCM on other routers, include the **tri-color** statement at the **[edit class-of-service]** hierarchy level:

```
[edit class-of-service]
tri-color;
```

This statement is necessary on the following routers:

- M320 and T Series routers with Enhanced II FPCs

- T640 routers with Enhanced Scaling FPC4s

If you do not include this statement in the configuration on platforms that require it, you cannot configure medium-low or medium-high PLP for classifiers, rewrite rules, drop profiles, or firewall filters.

Configuring Tricolor Marking Policers

A tricolor marking policer polices traffic on the basis of metering rates, including the CIR, the PIR, their associated burst sizes, and any policing actions configured for the traffic. To configure a tricolor marking policer, include the following statements at the **[edit firewall]** hierarchy level:

```
[edit firewall]
three-color-policer name {
  action {
    loss-priority high then discard; # Only for IQ2 PICs
  }
  logical-interface-policer;
  single-rate {
    (color-aware | color-blind);
    committed-information-rate bps;
    committed-burst-size bytes;
    excess-burst-size bytes;
  }
  two-rate {
    (color-aware | color-blind);
    committed-information-rate bps;
    committed-burst-size bytes;
    peak-information-rate bps;
    peak-burst-size bytes;
  }
}
```

You can configure a tricolor policer to discard high loss priority traffic on a logical interface in the ingress or egress direction. To configure a policer on a logical interface using tricolor marking policing to discard high loss priority traffic, include the **logical-interface-policer** statement and **action** statement.

In all cases, the range of allowable bits-per-second or byte values is 1500 to 100,000,000,000. You can specify the values for bps and bytes either as complete decimal numbers or as decimal numbers followed by the abbreviation **k** (1000), **m** (1,000,000), or **g** (1,000,000,000).

The color-aware policer implicitly marks packets into four loss priority categories:

- Low
- Medium-low
- Medium-high
- High

The color-blind policer implicitly marks packets into three loss priority categories:

- Low
- Medium-high
- High

Table 19 on page 99 describes all the configurable TCM statements.

Table 19: Tricolor Marking Policer Statements

Statement	Meaning	Configurable Values
single-rate	Marking is based on the CIR, CBS, and EBS.	–
two-rate	Marking is based on the CIR, PIR, and rated burst sizes.	–
color-aware	Metering depends on the packet's preclassification. Metering can increase a packet's assigned PLP, but cannot decrease it.	–
color-blind	All packets are evaluated by the CIR or CBS. If a packet exceeds the CIR or CBS, it is evaluated by the PIR or EBS.	–
committed-information-rate	Guaranteed bandwidth under normal line conditions and the average rate up to which packets are marked green.	1500 through 100,000,000,000 bps
committed-burst-size	Maximum number of bytes allowed for incoming packets to burst above the CIR, but still be marked green.	1500 through 100,000,000,000 bytes
excess-burst-size	Maximum number of bytes allowed for incoming packets to burst above the CIR, but still be marked yellow.	1500 through 100,000,000,000 bytes
peak-information-rate	Maximum achievable rate. Packets that exceed the CIR but are below the PIR are marked yellow. Packets that exceed the PIR are marked red.	1500 through 100,000,000,000 bps
peak-burst-size	Maximum number of bytes allowed for incoming packets to burst above the PIR, but still be marked yellow.	1500 through 100,000,000,000 bytes

Applying Tricolor Marking Policers to Firewall Filters

To rate-limit traffic by applying a tricolor marking policer to a firewall filter, include the **three-color-policer** statement:

```
three-color-policer {
    (single-rate | two-rate) policer-name;
}
```

You can include this statement at the following hierarchy levels:

- **[edit firewall family *family* filter *filter-name* term *rule-name* then]**
- **[edit firewall filter *filter-name* term *rule-name* then]**

In the **family** statement, the protocol family can be **any**, **ccc**, **inet**, **inet6**, **mpls**, or **vpls**.

You must identify the referenced policer as a **single-rate** or **two-rate** policer, and this statement must match the configured TCM policer. Otherwise, an error message appears in the configuration listing.

For example, if you configure **srTCM** as a single-rate TCM policer and try to apply it as a two-rate policer, the following message appears:

```
[edit firewall]
user@host# show three-color-policer srTCM
single-rate {
  color-aware;
  . . .
}
user@host# show filter TESTER
term A {
  then {
    three-color-policer {
      ##
      ## Warning: Referenced two-rate policer does not exist
      ##
      two-rate srTCM;
    }
  }
}
```

Example: Applying a Two-Rate Tricolor Marking Policer to a Firewall Filter

Apply the **trtcm1-cb** policer to a firewall filter:

```
firewall {
  three-color-policer trtcm1-cb {  # Configure the trtcm1-cb policer.
  two-rate {
    color-blind;
    committed-information-rate 1048576;
    committed-burst-size 65536;
    peak-information-rate 10485760;
    peak-burst-size 131072;
  }
}
filter fil {   # Configure the fil firewall filter, applying the trtcm1-cb policer.
term default {
  then {
    three-color-policer {
      two-rate trtcm1-cb;
    }
  }
}
```

For more information about applying policers to firewall filters, see the *Junos OS Policy Framework Configuration Guide*.

Applying Firewall Filter Tricolor Marking Policers to Interfaces

To apply a tricolor marking policer to an interface, you must reference the filter name in the interface configuration. To do this, include the **filter** statement:

```
filter {
  input filter-name;
  output filter-name;
}
```

You can include these statements at the following hierarchy levels:

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

The filter name that you reference should have an attached tricolor marking policer, as shown in "Applying Tricolor Marking Policers to Firewall Filters" on page 99.

Example: Applying a Single-Rate Tricolor Marking Policer to an Interface

Apply the **trtcm1-cb** policer to an interface:

```
firewall {
  three-color-policer srtcm1 {   # Configure the srtcm1-cb policer.
  single-rate {
    color-blind;
    committed-information-rate 1048576;
    committed-burst-size 65536;
    excess-burst-size 131072;
  }
}
filter fil {   # Configure the fil firewall filter, applying the srtcm1-cb policer.
term default {
  then {
    three-color-policer {
      single-rate srtcm1-cb;   # The TCM policer must be single-rate.
    }
  }
}
interfaces {   # Configure the interface, which attaches the fil firewall filter.
so-1/0/0 {
  unit 0 {
    family inet {
      filter {
        input fil;
      }
    }
  }
}
```

Applying Layer 2 Policers to Gigabit Ethernet Interfaces

To rate-limit traffic by applying a policer to a Gigabit Ethernet interface (or a 10-Gigabit Ethernet interface [**xe-*fpc/pic/port***]), include the **layer2-policer** statement with the direction, type, and name of the policer:

```
[edit interfaces ge-fpc/pic/port unit 0]
layer2-policer {
  input-policer policer-name;
```

```
    input-three-color policer-name;
    output-policer policer-name;
    output-three-color policer-name;
}
```

The direction (input or output) and type (policer or three-color) are combined into one statement and the policer named must be properly configured.

One input or output policer of either type can be configured on the interface.

Examples: Applying Layer 2 Policers to a Gigabit Ethernet Interface

Apply color-blind and color-aware two-rate TCM policers as input and output policers to a Gigabit Ethernet interface:

```
ge-1/0/0 {
    unit 0
    layer2-policer {
        input-three-color trTCM1-cb;    # Apply the trTCM1-color-blind policer.
        output-three-color trTCM1-ca;    # Apply the trTCM1-color-aware policer.
    }
}
```

Apply two-level and color-blind single-rate TCM policers as input and output policers to a Gigabit Ethernet interface:

```
ge-1/0/0 {
    unit 1
    layer2-policer {
        input-policer two-color-policer;    # Apply a two-color policer.
        output-three-color srTCM2-cb;    # Apply the srTCM1-color-blind policer.
    }
}
```

Apply a color-aware single-rate TCM policer as output policer on a Gigabit Ethernet interface:

```
ge-1/0/0 {
    unit 2
    layer2-policer {
        output-three-color srTCM3-ca {    # Apply the srTCM3-color-aware policer.
    }
}
```

Using BA Classifiers to Set PLP

Behavior aggregate (BA) classifiers take action on incoming packets. When TCM is enabled, Juniper Networks M320 Multiservice Edge Routers and T Series Core Routers support four classifier PLP designations: **low**, **medium-low**, **medium-high**, and **high**. To configure the PLP for a classifier, include the following statements at the **[edit class-of-service]** hierarchy level:

```
[edit class-of-service]
classifiers {
    (dscp | dscp-ipv6 | exp | ieee-802.1 | inet-precedence) classifier-name {
        import (classifier-name | default);
```



```
      forwarding-class class-name {
        loss-priority (low | medium-low | medium-high | high) code-points [ aliases ]
          [ bit-patterns ];
      }
    }
  }
```

The inputs for a classifier are the CoS values. The outputs for a classifier are the forwarding class and the loss priority (PLP). A classifier sets the forwarding class and the PLP for each packet entering the interface with a specific set of CoS values.

For example, in the following configuration, the **assured-forwarding** forwarding class and **medium-low** PLP are assigned to all packets entering the interface with the **101110** CoS values:

```
class-of-service {
  classifiers {
    dscp dscp-cl {
      forwarding-class assured-forwarding {
        loss-priority medium-low {
          code-points 101110;
        }
      }
    }
  }
}
```

To use this classifier, you must configure the settings for the **assured-forwarding** forwarding class at the **[edit class-of-service forwarding-classes queue *queue-number* assured-forwarding]** hierarchy level. For more information, see "Overview of Forwarding Classes" on page 111.

Using Multifield Classifiers to Set PLP

Multifield classifiers take action on incoming or outgoing packets, depending whether the firewall rule is applied as an input filter or an output filter. When TCM is enabled, Juniper Networks M320 Multiservice Edge Routers and T Series Core Routers support four multifield classifier PLP designations: **low**, **medium-low**, **medium-high**, and **high**.

To configure the PLP for a multifield classifier, include the **loss-priority** statement in a policer or firewall filter that you configure at the at the **[edit firewall]** hierarchy level:

```
[edit firewall]
family family-name {
  filter filter-name {
    term term-name {
      from {
        match-conditions;
      }
      then {
        loss-priority (low | medium-low | medium-high | high);
        forwarding-class class-name;
      }
    }
  }
```

```
}
```

The inputs (match conditions) for a multifield classifier are one or more of the six packet header fields: destination address, source address, IP protocol, source port, destination port, and DSCP. The outputs for a multifield classifier are the forwarding class and the loss priority (PLP). In other words, a multifield classifier sets the forwarding class and the PLP for each packet entering or exiting the interface with a specific destination address, source address, IP protocol, source port, destination port, or DSCP.

For example, in the following configuration, the forwarding class **expedited-forwarding** and PLP **medium-high** are assigned to all IPv4 packets with the **10.1.1.0/24** or **10.1.2.0/24** source address:

```
firewall {
  family inet {
    filter classify-customers {
      term isp1-customers {
        from {
          source-address 10.1.1.0/24;
          source-address 10.1.2.0/24;
        }
        then {
          loss-priority medium-high;
          forwarding-class expedited-forwarding;
        }
      }
    }
  }
}
```

To use this classifier, you must configure the settings for the **expedited-forwarding** forwarding class at the **[edit class-of-service forwarding-classes queue *queue-number* expedited-forwarding]** hierarchy level. For more information, see “Overview of Forwarding Classes” on page 111.

Configuring PLP for Drop-Profile Maps

RED drop profiles take action on outgoing packets. When TCM is enabled, M320 and T Series routers support four drop-profile map PLP designations: **low**, **medium-low**, **medium-high**, and **high**.

To configure the PLP for the drop-profile map, include the **schedulers** statement at the **[edit class-of-service]** hierarchy level:

```
[edit class-of-service]
schedulers {
  scheduler-name {
    drop-profile-map loss-priority (any | low | medium-low | medium-high | high)protocol
      any drop-profile profile-name;
  }
}
```

When you configure TCM, the drop-profile map’s protocol type must be **any**.

The inputs for a drop-profile map are the loss priority and the protocol type. The output for a drop-profile map is the drop profile name. In other words, the map sets the drop profile for each packet with a specific PLP and protocol type exiting the interface.

For example, in the following configuration, the **dp** drop profile is assigned to all packets exiting the interface with a medium-low PLP and belonging to any protocol:

```
class-of-service {
  schedulers {
    af {
      drop-profile-map loss-priority medium-low protocol any drop-profile dp;
    }
  }
}
```

To use this drop-profile map, you must configure the settings for the **dp** drop profile at the **[edit class-of-service drop-profiles dp]** hierarchy level. For more information, see "RED Drop Profiles Overview" on page 231.

Configuring Rewrite Rules Based on PLP

Rewrite rules take action on outgoing packets. When TCM is enabled, M320 and T Series routers support four rewrite PLP designations: **low**, **medium-low**, **medium-high**, and **high**. To configure the PLP for a rewrite rule, include the following statements at the **[edit class-of-service]** hierarchy level:

```
[edit class-of-service]
rewrite-rules {
  (dscp | dscp-ipv6 | exp | ieee-802.1 | inet-precedence) rewrite-name {
    import (rewrite-name | default);
    forwarding-class class-name {
      loss-priority (low | medium-low | medium-high | high) code-point (alias | bits);
    }
  }
}
```

The inputs for a rewrite rule are the forwarding class and the loss priority (PLP). The output for a rewrite rule are the CoS values. In other words, a rewrite rule sets the CoS values for each packet exiting the interface with a specific forwarding class and PLP.

For example, if you configure the following, the **000000** CoS values are assigned to all packets exiting the interface with the **assured-forwarding** forwarding class and **medium-high** PLP:

```
class-of-service {
  rewrite-rules {
    dscp dscp-rw {
      forwarding-class assured-forwarding {
        loss-priority medium-high code-point 000000;
      }
    }
  }
}
```


To use this classifier, you must configure the settings for the **assured-forwarding** forwarding class at the **[edit class-of-service forwarding-classes queue *queue-number* assured-forwarding]** hierarchy level. For more information, see "Overview of Forwarding Classes" on page 111.

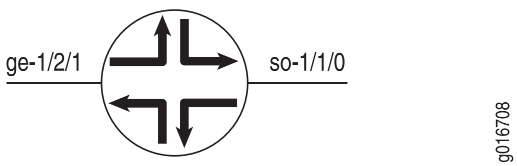
Example: Configuring and Verifying Two-Rate Tricolor Marking

This example configures a two-rate tricolor marking policer on an input Gigabit Ethernet interface and shows commands to verify its operation.

Traffic enters the Gigabit Ethernet interface and exits a SONET/SDH OC12 interface. Oversubscription occurs when you send line-rate traffic from the Gigabit Ethernet interface out the OC12 interface.

Figure 10 on page 106 shows the sample topology.

Figure 10: Tricolor Marking Sample Topology



- Applying a Policer to the Input Interface on page 106
- Applying Profiles to the Output Interface on page 107
- Marking Packets with Medium-Low Loss Priority on page 108
- Verifying Two-Rate Tricolor Marking Operation on page 109

Applying a Policer to the Input Interface

The tricolor marking and policer are applied on the ingress Gigabit Ethernet interface. Incoming packets are metered. Packets that do not exceed the CIR are marked with low loss priority. Packets that exceed the CIR but do not exceed the PIR are marked with medium-high loss priority. Packets that exceed the PIR are marked with high loss priority.

```
[edit]
interfaces {
   ge-1/2/1 {
      unit 0 {
         family inet {
            filter {
               input trtcm-filter;
            }
         }
      }
   }
}
firewall {
   three-color-policer trtcm1 {
      two-rate {
         color-aware;
         committed-information-rate 100m;
```



```
            committed-burst-size 65536;
            peak-information-rate 200m;
            peak-burst-size 131072;
        }
    }
    filter trtcm-filter {
        term one {
            then {
                three-color-policer {
                    two-rate trtcm1;
                }
            }
        }
    }
}
```

Applying Profiles to the Output Interface

Transmission scheduling and weighted random early detection (WRED) profiles are applied on the output OC12 interface. The software drops traffic in the low, medium-high, and high drop priorities proportionally to the configured drop profiles.

```
[edit]
class-of-service {
    drop-profiles {
        low-tcm {
            fill-level 80 drop-probability 100;
        }
        med-tcm {
            fill-level 40 drop-probability 100;
        }
        high-tcm {
            fill-level 10 drop-probability 100;
        }
    }
    tri-color;
    interfaces {
        so-1/1/0 {
            scheduler-map tcm-sched;
        }
    }
    scheduler-maps {
        tcm-sched {
            forwarding-class queue-0 scheduler q0-sched;
            forwarding-class queue-3 scheduler q3-sched;
        }
    }
    schedulers {
        q0-sched {
            transmit-rate percent 50;
            buffer-size percent 50;
            drop-profile-map loss-priority low protocol any drop-profile low-tcm;
            drop-profile-map loss-priority medium-high protocol any drop-profile med-tcm;
            drop-profile-map loss-priority high protocol any drop-profile high-tcm;
        }
        q3-sched {
```

```
            transmit-rate percent 50;
            buffer-size percent 50;
        }
    }
}
```

Marking Packets with Medium-Low Loss Priority

In another example, the 4PLP filter and policer causes certain packets to be marked with medium-low loss priority.

```
interfaces {
    ge-7/2/0 {
        unit 0 {
            family inet {
                filter {
                    input 4PLP;
                }
                policer {
                    input 4PLP;
                }
                address 10.45.10.2/30;
            }
        }
    }
}

firewall {
    three-color-policer trTCM {
        two-rate {
            color-blind;
            committed-information-rate 400m;
            committed-burst-size 100m;
            peak-information-rate 1g;
            peak-burst-size 500m;
        }
    }
    policer 4PLP {
        if-exceeding {
            bandwidth-limit 40k;
            burst-size-limit 4k;
        }
        then loss-priority medium-low;
    }
    family inet {
        filter 4PLP {
            term 0 {
                from {
                    precedence 1;
                }
                then loss-priority medium-low;
            }
        }
        filter filter_trTCM {
            term default {
                then {
```



```
                three-color-policer {
                    two-rate trTCM;
                }
            }
        }
    }
}
```

Verifying Two-Rate Tricolor Marking Operation

The following operational mode commands are useful for checking the results of your configuration:

- **show class-of-service forwarding-table classifiers**
- **show interfaces *interface-name* extensive**
- **show interfaces queue *interface-name***

For information about these commands, see the *Junos OS Interfaces Command Reference* and *Junos OS System Basics and Services Command Reference*.

CHAPTER 7

Configuring Forwarding Classes

This topic discusses the following:

Overview of Forwarding Classes

It is helpful to think of forwarding classes as output queues. In effect, the end result of classification is the identification of an output queue for a particular packet.

For a classifier to assign an output queue to each packet, it must associate the packet with one of the following forwarding classes:

- 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 best effort 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.

For Juniper Networks M Series Multiservice Edge Routers (except the M320), you can configure up to four forwarding classes, one of each type: expedited forwarding (EF), assured forwarding (AF), best effort (BE), and network control (NC).

The Juniper Networks M320 Multiservices Edge Routers and T Series Core Routers support 16 forwarding classes, enabling you to classify packets more granularly. For example, you can configure multiple classes of EF traffic: EF, EF1, and EF2. The software supports up to eight output queues; therefore, if you configure more than eight forwarding classes, you must map multiple forwarding classes to single output queues. For more information, see "Configuring Up to 16 Forwarding Classes" on page 120.

By default, the loss priority is low. On most routers, you can configure high or low loss priority. On the following routers you can configure high, low, medium-high, or medium-low loss priority:

- J Series Services Router interfaces
- M320 routers and T Series routers with Enhanced II Flexible PIC Concentrators (FPCs)
- T640 routers with Enhanced Scaling FPC4s

For more information, see the J Series router documentation and "Tricolor Marking Policer Overview" on page 85.

To configure CoS forwarding classes, include the **forwarding-classes** statement at the **[edit class-of-service]** hierarchy level:

```
[edit class-of-service]
forwarding-classes {
  class class-name queue-num queue-number priority (high | low);
  queue queue-number class-name priority (high | low);
}
forwarding-classes-interface-specific forwarding-class-map-name {
  class class-name queue-num queue-number [ restricted-queue queue-number ];
}
interfaces {
  interface-name {
    unit logical-unit-number {
      forwarding-class class-name;
      forwarding-classes-interface-specific forwarding-class-map-name;
    }
  }
}
restricted-queues {
  forwarding-class class-name queue queue-number;
}
```

Default Forwarding Classes

By default, four queues are assigned to four forwarding classes, each with a queue number, name, and abbreviation.

These default mappings apply to all routers. The four forwarding classes defined by default are shown in Table 20 on page 113.

If desired, you can rename the forwarding classes associated with the queues supported on your hardware. Assigning a new class name to an output queue does not alter the default classification or scheduling that is applicable to that queue. CoS configurations can be quite complicated, so unless it is required by your scenario, we recommend that you not alter the default class names or queue number associations.

Some routers support eight queues. Queues 4 through 7 have no default mappings to forwarding classes. To use queues 4 through 7, you must create custom forwarding class names and map them to the queues. For more information, see the Juniper Networks J Series Services Router documentation.

Table 20: Default Forwarding Classes

Queue	Forwarding Class Name	Comments
Queue 0	**best-effort** (**be**)	The software 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 software delivers assured bandwidth, low loss, low delay, and low delay variation (jitter) end-to-end for packets in this service class. Routers 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 software offers a high level of assurance that the packets are delivered as long as the packet flow from the customer stays within a certain service profile that you define. The software accepts excess traffic, but applies a RED drop profile to determine if the excess packets are dropped and not forwarded. Depending on router type, up to four drop probabilities (low, medium-low, medium-high, and high) are defined for this service class.
Queue 3	**network-control** (**nc**)	The software 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.

The following rules govern queue assignment:

- If classifiers fail to classify a packet, the packet always receives the default classification to the class associated with queue 0.
- The number of queues is dependent on the hardware plugged into the chassis. CoS configurations are inherently contingent on the number of queues on the system. Only two classes, **best-effort** and **network-control**, are referenced in the default configuration. The default configuration works on all routers.

- CoS configurations that specify more queues than the router can support are not accepted. The commit fails with a detailed message that states the total number of queues available.
- All default CoS configuration is based on queue number. The name of the forwarding class that shows up when the default configuration is displayed is the forwarding class currently associated with that queue.

This is the default configuration for the **forwarding-classes** statement:

```
[edit class-of-service]
forwarding-classes {
   queue 0 best-effort;
   queue 1 expedited-forwarding;
   queue 2 assured-forwarding;
   queue 3 network-control;
}
```

If you reassign the forwarding-class names, the **best-effort** forwarding-class name appears in the locations in the configuration previously occupied by **network-control** as follows:

```
[edit class-of-service]
forwarding-classes {
   queue 0 network-control;
   queue 1 assured-forwarding;
   queue 2 expedited-forwarding;
   queue 3 best-effort;
}
```

All the default rules of classification and scheduling that applied to Queue 3 still apply. Queue 3 is simply now renamed **best-effort**.

On Juniper Networks M320 Multiservice Edge Routers and T Series Core Routers, you can assign multiple forwarding classes to a single queue. If you do so, the first forwarding class that you assign to queue 0 acquires the default BE classification and scheduling. The first forwarding class that you assign to queue 1 acquires the default EF classification and scheduling. The first forwarding class that you assign to queue 2 acquires the default AF classification and scheduling. The first forwarding class that you assign to queue 3 acquires the default NC classification and scheduling. For more information, see “Configuring Up to 16 Forwarding Classes” on page 120.

- In the current default configuration:
  - Only IP precedence classifiers are associated with interfaces.
  - The only classes designated are **best-effort** and **network-control**.
  - Schedulers are not defined for the **expedited-forwarding** or **assured-forwarding** forwarding classes.
- You must explicitly classify packets to the **expedited-forwarding** or **assured-forwarding** forwarding class and define schedulers for these classes.

- For Asynchronous Transfer Mode (ATM) interfaces on Juniper Networks M Series Multiservice Edge Routers, when you use fixed classification with multiple logical interfaces classifying to separate queues, a logical interface without a classifier attached inherits the most recent classifier applied on a different logical interface. For example, suppose you configure traffic through logical unit 0 to be classified into queue 1, and you configure traffic through logical unit 1 to be classified into queue 3. You want traffic through logical unit 2 to be classified into the default classifier, which is queue 0. In this case, traffic through logical unit 2 is classified into queue 3, because the configuration of logical unit 1 was committed last.

For more information, see “Hardware Capabilities and Limitations” on page 259.

Configuring Forwarding Classes

You assign each forwarding class to an internal queue number by including the **forwarding-classes** statement at the **[edit class-of-service]** hierarchy level:

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

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

CAUTION: We do not recommend classifying packets into a forwarding class that has no associated scheduler on the egress interface. Such a configuration can cause unnecessary packet drops because an unconfigured scheduling class might lack adequate buffer space. For example, if you configure a custom scheduler map that does not define queue 0, and the default classifier assigns incoming packets to the best-effort class (queue 0), the unconfigured egress queue for the best-effort forwarding class might not have enough space to accommodate even short packet bursts.

A default congestion and transmission control mechanism is used when an output interface is not configured for a certain forwarding class, but receives packets destined for that unconfigured forwarding class. This default mechanism uses the delay buffer and weighted round robin (WRR) credit allocated to the designated forwarding class, with a default drop profile. Because the buffer and WRR credit allocation is minimal, packets might be lost if a larger number of packets are forwarded without configuring the forwarding class for the interface.

Applying Forwarding Classes to Interfaces

You can configure *fixed classification* on a logical interface by specifying a forwarding class to be applied to all packets received by the logical interface, regardless of the packet contents.

To apply a forwarding class configuration to the input logical interface, include the **forwarding-class** 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]
forwarding-class class-name;
```

You can include interface wildcards for ***interface-name*** and ***logical-unit-number***.

In the following example, all packets coming into the router from the **ge-3/0/0.0** interface are assigned to the **assured-forwarding** forwarding class:

```
[edit class-of-service]
interfaces {
  ge-3/0/0 {
    unit 0 {
      forwarding-class assured-forwarding;
    }
  }
}
```

Classifying Packets by Egress Interface

For Juniper Networks M320 Multiservice Edge Routers and T Series Core Routers with the Intelligent Queuing (IQ), IQ2, Enhanced IQ (IQE), Multiservices link services intelligent queuing (LSQ) interfaces, or ATM2 PICs, you can classify unicast and multicast packets based on the egress interface. For unicast traffic, you can also use a multifield filter, but only egress interface classification applies to multicast traffic as well as unicast. If you configure egress classification of an interface, you cannot perform Differentiated Services code point (DSCP) rewrites on the interface. By default, the system will not perform any classification based on egress interface.

To enable packet classification by egress interface, you first configure a forwarding class map and one or more queue numbers for the egress interface at the **[edit class-of-service forwarding-classes-interface-specific *forwarding-class-map-name*]** hierarchy level:

```
[edit class-of-service]
forwarding-classes-interface-specific  forwarding-class-map-name {
  class class-name queue-num queue-number [ restricted-queue queue-number ];
}
```

For T Series routers that are restricted to only four queues, you can control the queue assignment with the **restricted-queue** option, or you can allow the system to automatically determine the queue in a modular fashion. For example, a map assigning packets to queue 6 would map to queue 2 on a four-queue system.

NOTE: If you configure an output forwarding class map associating a forwarding class with a queue number, these maps are not supported on multiservices link services intelligent queuing (lsq-) interfaces.

Once the forwarding class map has been configured, you apply the map to the logical interface using the **output-forwarding-class-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]
output-forwarding-class-map forwarding-class-map-name;
```

All parameters relating to the queues and forwarding class must be configured as well. For more information about configuring forwarding classes and queues, see “Configuring Forwarding Classes” on page 115.

This example configures an interface-specific forwarding-class map named **FCMAP1** that restricts queues 5 and 6 to different queues on four-queue systems and then applies **FCMAP1** to **unit 0** of interface **ge-6/0/0**:

```
[edit class-of-service]
forwarding-classes-interface-specific FCMAP1 {
   class FC1 queue-num 6 restricted-queue 3;
   class FC2 queue-num 5 restricted-queue 2;
   class FC3 queue-num 3;
   class FC4 queue-num 0;
   class FC3 queue-num 0;
   class FC4 queue-num 1;
}

[edit class-of-service]
interfaces {
   ge-6/0/0 unit 0 {
      output-forwarding-class-map FCMAP1;
   }
}
```

Note that without the **restricted-queue** option in **FCMAP1**, the example would assign **FC1** and **FC2** to queues 2 and 1, respectively, on a system restricted to four queues.

Use the **show class-of-service forwarding-class *forwarding-class-map-name*** command to display the forwarding-class map queue configuration:

```
user@host> show class-of-service forwarding-class FCMAP2

Forwarding class                       ID      Queue  Restricted queue
  FC1                                   0        6           3
  FC2                                   1        5           2
  FC3                                   2        3           3
  FC4                                   3        0           0
  FC5                                   4        0           0
  FC6                                   5        1           1
  FC7                                   6        6           2
  FC8                                   7        7           3
```

Use the **show class-of-service interface *interface-name*** command to display the forwarding-class maps (and other information) assigned to a logical interface:

```
user@host> show class-of-service interface ge-6/0/0

Physical interface: ge-6/0/0, Index: 128
Queues supported: 8, Queues in use: 8
```



```
 Scheduler map: <default>, Index: 2
 Input scheduler map: <default>, Index: 3
 Chassis scheduler map: <default-chassis>, Index: 4

Logical interface: ge-6/0/0.0, Index: 67
 Object                      Name              Type                     Index
 Scheduler-map               sch-map1          Output                    6998
 Scheduler-map               sch-map1          Input                     6998
 Classifier                  dot1p             ieee8021p                 4906
 forwarding-class-map        FCMAP1            Output                    1221

Logical interface: ge-6/0/0.1, Index 68
 Object                      Name              Type                     Index
 Scheduler-map               <default>         Output                       2
 Scheduler-map               <default>         Input                        3

Logical interface: ge-6/0/0.32767, Index 69
 Object                      Name              Type                     Index
 Scheduler-map               <default>         Output                       2
 Scheduler-map               <default>         Input                        3
```

Example: DSCP IPv6 Rewrites and Forwarding Class Maps

You cannot configure a DSCP IPv6 rewrite rule and output forwarding class map on the same logical interface (unit). These must be used on different logical interfaces. Although a warning is issued, there is nothing in the CLI that prevents this configuration. An error message appears when you attempt to commit the configuration.

This example shows the warning and error message that results when the default DSCP IPv6 rewrite rule is configured on logical interface **ge-1/0/4.0** with output forwarding class map **vg1**.

```
[edit class-of-service]
interfaces {
  ge-1/0/4 {
    unit 0 {
      ##
      ## Warning: DSCP-IPv6 rewrite and forwarding class map not allowed on same unit
      ##
      output-forwarding-class-map vg1;
      rewrite-rules {
        dscp-ipv6 default;
      }
    }
  }
}
```

```
user@router# commit
[edit class-of-service interfaces ge-1/0/4 unit 0 output-forwarding-class-map]
  'output-forwarding-class-map vg1'
    DSCP-IPv6 rewrite and forwarding class map not allowed on same unit
error: commit failed: (statements constraint check failed)
```

Related Documentation

- Applying Forwarding Classes to Interfaces on page 115

Assigning Forwarding Class and DSCP Value for Routing Engine-Generated Traffic

You can set the forwarding class and differentiated service code point (DSCP) value for traffic originating in the Routing Engine. To configure forwarding class and DSCP values that apply to Routing Engine–generated traffic only, apply an output filter to the loopback (**lo.0**) interface and set the appropriate forwarding class and DSCP bit configuration for various protocols. For example, you can set the DSCP value on OSPF packets that originate in the Routing Engine to **10** and assign them to the AF (assured forwarding) forwarding class while the DSCP value on ping packets are set to **0** and use forwarding class BE (best effort).

This particular classification ability applies to packets generated by the Routing Engine only.

The following example assigns Routing Engine sourced ping packets (using ICMP) a DSCP value of **38** and a forwarding class of **af17**, OSPF packets a DSCP value of **12** and a forwarding class of **af11**, and BGP packets (using TCP) a DSCP value of **10** and a forwarding class of **af16**.

```
[edit class-of-service]
forwarding-classes {
  class af11 queue-num 7;
  class af12 queue-num 1;
  class af13 queue-num 2;
  class af14 queue-num 4;
  class af15 queue-num 5;
  class af16 queue-num 4;
  class af17 queue-num 6;
  class af18 queue-num 7;
}

[edit firewall filter family inet]
filter loopback-filter {
  term t1 {
    from {
      protocol icmp; # For pings
    }
    then {
      forwarding-class af17;
      dscp 38;
    }
  }
  term t2 {
    from {
      protocol ospf; # For OSPF
    }
    then {
      forwarding-class af11;
      dscp 12;
    }
  }
  term t3 {
    from {
```



```
            protocol tcp; # For BGP
        }
        then {
            forwarding-class af16;
            dscp 10;
        }
    }
    term t4 {
        then accept; # Do not forget!
    }
}

[edit interfaces]
lo0 {
    unit 0 {
        family inet {
            filter {
                output loopback-filter;
            }
        }
    }
}
```

NOTE: This is not a complete router configuration. You still have to assign resources to the queues, configure the routing protocols, addresses, and so on.

Overriding Fabric Priority Queuing

On M320 and T Series routers, the default behavior is for fabric priority queuing on egress interfaces to match the scheduling priority you assign. High-priority egress traffic is automatically assigned to high-priority fabric queues. Likewise, low-priority egress traffic is automatically assigned to low-priority fabric queues.

You can override the default fabric priority queuing of egress traffic by including the **priority** statement at the **[edit class-of-service forwarding-classes queue *queue-number* *class-name*]** hierarchy level:

[edit class-of-service **forwarding-classes queue** *queue-number class-name*]
priority (high | low);

For information about associating a scheduler with a fabric priority, see “Associating Schedulers with Fabric Priorities” on page 200.

Configuring Up to 16 Forwarding Classes

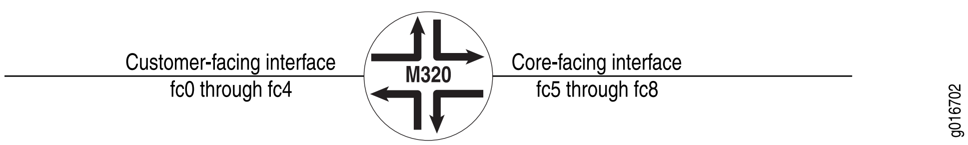
By default on all routers, four output queues are mapped to four forwarding classes, as shown in the topic “Default Forwarding Classes” on page 112. On Juniper Networks J Series Services Routers, M120 and M320 Multiservice Edge Routers, and T Series Core Routers, you can configure more than four forwarding classes and queues. For information about configuring J Series routers, see the J Series router documentation.

NOTE: You cannot use CoS-based forwarding features if you configure more than eight queues on the device.

On M120, M320, MX Series, and T Series routers, you can configure up to 16 forwarding classes and eight queues, with multiple forwarding classes assigned to single queues. The concept of assigning multiple forwarding classes to a queue is sometimes referred to as creating *forwarding-class aliases*. This section explains how to configure M320 and T Series routers.

Mapping multiple forwarding classes to single queues is useful. Suppose, for example, that forwarding classes are set based on multifield packet classification, and the multifield classifiers are different for core-facing interfaces and customer-facing interfaces. Suppose you need four queues for a core-facing interface and five queues for a customer-facing interface, where **fc0** through **fc4** correspond to the classifiers for the customer-facing interface, and **fc5** through **fc8** correspond to classifiers for the core-facing interface, as shown in Figure 11 on page 121.

Figure 11: Customer-Facing and Core-Facing Forwarding Classes



In this example, there are nine classifiers and, therefore, nine forwarding classes. The forwarding class-to-queue mapping is shown in Table 21 on page 121.

Table 21: Sample Forwarding Class-to-Queue Mapping

Forwarding Class Names	Queue Number
fc0 **fc5**	0
fc1 **fc6**	1
fc2 **fc7**	2
fc3 **fc8**	3
fc4	4

To configure up to 16 forwarding classes, include the **class** and **queue-num** statements at the **[edit class-of-service forwarding-classes]** hierarchy level:

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

You can configure up 16 different forwarding-class names. The corresponding output queue number can be from 0 through 7. Therefore, you can map multiple forwarding classes to a single queue. If you map multiple forwarding classes to a queue, the multiple forwarding classes must refer to the same scheduler (at the **[edit class-of-service scheduler-maps** *map-name* **forwarding-class** *class-name* **scheduler** *scheduler-name***]** hierarchy level).

When you configure up to 16 forwarding classes, you can use them as you can any other forwarding class—in classifiers, schedulers, firewall filters (multifield classifiers), policers, and rewrite rules.

When you configure up to 16 forwarding classes, the following limitations apply:

- The **class** and **queue** statements at **[edit class-of-service forwarding-classes]** hierarchy level are mutually exclusive. In other words, you can include one or the other of the following configurations, but not both:

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

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

- When you configure IEEE 802.1p rewrite marking on Gigabit Ethernet IQ and Gigabit Ethernet IQ2 PICs, you cannot configure more than eight forwarding classes.
- For GRE and IP-IP tunnels, IP precedence and DSCP rewrite marking of the inner header do not work with more than eight forwarding classes.
- When you use CoS-based forwarding features, you cannot configure more than eight queues.
- A scheduler map that maps eight different forwarding classes to eight different schedulers can only be applied to interfaces that support eight queues. If you apply this type of scheduler map to an interface that only supports four queues, then the commit will fail.
- We recommend that you configure the statements changing PICs to support eight queues and then applying an eight queue scheduler map in two separate steps. Otherwise, the commit might succeed but the PIC might not have eight queues when the scheduler map is applied, generating an error.
- If the ID assigned to a forwarding class is from 8 through 15 and if the incoming interface is on a Gigabit Ethernet IQ2 PIC, fixed classification does not work. Fixed classification works on Gigabit Ethernet IQ2 PICs if the forwarding class used for fixed classification has an ID from 0 through 7.

You can determine the ID number assigned to a forwarding class by issuing the **show class-of-service forwarding-class** command. You can determine whether the classification is fixed by issuing the **show class-of-service forwarding-table classifier mapping** command. In the command output, if the **Table Type** field appears as **Fixed**, the classification is fixed.

For more information about fixed classification, see "Applying Forwarding Classes to Interfaces" on page 115.

For information about configuring eight forwarding classes on ATM2 IQ interfaces, see "Enabling Eight Queues on ATM Interfaces" on page 418.

This section discusses the following topics:

Enabling Eight Queues on Interfaces

By default, Intelligent Queuing (IQ), Intelligent Queuing 2 (IQ2), Intelligent Queuing Enhanced (IQE), and Intelligent Queuing 2 Enhanced (IQ2E) PICs on M320 and T Series routers are restricted to a maximum of four egress queues per interface. To configure a maximum of eight egress queues on these interfaces, include the **max-queues-per-interface** statement at the **[edit chassis fpc *slot-number* pic *pic-number*]** hierarchy level:

[edit chassis fpc *slot-number* pic *pic-number*]
max-queues-per-interface (4 | 8);

On a TX Matrix or TX Matrix Plus router, include the **max-queues-per-interface** statement at the **[edit chassis lcc *number* fpc *slot-number* pic *pic-number*]** hierarchy level:

[edit chassis lcc *number* fpc *slot-number* pic *pic-number*]
max-queues-per-interface (4 | 8);

The numerical value can be **4** or **8**.

For Juniper Networks J Series routers, this statement is not supported. J Series routers always have eight queues available.

NOTE: In addition to configuring eight queues at the **[edit chassis]** hierarchy level, the configuration at the **[edit class-of-service]** hierarchy level must support eight queues per interface.

The maximum number of queues per IQ PIC can be **4** or **8**. If you include the **max-queues-per-interface** statement, all ports on the IQ PIC use configured mode and all interfaces on the IQ PIC have the same maximum number of queues.

To determine how many queues an interface supports, you can check the **CoS queues** output field of the **show interfaces *interface-name* extensive** command:

```
user@host> show interfaces so-1/0/0 extensive
CoS queues: 8 supported
```

If you include the **max-queues-per-interface 4** statement, you can configure all four ports and configure up to four queues per port.

For 4-port OC3c/STM1 Type I and Type II PICs on M320 and T Series routers, when you include the **max-queues-per-interface 8** statement, you can configure up to eight queues on ports 0 and 2. After you commit the configuration, the PIC goes offline and comes back online with only ports 0 and 2 operational. No interfaces can be configured on ports 1 and 3.

For Quad T3 and Quad E3 PICs, when you include the **max-queues-per-interface 8** statement, you can configure up to eight queues on ports 0 and 2. After you commit the configuration, the PIC goes offline and comes back online with only ports 0 and 2 operational. No interfaces can be configured on ports 1 and 3.

When you include the **max-queues-per-interface** statement and commit the configuration, all physical interfaces on the IQ PIC are deleted and readded. Also, the PIC is taken offline and then brought back online immediately. You do not need to take the PIC offline and online manually. You should change modes between four queues and eight queues only when there is no active traffic going to the IQ PIC.

Multiple Forwarding Classes and Default Forwarding Classes

For queues 0 through 3, if you assign multiple forwarding classes to a single queue, default forwarding class assignment works as follows:

- The first forwarding class that you assign to queue 0 acquires the default BE classification and scheduling.
- The first forwarding class that you assign to queue 1 acquires the default EF classification and scheduling.
- The first forwarding class that you assign to queue 2 acquires the default AF classification and scheduling.
- The first forwarding class that you assign to queue 3 acquires the default NC classification and scheduling.

Of course you can override the default classification and scheduling by configuring custom classifiers and schedulers.

If you do not explicitly map forwarding classes to queues 0 through 3, then the respective default classes are automatically assigned to those queues. When you are counting the 16 forwarding classes, you must include in the total any default forwarding classes automatically assigned to queues 0 through 3. As a result, you can map up to 13 forwarding classes to a single queue when the single queue is queue 0, 1, 2, or 3. You can map up to 12 forwarding classes to a single queue when the single queue is queue 4, 5, 6, or 7. In summary, there must be at least one forwarding class each (default or otherwise) assigned to queue 0 through 3, and you can assign the remaining 12 forwarding classes (16–4) to any queue.

For example, suppose you assign two forwarding classes to queue 0 and you assign no forwarding classes to queues 1 through 3. The software automatically assigns one default forwarding class each to queues 1 through 3. This means 11 forwarding classes (16–5) are available for you to assign to queues 4 through 7.

For more information about forwarding class defaults, see “Default Forwarding Classes” on page 112.

PICs Restricted to Four Queues

Some Juniper Networks T Series Core Router PICs support up to 16 forwarding classes and are restricted to 4 queues. Contact Juniper Networks customer support for a current list of T Series router PICs that are restricted to four queues. To determine how many queues an interface supports, you can check the **CoS queues** output field of the **show interfaces *interface-name* extensive** command:

```
user@host> show interfaces so-1/0/0 extensive
CoS queues: 8 supported
```

By default, for T Series router PICs that are restricted to four queues, the router overrides the global configuration based on the following formula:

Qr = Qd mod Rmax

Qr is the queue number assigned if the PIC is restricted to four queues.

Qd is the queue number that would have been mapped if this PIC were not restricted.

Rmax is the maximum number of restricted queues available. Currently, this is four.

For example, assume you map the forwarding class **ef** to queue 6. For a PIC restricted to four queues, the queue number for forwarding class **ef** is **Qr** = 6 mod 4 = 2.

To determine which queue is assigned to a forwarding class, use the **show class-of-service forwarding-class** command from the top level of the CLI. The output shows queue assignments for both global queue mappings and restricted queue mappings:

```
user@host> show class-of-service forwarding-class
Forwarding class           Queue     Restricted Queue  Fabric priority
  be                                     0           2              low
  ef                                     1           2              low
  assured-forwarding                     2           2              low
  network-control                        3           3              low
```

For T Series router PICs restricted to four queues, you can override the formula-derived queue assignment by including the **restricted-queues** statement at the **[edit class-of-service]** hierarchy level:

```
[edit class-of-service]
restricted-queues {
  forwarding-class class-name queue queue-number;
}
```

You can configure up to 16 forwarding classes. The output queue number can be from 0 through 3. Therefore, for PICs restricted to four queues, you can map multiple forwarding classes to single queues. If you map multiple forwarding classes to a queue, the multiple forwarding classes must refer to the same scheduler. The class name you configure at the **[edit class-of-service restricted-queues]** hierarchy level must be either a default forwarding class name or a forwarding class you configure at the **[edit class-of-service forwarding-classes]** hierarchy level.

Examples: Configuring Up to 16 Forwarding Classes

Configure 16 forwarding classes:

Configuring 16 Forwarding Classes

```
[edit class-of-service]
forwarding-classes {
   class fc0 queue-num 0;
   class fc1 queue-num 0;
   class fc2 queue-num 1;
   class fc3 queue-num 1;
   class fc4 queue-num 2;
   class fc5 queue-num 2;
   class fc6 queue-num 3;
   class fc7 queue-num 3;
   class fc8 queue-num 4;
   class fc9 queue-num 4;
   class fc10 queue-num 5;
   class fc11 queue-num 5;
   class fc12 queue-num 6;
   class fc13 queue-num 6;
   class fc14 queue-num 7;
   class fc15 queue-num 7;
}
```

For PICs restricted to four queues, map four forwarding classes to each queue:

Restricted Queues: Mapping Two Forwarding Classes to Each Queue

```
[edit class-of-service]
restricted-queues {
   forwarding-class fc0 queue 0;
   forwarding-class fc1 queue 0;
   forwarding-class fc2 queue 0;
   forwarding-class fc3 queue 0;
   forwarding-class fc4 queue 1;
   forwarding-class fc5 queue 1;
   forwarding-class fc6 queue 1;
   forwarding-class fc7 queue 1;
   forwarding-class fc8 queue 2;
   forwarding-class fc9 queue 2;
   forwarding-class fc10 queue 2;
   forwarding-class fc11 queue 2;
   forwarding-class fc12 queue 3;
   forwarding-class fc13 queue 3;
   forwarding-class fc14 queue 3;
   forwarding-class fc15 queue 3;
}
```

For PICs restricted to four queues, if you map multiple forwarding classes to a queue, the multiple forwarding classes must refer to the same scheduler:

Configuring a Scheduler Map Applicable to an Interface Restricted to Four Queues

```
[edit class-of-service]
scheduler-maps {
   interface-restricted {
      forwarding-class be scheduler Q0;
      forwarding-class ef scheduler Q1;
      forwarding-class ef1 scheduler Q1;
      forwarding-class ef2 scheduler Q1;
```



```
        forwarding-class af1 scheduler Q2;
        forwarding-class af scheduler Q2;
        forwarding-class nc scheduler Q3;
        forwarding-class nc1 scheduler Q3;
    }
}
[edit class-of-service]
restricted-queues {
    forwarding-class be queue 0;
    forwarding-class ef queue 1;
    forwarding-class ef1 queue 1;
    forwarding-class ef2 queue 1;
    forwarding-class af queue 2;
    forwarding-class af1 queue 2;
    forwarding-class nc queue 3;
    forwarding-class nc1 queue 3;
}
```

CHAPTER 8

Configuring Forwarding Policy Options

This topic discusses the following:

Forwarding Policy Options Overview

Class-of-service (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 might want to specify a particular interface or next hop to carry high-priority traffic while all best-effort traffic takes some other path. When a routing protocol discovers equal-cost paths, it can pick a path at random or load-balance across the paths through either hash selection or round robin. CBF allows path selection based on class.

To configure CBF properties, include the following statements at the **[edit class-of-service]** hierarchy level:

```
[edit class-of-service]
forwarding-policy {
  next-hop-map map-name {
    forwarding-class class-name {
      next-hop [ next-hop-name ];
      lsp-next-hop [ lsp-regular-expression ];
      non-lsp-next-hop;
      discard;
    }
  }
  class class-name {
    classification-override {
      forwarding-class class-name;
    }
```

```
    }
  }
```

Configuring CoS-Based Forwarding

You can apply CoS-based forwarding (CBF) only to a defined set of routes. Therefore you must configure a policy statement as in the following example:

```
[edit policy-options]
policy-statement my-cos-forwarding {
  from {
    route-filter destination-prefix match-type;
  }
  then {
    cos-next-hop-map map-name;
  }
}
```

This configuration specifies that routes matching the route filter are subject to the CoS next-hop mapping specified by ***map-name***. For more information about configuring policy statements, see the *Junos OS Policy Framework Configuration Guide*.

NOTE: On M Series routers (except the M120 and M320 routers), forwarding-class-based matching and CBF do not work as expected if the forwarding class has been set with a multifield filter on an input interface.

To specify a CoS next-hop map, include the **forwarding-policy** statement at the **[edit class-of-service]** hierarchy level:

```
[edit class-of-service]
forwarding-policy {
  next-hop-map map-name {
    forwarding-class class-name {
      next-hop [ next-hop-name ];
      lsp-next-hop [ lsp-regular-expression ];
      discard;
    }
  }
}
```

When you configure CBF with OSPF as the interior gateway protocol (IGP), you must specify the next hop as an interface name or next-hop alias, not as an IP address. This is true because OSPF adds routes with the interface as the next hop for point-to-point interfaces; the next hop does not contain the IP address. For an example configuration, see "Example: Configuring CoS-Based Forwarding" on page 133.

For Layer 3 VPNs, when you use class-based forwarding for the routes received from the far-end provider-edge (PE) router within a VRF instance, the software can match the routes based on the attributes that come with the received route only. In other words, the matching can be based on the route within RIB-in. In this case, the **route-filter** statement you include at the **[edit policy-options policy-statement my-cos-forwarding**

from] hierarchy level has no effect because the policy checks the **bgp.l3vpn.0** table, not the ***vrf*.inet.0** table.

The Junos OS applies the CoS next-hop map to the set of next hops previously defined; the next hops themselves can be located across any outgoing interfaces on the router. For example, the following configuration associates a set of forwarding classes and next-hop identifiers:

```
[edit class-of-service forwarding-policy]
next-hop-map map1 {
   forwarding-class expedited-forwarding {
      next-hop next-hop1;
      next-hop next-hop2;
   }
   forwarding-class best-effort {
      next-hop next-hop3;
      lsp-next-hop lsp-next-hop4;
   }
}
```

In this example, **next-hop *N*** is either an IP address or an egress interface for some next hop, and **lsp-next-hop4** is a regular expression corresponding to any next hop with that label. Q1 through Q*N* are a set of forwarding classes that map to the specific next hop. That is, when a packet is switched with Q1 through Q*N*, it is forwarded out the interface associated with the associated next hop.

This configuration has the following implications:

- A single forwarding class can map to multiple standard next hops or LSP next hops. This implies that load sharing is done across standard next hops or LSP next hops servicing the same class value. To make this work properly, the Junos OS creates a list of the equal-cost next hops and forwards packets according to standard load-sharing rules for that forwarding class.
- If a forwarding class configuration includes LSP next hops and standard next hops, the LSP next hops are preferred over the standard next hops. In the preceding example, if both **next-hop3** and **lsp-next-hop4** are valid next hops for a route to which **map1** is applied, the forwarding table includes entry **lsp-next-hop4** only.
- If **next-hop-map** does not specify all possible forwarding classes, the default forwarding class is selected as the default. If the default forwarding class is not specified in the next-hop map, a default is designated randomly. The default forwarding class is the class associated with queue 0.
- For LSP next hops, the Junos OS uses UNIX **regex(3)**-style regular expressions. For example, if the following labels exist: **lsp**, **lsp1**, **lsp2**, **lsp3**, the statement **lsp-next-hop lsp** matches **lsp**, **lsp1**, **lsp2**, and **lsp3**. If you do not desire this behavior, you must use the anchor characters **lsp-next-hop " ^lsp$"**, which match **lsp** only.
- The route filter does not work because the policy checks against the **bgp.l3vpn.0** table instead of the ***vrf*.inet.0** table.

The final step is to apply the route filter to routes exported to the forwarding engine. This is shown in the following example:

```
routing-options {
  forwarding-table {
    export my-cos-forwarding;
  }
}
```

This configuration instructs the routing process to insert routes to the forwarding engine matching **my-cos-forwarding** with the associated next-hop CBF rules.

The following algorithm is used when you apply a configuration to a route:

- If the route is a single next-hop route, all traffic goes to that route; that is, no CBF takes effect.
- For each next hop, associate the proper forwarding class. If a next hop appears in the route but not in the **cos-next-hop** map, it does not appear in the forwarding table entry.
- The default forwarding class is used if all forwarding classes are not specified in the next-hop map. If the default is not specified, one is chosen randomly.

Overriding the Input Classification

For IPv4 or IPv6 packets, you can override the incoming classification, assigning them to the same forwarding class based on their input interface, input precedence bits, or destination address. You do so by defining a policy class when configuring CoS properties and referencing this class when configuring a routing policy.

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. Also, if the packet loss priority (PLP) bit was set in the packet by the incoming interface, the PLP bit is cleared.

To override the input packet classification, do the following:

1. Define the policy class by including the **class** statement at the **[edit class-of-service policy]** hierarchy level:

   ```
   [edit class-of-service]
   forwarding-policy {
     class class-name {
       classification-override {
         forwarding-class class-name;
       }
     }
   }
   ```

   class-name is a name that identifies the class.

2. Associate the policy class with a routing policy by including it in a **policy-statement** statement at the **[edit policy-options]** hierarchy level. Specify the destination prefixes in the **route-filter** statement and the CoS policy class name in the **then** statement.

   ```
   [edit policy-options]
   policy-statement policy-name {
     term term-name {
   ```



```
            from {
                route-filter destination-prefix match-type <class class-name>
            }
            then class class-name;
        }
    }
```

3. Apply the policy by including the **export** statement at the **[edit routing-options]** hierarchy level:

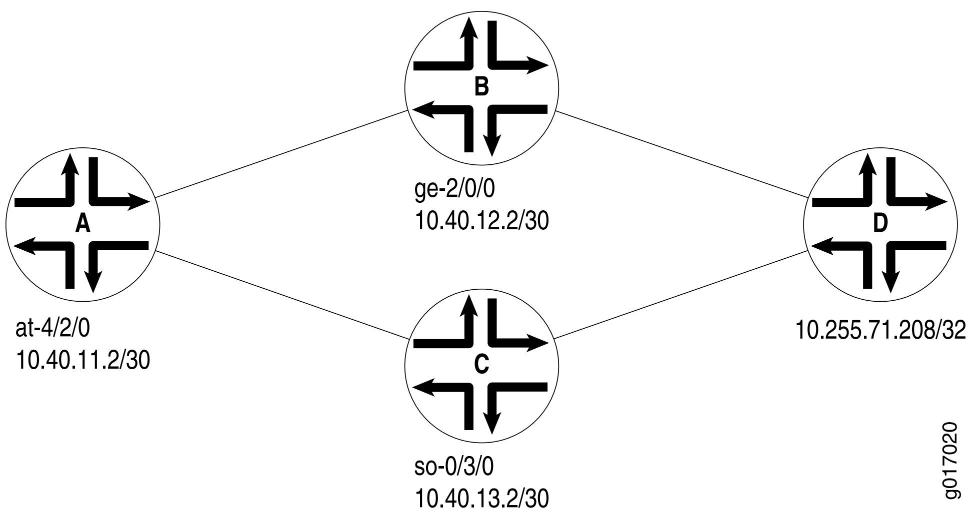
```
[edit routing-options]
forwarding-table {
    export policy-name;
}
```

Example: Configuring CoS-Based Forwarding

Router A has two routes to destination **10.255.71.208** on Router D. One route goes through Router B, and the other goes through Router C, as shown in Figure 12 on page 133.

Configure Router A with CBF to select Router B for queue 0 and queue 2, and Router C for queue 1 and queue 3.

Figure 12: Sample CoS-Based Forwarding



When you configure CBF with OSPF as the IGP, you must specify the next hop as an interface name, not as an IP address. The next hops in this example are specified as **ge-2/0/0.0** and **so-0/3/0.0**.

```
[edit class-of-service]
forwarding-policy {
    next-hop-map my_cbf {
        forwarding-class be {
            next-hop ge-2/0/0.0;
        }
        forwarding-class ef {
            next-hop so-0/3/0.0;
        }
```

```
        forwarding-class af {
            next-hop ge-2/0/0.0;
        }
        forwarding-class nc {
            next-hop so-0/3/0.0;
        }
    }
}
classifiers {
    inet-precedence inet {
        forwarding-class be {
            loss-priority low code-points [ 000 100 ];
        }
        forwarding-class ef {
            loss-priority low code-points [ 001 101 ];
        }
        forwarding-class af {
            loss-priority low code-points [ 010 110 ];
        }
        forwarding-class nc {
            loss-priority low code-points [ 011 111 ];
        }
    }
}
forwarding-classes {
    queue 0 be;
    queue 1 ef;
    queue 2 af;
    queue 3 nc;
}
interfaces {
    at-4/2/0 {
        unit 0 {
            classifiers {
                inet-precedence inet;
            }
        }
    }
}

[edit policy-options]
policy-statement cbf {
    from {
        route-filter 10.255.71.208/32 exact;
    }
    then cos-next-hop-map my_cbf;
}

[edit routing-options]
graceful-restart;
forwarding-table {
    export cbf;
}

[edit interfaces]
traceoptions {
```



```
    file trace-intf size 5m world-readable;
    flag all;
  }
  so-0/3/0 {
    unit 0 {
      family inet {
        address 10.40.13.1/30;
      }
      family iso;
      family mpls;
    }
  }
  ge-2/0/0 {
    unit 0 {
      family inet {
        address 10.40.12.1/30;
      }
      family iso;
      family mpls;
    }
  }
  at-4/2/0 {
    atm-options {
      vpi 1 {
        maximum-vcs 1200;
      }
    }
    unit 0 {
      vci 1.100;
      family inet {
        address 10.40.11.2/30;
      }
      family iso;
      family mpls;
    }
  }
```

Example: Configuring CoS-Based Forwarding for Different Traffic Types

One common use for CoS-based forwarding and next-hop maps is to enforce different handling for different traffic types, such as voice and video. For example, an LSP-based next hop can be used for voice and video, and a non-LSP next-hop can be used for best effort traffic.

Only the forwarding policy is shown in this example:

```
[edit class-of-service]
forwarding-policy {
  next-hop-map ldp-map {
    forwarding-class expedited-forwarding {
      lsp-next-hop voice;
      non-lsp-next-hop;
    }
    forwarding-class assured-forwarding {
      lsp-next-hop video;
```

```
            non-lsp-next-hop;
        }
        forwarding-class best-effort {
            non-lsp-next-hop;
            discard;
        }
    }
}
```

Example: Configuring CoS-Based Forwarding for IPv6

This example configures CoS-based forwarding (CBF) next-hop maps and CBF LSP next-hop maps for IPv6 addresses.

You can configure a next-hop map with both IPv4 and IPv6 addresses, or you can configure separate next-hop maps for IPv4 and IPv6 addresses and include the **from family (inet | inet6)** statements at the **[edit policy-options policy-options policy-statement *policy-name* term *term-name*]** hierarchy level to ensure that only next-hop maps of a specified protocol are applied to a specified route.

If you do not configure separate next-hop maps and include the **from family (inet | inet6)** statements in the configuration, when a route uses two next hops (whether IPv4, IPv6, interface, or LSP next hop) in at least two of the specified forwarding classes, CBF is used for the route; otherwise, the CBF policy is ignored.

1. Define the CBF next-hop map:

```
[edit class-of-service]
forwarding-policy {
    next-hop-map cbf-map {
        forwarding-class best-effort {
            next-hop [ ::192.168.139.38 192.168.139.38 ];
        }
        forwarding-class expedited-forwarding {
            next-hop [ ::192.168.140.5 192.168.140.5 ];
        }
        forwarding-class assured-forwarding {
            next-hop [ ::192.168.145.5 192.168.145.5 ];
        }
        forwarding-class network-control {
            next-hop [ ::192.168.141.2 192.168.141.2 ];
        }
    }
}
```

2. Define the CBF forwarding policy:

```
[edit policy-options]
policy-statement ls {
    then cos-next-hop-map cbf-map;
}
```

3. Export the CBF forwarding policy:

```
[edit routing-options]
forwarding-table {
```



```
    export ls;
}
```

CHAPTER 9

Configuring Fragmentation by Forwarding Class

This topic discusses the following:

Fragmentation by Forwarding Class Overview

For Adaptive Services (AS) Physical Interface Card (PIC) link services IQ (LSQ) and virtual LSQ redundancy (**rlsq-**) interfaces, you can specify fragmentation properties for specific forwarding classes. Traffic on each forwarding class can be either multilink fragmented or interleaved. By default, traffic in all forwarding classes is fragmented.

If you do not configure fragmentation properties for particular forwarding classes in multilink Point-to-Point Protocol (MLPPP) interfaces, the fragmentation threshold you set at the **[edit interfaces *interface-name* unit *logical-unit-number* fragment-threshold]** hierarchy level is used for all forwarding classes within the MLPPP interface. For multilink Frame Relay (MLFR) FRF.16 interfaces, the fragmentation threshold you set at the **[edit interfaces *interface-name* mlfr-uni-nni-bundle-options fragment-threshold]** hierarchy level is used for all forwarding classes within the MLFR FRF.16 interface. If you do not set a maximum fragment size anywhere in the configuration, packets are still fragmented if they exceed the smallest maximum transmission unit (MTU) of all the links in the bundle.

To configure fragmentation by forwarding class, include the following statements at the **[edit class-of-service]** hierarchy level:

```
[edit class-of-service]
fragmentation-maps {
  map-name {
    forwarding-class class-name {
      drop-timeout milliseconds;
      fragment-threshold bytes;
      multilink-class number;
```

```
            no-fragmentation;
        }
    }
}
interfaces {
    interface-name {
        unit logical-unit-number {
            fragmentation-map map-name;
        }
    }
}
```

Configuring Fragmentation by Forwarding Class

For AS PIC link services IQ (**lsq-**) interfaces only, you can configure fragmentation properties on a particular forwarding class. To do this, include the **fragmentation-maps** statement at the **[edit class-of-service]** hierarchy level:

```
[edit class-of-service]
fragmentation-maps {
    map-name {
        forwarding-class class-name {
            drop-timeout milliseconds;
            fragment-threshold bytes;
            multilink-class number;
            no-fragmentation;
        }
    }
}
```

To set a per-forwarding class fragmentation threshold, include the **fragment-threshold** statement in the fragmentation map. This statement sets the maximum size of each multilink fragment.

To set traffic on a particular forwarding class to be interleaved rather than fragmented, include the **no-fragmentation** statement in the fragmentation map. This statement specifies that an extra fragmentation header is not prepended to the packets received on this queue and that static link load balancing is used to ensure in-order packet delivery.

To change the resequencing interval for each fragmentation class, include the **drop-timeout** statement in the forwarding class. The interval is in milliseconds, and the default is 500 ms for link speeds of T1 or greater and 1500 ms for links slower than T1 speeds. You must also include a **multilink-class** value for resequencing fragments. If you include these statements, you cannot configure **no-fragmentation** for the forwarding class; they are mutually exclusive.

For a given forwarding class, include either the **fragment-threshold** or **no-fragmentation** statement; they are mutually exclusive.

Associating a Fragmentation Map with an MLPPP Interface or MLFR FRF.16 DLCI

To associate a fragmentation map with an MLPPP interface or MLFR FRF.16 DLCI, include the **fragmentation-map** statement at the **[edit class-of-service interfaces *interface-name* unit *logical-unit-number*]** hierarchy level:

```
[edit class-of-service interfaces]
lsq-fpc/pic/port {
  unit logical-unit-number { # Multilink PPP
  fragmentation-map map-name;
}
lsq-fpc/pic/port:channel { # MLFR FRF.16
unit logical-unit-number {
  fragmentation-map map-name;
}
```

For configuration examples, see the *Junos OS Services Interfaces Configuration Guide*.

Example: Configuring Fragmentation by Forwarding Class

Configure two logical units on an LSQ interface. The logical units use two different fragmentation maps.

```
class-of-service {
  interfaces {
    lsq-1/0/0 {
      unit 1 {
        fragmentation-map frag-map-A;
      }
      unit 2 {
        fragmentation-map frag-map-B;
      }
    }
  }
  fragmentation-maps {
    frag-map-A {
      forwarding-class {
        AF {
          no-fragmentation;
        }
        EF {
          no-fragmentation;
        }
        BE {
          fragment-threshold 100;
        }
      }
    }
    frag-map-B {
      forwarding-class {
        EF {
          fragment-threshold 200;
        }
        BE {
```



```
              fragment-threshold 200;
            }
            AF {
              fragment-threshold 200;
            }
          }
        }
      }
    }
```

Example: Configuring Drop Timeout Interval by Forwarding Class

For **LSQ** interfaces configured for multiclass MLPPP, you can change the drop timeout interval that the interface waits for fragment resequencing by forwarding class. This feature is mutually exclusive with the **no-fragmentation** statement configured for a forwarding class.

You can also disable the fragment resequencing function altogether by forwarding class. You do this by setting the **drop-timeout** interval to 0.

The **drop-timeout** interval can also be set at the bundle level. When the **drop-timeout** interval is set to 0 at the bundle level, *none* of the individual classes forward fragmented packets. Sequencing is ignored also, and packets are forwarded in the order in which they were received. The **drop-timeout** interval value configured at the bundle level overrides the values configured at the class level.

This example configures a logical unit on an LSQ interface with a fragmentation map setting different drop timeout values for each forwarding class:

- Best effort (BE)—The value of 0 means that no resequencing of fragments takes place for BE traffic.
- Expedited Forwarding (EF)—The value of 800 ms means that the multiclass MLPPP waits 800 ms for fragment to arrive on the link for EF traffic.
- Assured Forwarding (AF)—The absence of the timeout statements means that the default timeouts of 500 ms for links at T1 and higher speeds and 1500 ms for lower speeds are in effect for AF traffic.
- Network Control (NC)—The value of 100 ms means that the multiclass MLPPP waits 100 ms for fragment to arrive on the link for NC traffic.

```
class-of-service {
  interfaces {
    lsq-1/0/0 {
      unit 1 {
        fragmentation-map Timeout_Frag_Map;
      }
    }
  }
  fragmentation-maps {
    Timeout_Frag_Map {
      forwarding-class {
        BE {
```



```
                drop-timeout 0; # No resequencing of fragments for this class
                multilink-class 3;
                fragment-threshold 128;
            }
            EF {
                drop-timeout 800; # Timer set to 800 milliseconds for this class
                multilink-class 2;
            }
            AF {
                multilink-class 1;
                fragment-threshold 256; # Default timeout in effect for this class
            }
            NC {
                drop-timeout 100; # Timer set to 100 milliseconds for this class
                multilink-class 0;
                fragment-threshold 512;
            }
        }
    }
  }
}
```


CHAPTER 10

Configuring Schedulers

This topic discusses the following:

Schedulers Overview

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.

To configure class-of-service (CoS) schedulers, include the following statements at the **[edit class-of-service]** hierarchy level:

```
[edit class-of-service]
interfaces {
  interface-name {
    scheduler-map map-name;
    scheduler-map-chassis map-name;
    shaping-rate rate;
    unit {
      output-traffic-control-profile profile-name;
      scheduler-map map-name;
      shaping-rate rate;
    }
  }
}
fabric {
  scheduler-map {
    priority (high | low) scheduler scheduler-name;
  }
}
scheduler-maps {
  map-name {
    forwarding-class class-name scheduler scheduler-name;
  }
}
schedulers {
  scheduler-name {
    buffer-size (percent percentage | remainder | temporal microseconds );
    drop-profile-map loss-priority (any | low | medium-low | medium-high | high)protocol
      (any | non-tcp | tcp) drop-profile profile-name;
    excess-priority (low | high);
    excess-rate percent percentage;
    priority priority-level;
    transmit-rate (rate | percent percentage remainder) <exact | rate-limit>;
  }
}
traffic-control-profiles profile-name {
  delay-buffer-rate (percent percentage | rate);
  excess-rate percent percentage;
  guaranteed-rate (percent percentage | rate);
```



```
    scheduler-map map-name;
    shaping-rate (percent percentage | rate);
}
```

You cannot configure both the **shaping-rate** statement at the **[edit class-of-service interfaces *interface-name*]** hierarchy level and the **transmit-rate rate-limit** statement and option at the **[edit class-of-service schedulers *scheduler-name*]** hierarchy level. These statements are mutually exclusive. If you do configure both, you will not be able to commit the configuration:

```
[edit class-of-service]
'shaping-rate'
only one option (shaping-rate or transmit-rate rate-limit) can be configured at a time
error: commit failed (statements constraint check failed)
```

Default Schedulers

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

By default, the best effort forwarding class (queue 0) receives 95 percent of the bandwidth and buffer space for the output link, and the network control forwarding class (queue 3) receives 5 percent. The default drop profile causes the buffer to fill and then discard all packets until it has space.

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

Also 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 the offered load than the bandwidth allocated. For more information, see “Allocation of Leftover Bandwidth” on page 162.

The following default scheduler is provided when you install the Junos OS. These settings are not visible in the output of the **show class-of-service** command; rather, they are implicit.

```
[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;
        }
    }
```

Configuring Schedulers

You configure a scheduler by including the **scheduler** statement at the **[edit class-of-service]** hierarchy level:

```
schedulers {
    scheduler-name {
        buffer-size (percent percentage | remainder | temporal microseconds);
        drop-profile-map loss-priority (any | low | medium-low | medium-high | high)protocol
            (any | non-tcp | tcp) drop-profile profile-name;
        priority priority-level;
        transmit-rate (rate | percent percentage remainder) <exact | rate-limit>;
    }
}
```

For detailed information about scheduler configuration statements, see the indicated topics:

- Configuring the Scheduler Buffer Size on page 148
- Configuring Drop Profile Maps for Schedulers on page 159
- Configuring Scheduler Transmission Rate on page 160
- Configuring Schedulers for Priority Scheduling on page 165

Configuring the Scheduler Buffer Size

To control congestion at the output stage, you can configure the delay-buffer bandwidth. The delay-buffer bandwidth provides packet buffer space to absorb burst traffic up to the specified duration of delay. Once the specified delay buffer becomes full, packets with 100 percent drop probability are dropped from the head of the buffer.

The default scheduler transmission rate for queues 0 through 7 are 95, 0, 0, 5, 0, 0, 0, and 0 percent of the total available bandwidth.

The default buffer size percentages for queues 0 through 7 are 95, 0, 0, 5, 0, 0, 0, and 0 percent of the total available buffer. The total available buffer per queue differs by PIC type, as shown in Table 22 on page 149.

To configure the buffer size, include the **buffer-size** statement at the **[edit class-of-service schedulers**
***scheduler-name*]** hierarchy level:

```
[edit class-of-service schedulers scheduler-name]
buffer-size (percent percentage | remainder | temporal microseconds);
```


For each scheduler, you can configure the buffer size as one of the following:

- A percentage of the total buffer. The total buffer per queue is based on microseconds and differs by router type, as shown in Table 22 on page 149.
- The remaining buffer available. The remainder is the buffer percentage that is not assigned to other queues. For example, if you assign 40 percent of the delay buffer to queue 0, allow queue 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.
- A temporal value, in microseconds. For the temporal setting, the queuing algorithm starts dropping packets when it queues more than a computed number of bytes. This maximum is computed by multiplying the logical interface speed by the configured temporal value. The buffer size temporal value per queue differs by router type, as shown in Table 22 on page 149. The maximums apply to the logical interface, not each queue.

For information about configuring large buffer sizes on IQ PICs, see “Configuring Large Delay Buffers for Slower Interfaces” on page 150.

Table 22: Buffer Size Temporal Value Ranges by Router Type

Routers	Temporal Value Ranges
M320 and T Series router FPCs, Type 1 and Type 2	1 through 80,000 microseconds
M320 and T Series router FPCs, Type 3	1 through 50,000 microseconds
M120 router FEBs, M320 router E3-FPCs, and MX Series router nonenhanced Queuing DPCs	1 through 100,000 microseconds
M5, M7i, M10, and M10i router FPCs	1 through 100,000 microseconds
Other M Series router FPCs	1 through 200,000 microseconds
IQ PICs on all routers	1 through 100,000 microseconds
With Large Buffer Sizes Enabled	
IQ PICs on all routers	1 through 500,000 microseconds
Gigabit Ethernet IQ VLANs	
With shaping rate up to 10 Mbps	1 through 400,000 microseconds
With shaping rate up to 20 Mbps	1 through 300,000 microseconds
With shaping rate up to 30 Mbps	1 through 200,000 microseconds
With shaping rate up to 40 Mbps	1 through 150,000 microseconds

Table 22: Buffer Size Temporal Value Ranges by Router Type *(continued)*

Routers	Temporal Value Ranges
With shaping rate above 40 Mbps	1 through 100,000 microseconds

For more information about configuring delay buffers, see the following subtopics:

- Configuring Large Delay Buffers for Slower Interfaces on page 150
- Enabling and Disabling the Memory Allocation Dynamic per Queue on page 158

Configuring Large Delay Buffers for Slower Interfaces

By default, T1, E1, and *N*xDS0 interfaces and DLCIs configured on channelized IQ PICs are limited to 100,000 microseconds of delay buffer. (The default average packet size on the IQ PIC is 40 bytes.) For these interfaces, it might be necessary to configure a larger buffer size to prevent congestion and packet dropping. You can do so on the following PICs:

- Channelized IQ
- 4-port E3 IQ
- Gigabit Ethernet IQ and IQ2

Congestion and packet dropping occur when large bursts of traffic are received by slower interfaces. This happens when faster interfaces pass traffic to slower interfaces, which is often the case when edge devices receive traffic from the core of the network. For example, a 100,000-microsecond T1 delay buffer can absorb only 20 percent of a 5000-microsecond burst of traffic from an upstream OC3 interface. In this case, 80 percent of the burst traffic is dropped.

Table 23 on page 150 shows some recommended buffer sizes needed to absorb typical burst sizes from various upstream interface types.

Table 23: Recommended Delay Buffer Sizes

Length of Burst	Upstream Interface	Downstream Interface	Recommended Buffer on Downstream Interface
5000 microseconds	OC3	E1 or T1	500,000 microseconds
5000 microseconds	E1 or T1	E1 or T1	100,000 microseconds
1000 microseconds	T3	E1 or T1	100,000 microseconds

To ensure that traffic is queued and transmitted properly on E1, T1, and *N*xDS0 interfaces and DLCIs, you can configure a buffer size larger than the default maximum. To enable larger buffer sizes to be configured, include the **q-pic-large-buffer (large-scale | small-scale)** statement at the **[edit chassis fpc *slot-number* pic *pic-number*]** hierarchy level:

```
[edit chassis fpc slot-number pic pic-number]
q-pic-large-buffer large-scale;
```


If you specify the **large-scale** option, the feature supports a larger number of interfaces. If you specify **small-scale**, the default, then the feature supports a smaller number of interfaces.

When you include the **q-pic-large-buffer** statement in the configuration, the larger buffer is transparently available for allocation to scheduler queues. The larger buffer maximum varies by interface type, as shown in Table 24 on page 151.

Table 24: Maximum Delay Buffer with q-pic-large-buffer Enabled by Interface

Platform, PIC, or Interface Type	Maximum Buffer Size
With Large Buffer Sizes Not Enabled	
M320 and T Series router FPCs, Type 1 and Type 2	80,000 microseconds
M320 and T Series router FPCs, Type 3	50,000 microseconds
Other M Series router FPCs	200,000 microseconds
IQ PICs on all routers	100,000 microseconds
With Large Buffer Sizes Enabled	
Channelized T3 and channelized OC3 DLCIs—Maximum sizes vary by shaping rate:	
With shaping rate from 64,000 through 255,999 bps	4,000,000 microseconds
With shaping rate from 256,000 through 511,999 bps	2,000,000 microseconds
With shaping rate from 512,000 through 1,023,999 bps	1,000,000 microseconds
With shaping rate from 1,024,000 through 2,048,000 bps	500,000 microseconds
With shaping rate from 2,048,001 bps through 10 Mbps	400,000 microseconds
With shaping rate from 10,000,001 bps through 20 Mbps	300,000 microseconds
With shaping rate from 20,000,001 bps through 30 Mbps	200,000 microseconds
With shaping rate from 30,000,001 bps through 40 Mbps	150,000 microseconds
With shaping rate up to 40,000,001 bps and above	100,000 microseconds
*N*xDS0 IQ Interfaces—Maximum sizes vary by channel size:	
1xDSO through 3xDS0	4,000,000 microseconds
4xDSO through 7xDS0	2,000,000 microseconds
8xDSO through 15xDS0	1,000,000 microseconds

Table 24: Maximum Delay Buffer with q-pic-large-buffer Enabled by Interface *(continued)*

Platform, PIC, or Interface Type	Maximum Buffer Size
16xDSO through 32xDS0	500,000 microseconds
Other IQ interfaces	500,000 microseconds

If you configure a delay buffer larger than the new maximum, the candidate configuration can be committed successfully. However, the setting is rejected by the packet forwarding component, the default setting is used instead, and a system log warning message is generated.

For interfaces that support DLCI queuing, the large buffer is supported for DLCIs on which the configured shaping rate is less than or equal to the physical interface bandwidth. For instance, when you configure a Frame Relay DLCI on a Channelized T3 IQ PIC, and you configure the shaping rate to be 1.5 Mbps, the amount of delay buffer that can be allocated to the DLCI is 500,000 microseconds, which is equivalent to a T1 delay buffer. For more information about DLCI queuing, see “Applying Scheduler Maps and Shaping Rate to DLCIs and VLANs” on page 175.

For *N*xDS0 interfaces, the larger buffer sizes can be up to 4,000,000 microseconds, depending on the number of DS0 channels in the *N*xDS0 interface. For slower *N*xDS0 interfaces with fewer channels, the delay buffer can be relatively larger than for faster *N*xDS0 interfaces with more channels. This is shown in Table 26 on page 154. To calculate specific buffer sizes for various *N*xDS0 interfaces, see “Maximum Delay Buffer for NxDS0 Interfaces” on page 153.

You can allocate the delay buffer as either a percentage or a temporal value. The resulting delay buffer is calculated differently depending how you configure the delay buffer, as shown in Table 25 on page 152.

Table 25: Delay-Buffer Calculations

Delay Buffer Configuration	Formula	Example

Percentage	**available interface bandwidth * configured percentage buffer-size * maximum buffer = queue buffer**	If you configure a queue on a T1 interface to use 30 percent of the available delay buffer, the queue receives 28,125 bytes of delay buffer:
		`sched-expedited {`
		`  transmit-rate percent 30;`
		`  buffer-size percent 30;`
		`}`
		1.5 Mbps * 0.3 * 500,000 microseconds = 225,000 bits = 28,125 bytes

Table 25: Delay-Buffer Calculations *(continued)*

Delay Buffer Configuration	Formula	Example

Temporal	**available interface bandwidth * configured percentage transmit-rate * configured temporal buffer-size = queue buffer**	If you configure a queue on a T1 interface to use 500,000 microseconds of delay buffer and you configure the transmission rate to be 20 percent, the queue receives 18,750 bytes of delay buffer:
		sched-best {
		transmit-rate percent 20;
		buffer-size temporal 500000;
		}
		1.5 Mbps * 0.2 * 500,000 microseconds = 150,000 bits = 18,750 bytes

Percentage, with buffer size larger than transmit rate		In this example, the delay buffer is allocated twice the transmit rate. Maximum delay buffer latency can be up to twice the 500,000-microsecond delay buffer if the queue's transmit rate cannot exceed the allocated transmit rate.
		sched-extra-buffer {
		transmit-rate percent 10;
		buffer-size percent 20;
		}

FRF.16 LSQ bundles	For total bundle bandwidth < T1 bandwidth, the delay-buffer rate is 1 second. For total bundle bandwidth >= T1 bandwidth, the delay-buffer rate is 200 milliseconds (ms).	

For more information, see the following sections:

- Maximum Delay Buffer for NxDS0 Interfaces on page 153
- Example: Configuring Large Delay Buffers for Slower Interfaces on page 155

Maximum Delay Buffer for NxDS0 Interfaces

Because *N*xDS0 interfaces carry less bandwidth than a T1 or E1 interface, the buffer size on an *N*xDS0 interface can be relatively larger, depending on the number of DS0 channels combined. The maximum delay buffer size is calculated with the following formula:

```
Interface Speed * Maximum Delay Buffer Time = Delay Buffer Size
```

For example, a 1xDS0 interface has a speed of 64 kilobits per second (Kbps). At this rate, the maximum delay buffer time is 4,000,000 microseconds. Therefore, the delay buffer size is 32 kilobytes (KB):

```
64 Kbps * 4,000,000 microseconds = 32 KB
```

Table 26 on page 154 shows the delay-buffer calculations for 1xDS0 through 32xDS0 interfaces.

Table 26: NxDS0 Transmission Rates and Delay Buffers

Interface Speed	Delay Buffer Size
1xDS0 Through 4xDS0: Maximum Delay Buffer Time Is 4,000,000 Microseconds	
1xDS0: 64 Kbps	32 KB
2xDS0: 128 Kbps	64 KB
3xDS0: 192 Kbps	96 KB
4xDS0 Through 7xDS0: Maximum Delay Buffer Time Is 2,000,000 Microseconds	
4xDS0: 256 Kbps	64 KB
5xDS0: 320 Kbps	80 KB
6xDS0: 384 Kbps	96 KB
7xDS0: 448 Kbps	112 KB
8xDS0 Through 15xDS0: Maximum Delay Buffer Time Is 1,000,000 Microseconds	
8xDS0: 512 Kbps	64 KB
9xDS0: 576 Kbps	72 KB
10xDS0: 640 Kbps	80 KB
11xDS0: 704 Kbps	88 KB
12xDS0: 768 Kbps	96 KB
13xDS0: 832 Kbps	104 KB
14xDS0: 896vKbps	112 KB
15xDS0: 960 Kbps	120 KB
16xDS0 Through 32xDS0: Maximum Delay Buffer Time Is 500,000 Microseconds	
16xDS0: 1024 Kbps	64 KB
17xDS0: 1088 Kbps	68 KB
18xDS0: 1152 Kbps	72 KB
19xDS0: 1216 Kbps	76 KB
20xDS0: 1280 Kbps	80 KB
21xDS0: 1344 Kbps	84 KB

Table 26: NxDS0 Transmission Rates and Delay Buffers *(continued)*

Interface Speed	Delay Buffer Size
22xDS0: 1408 Kbps	88 KB
23xDS0: 1472 Kbps	92 KB
24xDS0: 1536 Kbps	96 KB
25xDS0: 1600 Kbps	100 KB
26xDS0: 1664 Kbps	104 KB
27xDS0: 1728 Kbps	108 KB
28xDS0: 1792 Kbps	112 KB
29xDS0: 1856 Kbps	116 KB
30xDS0: 1920 Kbps	120 KB
31xDS0: 1984 Kbps	124 KB
32xDS0: 2048 Kbps	128 KB

Example: Configuring Large Delay Buffers for Slower Interfaces

Set large delay buffers on interfaces configured on a Channelized OC12 IQ PIC. The CoS configuration binds a scheduler map to the interface specified in the chassis configuration. For information about the delay-buffer calculations in this example, see Table 25 on page 152.

```
chassis {
  fpc 0 {
    pic 0 {
      q-pic-large-buffer; # Enabling large delay buffer
      max-queues-per-interface 8; # Eight queues (M320, T Series, and TX Matrix routers)
    }
  }
}
```

Configuring the Delay Buffer Value for a Scheduler

You can assign to a physical or logical interface a scheduler map that is composed of different schedulers (or queues). The physical interface's large delay buffer can be distributed to the different schedulers (or queues) using the **transmit-rate** and **buffer-size** statements at the **[edit class-of-service schedulers *scheduler-name*]** hierarchy level.

The example shows two schedulers, **sched-best** and **sched-exped**, with the delay buffer size configured as a percentage (20 percent) and temporal value (300,000 microseconds), respectively. The **sched-best** scheduler has a transmit rate of 10 percent. The **sched-exped** scheduler has a transmit rate of 20 percent.

The **sched-best** scheduler's delay buffer is twice that of the specified transmit rate of 10 percent. Assuming that the **sched-best** scheduler is assigned to a T1 interface, this scheduler receives 20 percent of the total 500,000 microseconds of the T1 interface's delay buffer. Therefore, the scheduler receives 18,750 bytes of delay buffer:

```
available interface bandwidth * configured percentage buffer-size * maximum buffer
   = queue buffer

1.5 Mbps * 0.2 * 500,000 microseconds = 150,000 bits = 18,750 bytes
```

Assuming that the **sched-exped** scheduler is assigned to a T1 interface, this scheduler receives 300,000 microseconds of the T1 interface's 500,000-microsecond delay buffer with the traffic rate at 20 percent. Therefore, the scheduler receives 11,250 bytes of delay buffer:

```
available interface bandwidth * configured percentage transmit-rate
   * configured temporal buffer-size = queue buffer

1.5 Mbps * 0.2 * 300,000 microseconds = 90,000 bits = 11,250 bytes
```

```
[edit]
class-of-service {
   schedulers {
      sched-best {
         transmit-rate percent 10;
         buffer-size percent 20;
      }
      sched-exped {
         transmit-rate percent 20;
         buffer-size temporal 300000;
      }
   }
}
```

Configuring the Physical Interface Shaping Rate

In general, the physical interface speed is the basis for calculating the delay buffer size. However, when you include the **shaping-rate** statement, the shaping rate becomes the basis for calculating the delay buffer size. This example configures the shaping rate on a T1 interface to 200 Kbps, which means that the T1 interface bandwidth is set to 200 Kbps instead of 1.5 Mbps. Because 200 Kbps is less than 4xDS0, this interface receives 4 seconds of delay buffer, or 800 Kbps of traffic, which is 800 KB for a full second. For more information, see Table 26 on page 154.

```
class-of-service {
   interfaces {
      t1-0/0/0:1:1 {
         shaping-rate 200k;
      }
```



```
  }
}
```

Complete Configuration

This example shows a Channelized OC12 IQ PIC in FPC slot 0, PIC slot 0 and a channelized T1 interface with Frame Relay encapsulation. It also shows a scheduler map configuration on the physical interface.

```
chassis {
  fpc 0 {
    pic 0 {
      q-pic-large-buffer;
      max-queues-per-interface 8;
    }
  }
}
interfaces {
  coc12-0/0/0 {
    partition 1 oc-slice 1 interface-type coc1;
  }
  coc1-0/0/0:1 {
    partition 1 interface-type t1;
  }
  t1-0/0/0:1:1 {
    encapsulation frame-relay;
    unit 0 {
      family inet {
        address 1.1.1.1/24;
      }
      dlci 100;
    }
  }
}
class-of-service {
  interfaces {
    t1-0/0/0:1:1 {
      scheduler-map smap-1;
    }
  }
  scheduler-maps {
    smap-1 {
      forwarding-class best-effort scheduler sched-best;
      forwarding-class expedited-forwarding scheduler sched-exped;
      forwarding-class assured-forwarding scheduler sched-assure;
      forwarding-class network-control scheduler sched-network;
    }
  }
  schedulers {
    sched-best {
      transmit-rate percent 40;
      buffer-size percent 40;
    }
    sched-exped {
      transmit-rate percent 30;
      buffer-size percent 30;
    }
    sched-assure {
```

```
            transmit-rate percent 20;
            buffer-size percent 20;
        }
        sched-network {
            transmit-rate percent 10;
            buffer-size percent 10;
        }
    }
}
```

Enabling and Disabling the Memory Allocation Dynamic per Queue

In the Junos OS, the memory allocation dynamic (MAD) is a mechanism that dynamically provisions extra delay buffer when a queue is using more bandwidth than it is allocated in the transmit rate setting. With this extra buffer, queues absorb traffic bursts more easily, thus avoiding packet drops. The MAD mechanism can provision extra delay buffer only when extra transmission bandwidth is being used by a queue. This means that the queue might have packet drops if there is no surplus transmission bandwidth available.

For Juniper Networks M320 Multiservice Edge Routers, MX Services Ethernet Services Routers, and T Series Core Routers only, the MAD mechanism is enabled unless the delay buffer is configured with a temporal setting for a given queue. The MAD mechanism is particularly useful for forwarding classes carrying latency-immune traffic for which the primary requirement is maximum bandwidth utilization. In contrast, for latency-sensitive traffic, you might wish to disable the MAD mechanism because large delay buffers are not optimum.

MAD support is dependent on the FPC and Packet Forwarding Engine, not the PIC. All M320, MX Series, and T Series router FPCs and Packet Forwarding Engines support MAD. No IQ, IQ2, IQ2E or IQE PICs support MAD.

To enable the MAD mechanism on supported hardware, include the **buffer-size percent** statement at the **[edit class-of-service schedulers *scheduler-name*]** hierarchy level:

```
[edit class-of-service schedulers scheduler-name]
buffer-size percent percentage;
```

If desired, you can configure a buffer size that is greater than the configured transmission rate. The buffer can accommodate packet bursts that exceed the configured transmission rate, if sufficient excess bandwidth is available:

```
class-of-service {
    schedulers {
        sched-best {
            transmit-rate percent 20;
            buffer-size percent 30;
        }
    }
}
```

As stated previously, you can use a temporal delay buffer configuration to disable the MAD mechanism on a queue, thus limiting the size of the delay buffer. However, the effective buffer latency for a temporal queue is bounded not only by the buffer size value but also by the associated drop profile. If a drop profile specifies a drop probability of

100 percent at a fill-level less than 100 percent, the effective maximum buffer latency is smaller than the buffer size setting. This is because the drop profile specifies that the queue drop packets before the queue's delay buffer is 100 percent full.

Such a configuration might look like the following example:

```
class-of-service {
  drop-profiles {
    plp-high {
      fill-level 70 drop-probability 100;
    }
    plp-low {
      fill-level 80 drop-probability 100;
    }
  }
  schedulers {
    sched {
      buffer-size temporal 500000;
      drop-profile-map loss-priority low protocol any drop-profile plp-low;
      drop-profile-map loss-priority high protocol any drop-profile plp-high;
      transmit-rate percent 20;
    }
  }
}
```

Configuring Drop Profile Maps for Schedulers

Drop-profile maps associate drop profiles with a scheduler. The map examines the current loss priority setting of the packet (high, low, or any) and assigns a drop profile according to these values. For example, you can specify that all TCP packets with low loss priority are assigned a drop profile that you name **low-drop**. You can associate multiple drop-profile maps with a single queue.

The scheduler drop profile defines the drop probabilities across the range of delay-buffer occupancy, thereby supporting the RED process. Depending on the drop probabilities, RED might drop packets aggressively long before the buffer becomes full, or it might drop only a few packets even if the buffer is almost full. For information on how to configure drop profiles, see "RED Drop Profiles Overview" on page 231.

By default, the drop profile is mapped to packets with low PLP and any protocol type. To configure how packet types are mapped to a specified drop profile, include the **drop-profile-map** statement at the **[edit class-of-service schedulers *scheduler-name*]** hierarchy level:

```
[edit class-of-service schedulers scheduler-name ]
drop-profile-map loss-priority (any | low | medium-low | medium-high | high)protocol
  (any | non-tcp | tcp) drop-profile profile-name;
```

The map sets the drop profile for a specific PLP and protocol type. The inputs for the map are the PLP and the protocol type. The output is the drop profile. For more information about how CoS maps work, see "CoS Inputs and Outputs Overview" on page 9.

NOTE: On Juniper Network MX Series Ethernet Services Routers, you can only configure the **any** protocol option.

For each scheduler, you can configure separate drop profile maps for each loss priority (low or high).

You can configure a maximum of 32 different drop profiles.

Configuring Scheduler Transmission Rate

The transmission rate control determines the actual traffic bandwidth from 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 allows you to ensure that each queue receives the amount of bandwidth appropriate to its level of service.

On M Series routers other than the M120 and M320 routers, you should not configure a **buffer-size** larger than the **transmit-rate** for a rate-limited queue in a scheduler. If you do, the Packet Forwarding Engine will reject the CoS configuration. However, you can achieve the same effect by removing the **exact** option from the transmit rate or specifying the buffer size using the **temporal** option.

NOTE: For 8-port, 12-port, and 48-port Fast Ethernet PICs, transmission scheduling is not supported.

On Juniper Networks J Series Services Routers, you can include the **transmit-rate** statement described in this section to assign the WRR weights within a given priority level and not between priorities. For more information, see "Configuring Schedulers for Priority Scheduling" on page 165.

To configure transmission scheduling, include the **transmit-rate** statement at the **[edit class-of-service schedulers *scheduler-name*]** hierarchy level:

```
[edit class-of-service schedulers scheduler-name]
transmit-rate (rate | percent percentage | remainder) <exact | rate-limit>;
```

You can specify the transmit rate as follows:

- *rate*—Transmission rate, in bits per second. For all MX Series router interfaces, the rate can be from 65,535 through 160,000,000,000 bps. On all other platforms, the rate can be from 3200 through 160,000,000,000 bps.
- **percent *percentage***—Percentage of transmission capacity.

- **remainder**—Use remaining rate available. In the configuration, you cannot combine the **remainder** and **exact** options.
- **exact**—(Optional) Enforce the exact transmission rate or percentage you configure with the **transmit-rate** ***rate*** or **transmit-rate percent** statement. Under sustained congestion, a rate-controlled queue that goes into negative credit fills up and eventually drops packets. You specify the **exact** option as follows:

  [edit class-of-service **schedulers** *scheduler-name*]
  transmit-rate *rate* exact;

  [edit class-of-service **schedulers** *scheduler-name*]
  transmit-rate percent *percentage* exact;

  In the configuration, you cannot combine the **remainder** and **exact** options.

NOTE: Including the **exact** option is not supported on Enhanced Queuing Dense Port Concentrators (DPCs) on Juniper Network MX Series Ethernet Services Routers.

- **rate-limit**—(Optional) Limit the transmission rate to the specified amount. You can configure this option for all 8 queues of a logical interface (unit) and apply it to shaped or unshaped logical interfaces. If you configure a zero rate-limited transmit rate, all packets belonging to that queue are dropped. On IQE PICs, the **rate-limit** option for the schedulers' transmit rate is implemented as a static policer. Therefore, these schedulers are not aware of congestion and the maximum rate possible on these schedulers is limited by the value specified in the **transmit-rate** statement. Even if there is no congestion, the queue cannot send traffic above the transmit rate due to the static policer.

NOTE: You can apply a transmit rate limit to logical interfaces on Multiservices 100, 400, or 500 PICs. Typically, rate limits are used to prevent a strict-high queue (such as voice) from starving lower priority queues. You can only rate-limit one queue per logical interface. To apply a rate-limit to a Multiservices PIC interface, configure the rate limit in a scheduler and apply the scheduler map to the Multiservices (**lsq-**) interface at the **[edit class-of-service interfaces]** hierarchy level. For information about configuring other scheduler components, see "Configuring Schedulers" on page 148.

For more information about scheduler transmission rate, see the following sections:

Example: Configuring Scheduler Transmission Rate

Configure the **best-effort** scheduler to use the remainder of the bandwidth on any interface to which it is assigned:


```
class-of-service {
  schedulers {
    best-effort {
      transmit-rate remainder;
    }
  }
}
```

Allocation of Leftover Bandwidth

The allocation of leftover bandwidth is a complex topic. It is difficult to predict and to test, because the behavior of the software varies depending on the traffic mix.

If a queue receives offered loads in excess of the queue's bandwidth allocation, the queue has negative bandwidth credit, and receives a share of any available leftover bandwidth. Negative bandwidth credit means the queue has used up its allocated bandwidth. If a queue's bandwidth credit is positive, meaning it is not receiving offered loads in excess of its bandwidth configuration, then the queue does not receive a share of leftover bandwidth. If the credit is positive, then the queue does not need to use leftover bandwidth, because it can use its own allocation.

This use of leftover bandwidth is the default. If you do not want a queue to use any leftover bandwidth, you must configure it for strict allocation by including the **transmit-rate** statement with the **exact** option at the **[edit class-of-service schedulers *scheduler-name*]** hierarchy level. With rate control in place, the specified bandwidth is strictly observed. (On Juniper Networks J Series routers, the **exact** option is useful within a given priority, but not between the priorities. For more information, see "Configuring Schedulers for Priority Scheduling" on page 165.)

On J Series routers, leftover bandwidth is allocated to queues with negative credit in proportion to the configured transmit rate of the queues within a given priority level.

Juniper Networks M Series Multiservice Edge Routers and T Series Core Routers do not distribute leftover bandwidth in proportion to the configured transmit rate of the queues. Instead, the scheduler distributes the leftover bandwidth equally in round-robin fashion to queues that have negative bandwidth credit. All negative-credit queues can take the leftover bandwidth in equal share. This description suggests a simple round-robin distribution process among the queues with negative credits. In actual operation, a queue might change its bandwidth credit status from positive to negative and from negative to positive instantly while the leftover bandwidth is being distributed. Lower-rate queues tend to be allocated a larger share of leftover bandwidth, because their bandwidth credit is more likely to be negative at any given time, if they are overdriven persistently. Also, if there is a large packet size difference, (for example, queue 0 receives 64-byte packets, whereas queue 1 receives 1500-byte packets), then the actual leftover bandwidth distribution ratio can be skewed substantially, because each round-robin turn allows exactly one packet to be transmitted by a negative-credit queue, regardless of the packet size.

By default, on MX Series routers, and the M320 Enhanced Type 4 FPCs, excess bandwidth is shared in the ratio of the transmit rates. You can adjust this distribution by configuring the **excess-rate** statement at the **[edit class-of-service schedulers *scheduler-name*]** hierarchy level. You can specify the excess rate sharing by percentage or by proportion.

In summary, J Series routers distribute leftover bandwidth in proportion to the configured rates of the negative-credit queues within a given priority level. M Series and T Series routers distribute leftover bandwidth in equal shares for the queues with the same priority and same negative-credit status. MX Series routers and M320 Enhanced Type 4 FPCs, share excess bandwidth in the ratio of the transmit rates, but you can adjust this distribution.

Priority Scheduling Overview

The Junos OS supports multiple levels of transmission priority, which in order of increasing priority are **low**, **medium-low**, **medium-high**, and **high**, and **strict-high**. This allows the software to service higher-priority queues before lower-priority queues.

Priority scheduling 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. This is accomplished through a procedure in which the software examines the priority of the queue. In addition, the software determines if the individual queue is within its defined bandwidth profile. The bandwidth profile is discussed in "Configuring Scheduler Transmission Rate" on page 160. This binary decision, which is reevaluated on a regular time cycle, compares the amount of data transmitted by the queue against the amount of bandwidth allocated to it by the scheduler. When the transmitted amount is less than the allocated amount, the queue is considered to be in profile. A queue is out of profile when its transmitted amount is larger than its allocated amount.

The queues for a given output physical interface (or output logical interface if per-unit scheduling is enabled on that interface) are divided into sets based on their priority. Any such set contains queues of the same priority.

The software traverses the sets in descending order of priority. If at least one of the queues in the set has a packet to transmit, the software selects that set. A queue from the set is selected based on the weighted round robin (WRR) algorithm, which operates within the set.

The Junos OS performs priority queuing using the following steps:

1. The software locates all high-priority queues that are currently in profile. These queues are serviced first in a weighted round-robin fashion.
2. The software locates all medium-high priority queues that are currently in profile. These queues are serviced second in a weighted round-robin fashion.
3. The software locates all medium-low priority queues that are currently in profile. These queues are serviced third in a weighted round-robin fashion.
4. The software locates all low-priority queues that are currently in profile. These queues are serviced fourth in a weighted round-robin fashion.
5. The software locates all high-priority queues that are currently out of profile and are not rate limited. The weighted round-robin algorithm is applied to these queues for servicing.

6. The software locates all medium-high priority queues that are currently out of profile and are not rate limited. The weighted round-robin algorithm is applied to these queues for servicing.
7. The software locates all medium-low priority queues that are currently out of profile and are not rate limited. The weighted round-robin algorithm is applied to these queues for servicing.
8. The software locates all low-priority queues that are currently out of profile and are also not rate limited. These queues are serviced last in a weighted round-robin manner.

Platform Support for Priority Scheduling

Hardware platforms support queue priorities in different ways:

- On all platforms, you can configure one queue per interface to have strict-high priority. However, strict-high priority works differently on Juniper Networks J Series Services Routers than it does on M Series Multiservice Edge Routers and T Series Core Routers. For configuration instructions, see the J Series router documentation and "Configuring Schedulers for Priority Scheduling" on page 165.
- Strict-high priority works differently on AS PIC link services IQ (**lsq-**) interfaces. For link services IQ interfaces, a queue with strict-high priority might starve all the other queues. For more information, see the *Junos OS Services Interfaces Configuration Guide*.
- On Juniper Networks J Series Services Routers, **high** priority queues might starve **low** priority queues. For example:

  Queue priority and transmission rate:
  Queue 0: priority low, transmit-rate 50 percent
  Queue 2: priority high, transmit-rate 30 percent

  Traffic profile:
  Queue 0: 100 percent of the interface speed
  Queue 2: 100 percent of the interface speed

  Results:
  Queue 0: 0 percent of traffic is delivered.
  Queue 2: 100 percent of traffic is delivered.
- On J Series routers, you can include the **transmit-rate** statement at the **[edit class-of-service schedulers *scheduler-name*]** hierarchy level to assign the WRR weights within a given priority level and not between priorities.
- On J Series routers, adding the **exact** option with the **transmit-rate** statement is useful within a given priority and not between the priorities.
- The priority levels you configure map to hardware priority levels. These priority mappings depend on the FPC type in which the PIC is installed.

Table 27 on page 165 shows the priority mappings by FPC type. Note, for example, that on Juniper Networks M320 Multiservice Edge Routers FPCs, T Series Core Routers FPCs and T Series Enhanced FPCs, the software priorities **medium-low** and **medium-high** behave similarly because they map to the same hardware priority level.

Table 27: Scheduling Priority Mappings by FPC Type

Priority Levels	Mappings for FPCs	Mappings for M320 FPCs and T Series Enhanced FPCs	Mappings for M120 FEBs
low	0	0	0
medium-low	0	1	1
medium-high	1	1	2
high	1	2	3
strict-high (full interface bandwidth)	1	2	3

Configuring Schedulers for Priority Scheduling

To configure priority scheduling, include the **priority** statement at the **[edit class-of-service schedulers *scheduler-name*]** hierarchy level:

```
[edit class-of-service schedulers scheduler-name]
priority priority-level;
```

The priority level can be **low**, **medium-low**, **medium-high**, **high**, or **strict-high**. The priorities map to numeric priorities in the underlying hardware. In some cases, different priorities behave similarly, because two software priorities behave differently only if they map to two distinct hardware priorities. For more information, see “Platform Support for Priority Scheduling” on page 164.

Higher-priority queues transmit packets ahead of lower priority queues as long as the higher-priority forwarding classes retain enough bandwidth credit. When you configure a higher-priority queue with a significant fraction of the transmission bandwidth, the queue might lock out (or *starve*) lower priority traffic.

Strict-high priority queuing works differently on different platforms. For information about strict-high priority queuing on J Series Services Routers, see the J Series router documentation.

The following sections discuss priority scheduling:

Example: Configuring Priority Scheduling

Configure priority scheduling, as shown in the following example:

1. Configure a scheduler, **be-sched**, with **medium-low** priority.

```
[edit class-of-service]
schedulers {
  be-sched {
```

```
        priority medium-low;
    }
}
```

2. Configure a scheduler map, **be-map**, that associates **be-sched** with the **best-effort** forwarding class.

```
[edit class-of-service]
scheduler-maps {
    be-map {
        forwarding-class best-effort scheduler be-sched;
    }
}
```

3. Assign **be-map** to a Gigabit Ethernet interface, **ge-0/0/0**.

```
[edit class-of-service]
interfaces {
    ge-0/0/0 {
        scheduler-map be-map;
    }
}
```

Configuring Strict-High Priority on M Series and T Series Routers

On M Series Multiservice Edge Routers and T Series Core Routers, you can configure one queue per interface to have **strict-high** priority, which works the same as **high** priority, but provides unlimited transmission bandwidth. As long as the queue with **strict-high** priority has traffic to send, it receives precedence over all other queues, except queues with **high** priority. Queues with **strict-high** and **high** priority take turns transmitting packets until the **strict-high** queue is empty, the **high** priority queues are empty, or the **high** priority queues run out of bandwidth credit. Only when these conditions are met can lower priority queues send traffic.

When you configure a queue to have **strict-high** priority, you do not need to include the **transmit-rate** statement in the queue configuration at the **[edit class-of-service schedulers *scheduler-name*]** hierarchy level because the transmission rate of a **strict-high** priority queue is not limited by the WRR configuration. If you do configure a transmission rate on a **strict-high** priority queue, it does not affect the WRR operation. The transmission rate only serves as a placeholder in the output of commands such as the **show interface queue** command.

strict-high priority queues might starve **low** priority queues. The **high** priority allows you to protect traffic classes from being starved by traffic in a **strict-high** queue. For example, a network-control queue might require a small bandwidth allocation (say, 5 percent). You can assign **high** priority to this queue to prevent it from being underserved.

A queue with **strict-high** priority supersedes bandwidth guarantees for queues with lower priority; therefore, we recommend that you use the **strict-high** priority to ensure proper ordering of special traffic, such as voice traffic. You can preserve bandwidth guarantees for queues with lower priority by allocating to the queue with **strict-high** priority only the amount of bandwidth that it generally requires. For example, consider the following allocation of transmission bandwidth:

- Q0 BE—20 percent, low priority
- Q1 EF—30 percent, strict-high priority
- Q2 AF—40 percent, low priority
- Q3 NC—10 percent, low priority

This bandwidth allocation assumes that, in general, the EF forwarding class requires only 30 percent of an interface's transmission bandwidth. However, if short bursts of traffic are received on the EF forwarding class, 100 percent of the bandwidth is given to the EF forwarding class because of the **strict-high** setting.

Configuring Scheduler Maps

After defining a scheduler, you can associate it with a specified forwarding class by including it in a *scheduler map*. To do this, include the **scheduler-maps** statement at the **[edit class-of-service]** hierarchy level:

```
[edit class-of-service]
scheduler-maps {
  map-name {
    forwarding-class class-name scheduler scheduler-name;
  }
}
```

Applying Scheduler Maps Overview

Physical interfaces (for example, **t3-0/0/0**, **t3-0/0/0:0**, 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 have applied scheduling to one or more of the associated logical interfaces.

Logical interfaces (for example, **t3-0/0/0 unit 0** and **ge-0/0/0 unit 0**) support scheduling on data link connection identifiers (DLCIs) or VLANs only.

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

Although channelized interfaces are generally thought of as logical or virtual, the Junos OS sees T3, T1, and *N*xDS0 interfaces within a channelized IQ PIC as physical interfaces. For example, both **t3-0/0/0** and **t3-0/0/0:1** are treated as physical interfaces by the Junos OS. In contrast, **t3-0/0/0.2** and **t3-0/0/0:1.2** are considered logical interfaces because they have the **.2** at the end of the interface names.

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 t3-0/0/0 unit 0 scheduler-map map1
```

Related Documentation

To apply a scheduler map to network traffic, you associate the map with an interface. See the following topics:

- Applying Scheduler Maps to Physical Interfaces on page 168
- Applying Scheduler Maps and Shaping Rate to Physical Interfaces on IQ PICs on page 168
- Applying Scheduler Maps and Shaping Rate to DLCIs and VLANs on page 175
- Oversubscribing Interface Bandwidth on page 184
- Providing a Guaranteed Minimum Rate on page 191
- Applying Scheduler Maps to Packet Forwarding Component Queues on page 194
- Default Fabric Priority Queuing on page 200
- Associating Schedulers with Fabric Priorities on page 200

Applying Scheduler Maps to Physical Interfaces

After you have defined a scheduler map, as described in “Configuring Scheduler Maps” on page 167, you can apply it to an output interface. Include the **scheduler-map** statement at the **[edit class-of-service interfaces *interface-name*]** hierarchy level:

```
[edit class-of-service interfaces interface-name]
scheduler-map map-name;
```

Interface wildcards are supported. However, scheduler maps using wildcard interfaces are not checked against router interfaces at commit time and can result in a configuration that is incompatible with installed hardware. Fully specified interfaces, on the other hand, check the configuration against the hardware and report errors or warning if the hardware does not support the configuration.

Generally, you can associate schedulers with physical interfaces only. For some IQ interfaces, you can also associate schedulers with the logical interface. For more information, see “Applying Scheduler Maps and Shaping Rate to DLCIs and VLANs” on page 175.

NOTE: For original Channelized OC12 PICs, limited CoS functionality is supported. For more information, contact Juniper Networks customer support.

Applying Scheduler Maps and Shaping Rate to Physical Interfaces on IQ PICs

For IQ PICs, you can configure physical interfaces to shape traffic based on the rate-limited bandwidth of the total interface bandwidth. This allows you to shape the output of the physical interface, so that the interface transmits less traffic than it is physically capable of carrying.

If you do not configure a shaping rate on the physical interface, the default physical interface bandwidth is based on the channel bandwidth and the time slot allocation.

NOTE: The **shaping-rate** statement cannot be applied to a physical interface on J Series routers.

To configure shaping on the interface, include the **shaping-rate** statement at the **[edit class-of-service interfaces *interface-name*]** hierarchy level:

```
[edit class-of-service interfaces interface-name]
shaping-rate rate;
```

You can specify a peak bandwidth rate in bps, either as a complete decimal number or as a decimal number followed by the abbreviation **k** (1000), **m** (1,000,000), or **g** (1,000,000,000). For physical interfaces, the range is from 1000 through 160,000,000,000 bps. For the IQ2 Gigabit Ethernet PIC, the minimum is 80,000 bps, and for the IQ2 10 Gigabit Ethernet PIC, the minimum is 160,000 bps. (For logical interfaces, the range is 1000 through 32,000,000,000 bps.) The sum of the bandwidths you allocate to all physical interfaces on a PIC must not exceed the bandwidth of the PIC.

NOTE: For MX Series routers, the shaping rate value for the physical interface at the **[edit class-of-service interfaces *interface-name*]** hierarchy level must be a minimum of 160 Kbps.

If you configure a shaping rate that exceeds the physical interface bandwidth, the new configuration is ignored, and the previous configuration remains in effect. For example, if you configure a shaping rate that is 80 percent of the physical interface bandwidth, then change the configuration to 120 percent of the physical interface bandwidth, the 80 percent setting remains in effect. This holds true unless the PIC is restarted, in which case the default bandwidth goes into effect. As stated previously, the default bandwidth is based on the channel bandwidth and the time slot allocation.

Optionally, you can instead configure scheduling and rate shaping on logical interfaces, as described in "Applying Scheduler Maps and Shaping Rate to DLCIs and VLANs" on page 175. In general, logical and physical interface traffic shaping is mutually exclusive. You can include the **shaping-rate** statement at the **[edit class-of-service interfaces *interface-name*]** hierarchy level or the **[edit class-of-service interfaces *interface-name* unit *logical-unit-number*]** hierarchy level, but not both. For Gigabit Ethernet IQ2 and IQ2E PICs, you can configure hierarchical traffic shaping, meaning the shaping is performed on both the physical interface and the logical interface. For more information, see "Configuring Hierarchical Input Shapers" on page 341.

To view the results of your configuration, issue the following **show** commands:

- **show class-of-service interface *interface-name***
- **show interfaces *interface-name* extensive**
- **show interfaces queue**

For more information, see the following sections:

- Shaping Rate Calculations on page 170
- Examples: Applying a Scheduler Map and Shaping Rate to Physical Interfaces on page 171

Shaping Rate Calculations

For shaping rate and WRR, the information included in the calculations varies by PIC type, as shown in Table 28 on page 170.

NOTE: The 10-port 10-Gigabit Oversubscribed Ethernet (OSE) PICs and Gigabit Ethernet IQ2 PICs are unique in supporting ingress scheduling and shaping. The calculations shown for 10-port 10-Gigabit OSE and Gigabit Ethernet IQ2 PICs apply to both ingress and egress scheduling and shaping. For other PICs, the calculations apply to egress scheduling and shaping only.

For more information, see "CoS on Enhanced IQ2 PICs Overview" on page 325.

Table 28: Shaping Rate and WRR Calculations by PIC Type

PIC Type	Platform	Shaping Rate and WRR Calculations Include
10-port 10-Gigabit OSE PIC	T Series Core Routers	For ingress and egress: L3 header + L2 header + frame check sequence (FCS) + interpacket gap (IPG) + preamble
Gigabit Ethernet IQ2 PIC	All	For ingress and egress: L3 header + L2 header + frame check sequence (FCS)
Gigabit Ethernet IQ PIC	All	L3 header + L2 header + FCS
Non-IQ PIC	M320 and T Series Enhanced FPCs	L3 header + L2 header + 4-byte FCS + interpacket gap (IPG) + start-of-frame delimiter (SFD)+ preamble
	T Series non-Enhanced FPCs	L3 header
	Other M Series FPCs	L3 header+ L2 header
IQ PIC with a SONET/SDH interface	All	L3 header+ L2 header + FCS
Non-IQ PIC with a SONET/SDH interface	M320 and T Series Enhanced FPCs	L3 header +L2 header + 4-byte FCS + IPG + SFD + Preamble
	T Series non-Enhanced FPCs	L3 header
	Other M Series FPCs	L3 header+L2 header

Examples: Applying a Scheduler Map and Shaping Rate to Physical Interfaces

Applying a Shaping Rate to a Clear-Channel T1 Interface on a Channelized T1 IQ PIC

```
[edit interfaces]
ct1-2/1/0 {
  no-partition interface-type t1;
}
t1-2/1/0 {
  unit 0 {
    family inet {
      address 10.40.1.1/30;
    }
  }
}

[edit class-of-service]
interfaces {
  t1-2/1/0 {
    shaping-rate 3000;
  }
}
```

Applying a Scheduler Map and Shaping Rate to a DS0 Channel of a Channelized T1 Interface on a Channelized T1 IQ PIC

```
[edit interfaces]
ct1-0/0/9 {
  partition 1 timeslots 1-2 interface-type ds;
}
ds-0/0/9:1 {
  no-keepalives;
  unit 0 {
    family inet {
      address 10.10.1.1/30;
    }
  }
}

[edit class-of-service]
interfaces {
  ds-0/0/9:1 {
    scheduler-map sched_port_1;
    shaping-rate 2000;
  }
}
```

Applying a Shaping Rate to a Clear-Channel E1 Interface on a Channelized E1 IQ PIC

```
[edit interfaces]
ce1-2/1/0 {
  no-partition interface-type e1;
}
e1-2/1/0 {
  unit 0 {
    family inet {
      address 10.40.1.1/30;
    }
  }
}

[edit class-of-service]
```



```
interfaces {
   e1-2/1/0 {
      shaping-rate 4000;
   }
}
```

Applying a Scheduler Map and Shaping Rate to DS0 Channels of a Channelized E1 Interface on a Channelized E1 IQ PIC

```
[edit interfaces]
ce1-1/3/1 {
   partition 1 timeslots 1-4 interface-type ds;
   partition 2 timeslots 5-6 interface-type ds;
}
ds-1/3/1:1 {
   no-keepalives;
   unit 0 {
      family inet {
         address 10.10.1.1/30;
      }
   }
}
ds-1/3/1:2 {
   no-keepalives;
   unit 0 {
      family inet {
         address 10.10.1.5/30;
      }
   }
}

[edit class-of-service]
interfaces {
   ds-1/3/1:1 {
      scheduler-map sched_port_1;
      shaping-rate 1000;
   }
   ds-1/3/1:2 {
      scheduler-map sched_port_1;
      shaping-rate 1500;
   }
}
```

Applying a Scheduler Map and Shaping Rate to a Clear-Channel T3 Interface on a Channelized DS3 IQ PIC

```
[edit interfaces]
ct3-2/1/0 {
   no-partition;
}
t3-2/1/0 {
   unit 0 {
      family inet {
         address 10.40.1.1/30;
      }
   }
}

[edit class-of-service]
interfaces {
   t3-2/1/0 {
```

```
    shaping-rate 2500;
    unit 0 {
      scheduler-map sched_port_1;
    }
  }
}
```

Applying a Scheduler Map and Shaping Rate to Fractional T1 Interfaces on a Channelized DS3 IQ PIC

```
[edit interfaces]
ct3-1/1/3 {
  partition 1-3 interface-type t1;
}
t1-1/1/3:1 {
  t1-options {
    timeslots 1-2;
  }
  unit 0 {
    family inet {
      address 10.10.1.1/30;
    }
  }
}
t1-1/1/3:2 {
  t1-options {
    timeslots 3-6;
  }
  unit 0 {
    family inet {
      address 10.10.1.5/30;
    }
  }
}
t1-1/1/3:3 {
  t1-options {
    timeslots 7-12;
  }
  unit 0 {
    family inet {
      address 10.10.1.9/30;
    }
  }
}

[edit class-of-service]
interfaces {
  t1-1/1/3:1 {
    scheduler-map sched_port_1;
    shaping-rate 1200;
  }
  t1-1/1/3:2 {
    scheduler-map sched_port_1;
    shaping-rate 1300;
  }
  t1-1/1/3:3 {
    scheduler-map sched_port_1;
    shaping-rate 1400;
  }
```



```
}
```

Applying a Scheduler Map and Shaping Rate to a DS0 Channel of a T1 Interface in a Channelized T3 Interface on a Channelized DS3 IQ PIC

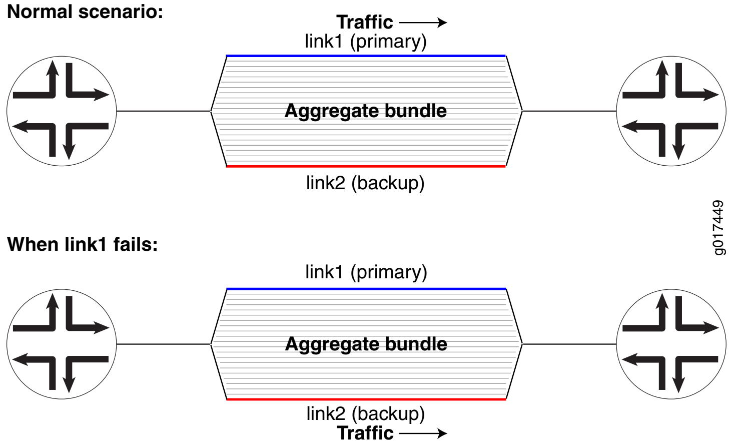
```
[edit interfaces]
ct3-2/1/3 {
  partition 1 interface-type ct1;
}
ct1-2/1/3:1 {
  partition 1 timeslots 1-4 interface-type ds;
}
ds-2/1/3:1:1 {
  unit 0 {
    family inet {
      address 10.20.144.1/30;
    }
  }
}

[edit class-of-service]
interfaces {
  ds-2/1/3:1:1 {
    scheduler-map sched_port_1;
    shaping-rate 1100;
  }
}
```

Example: Configuring VLAN Shaping on Aggregated Interfaces

Virtual LAN (VLAN) shaping (per-unit scheduling) is supported on aggregated Ethernet interfaces when link protection is enabled on the aggregated Ethernet interface. When VLAN shaping is configured on aggregate Ethernet interfaces with link protection enabled, the shaping is applied to the active child link. To configure link protection on aggregated Ethernet interfaces, include the **link-protection** statement at the **[edit interfaces ae*x* aggregated-ether-options]** hierarchy level. Traffic passes only through the designated primary link. This includes transit traffic and locally generated traffic on the router. When the primary link fails, traffic is routed through the backup link. You also can reverse traffic, from the designated backup link to the designated primary link. To revert back to sending traffic to the primary designated link when traffic is passing through the designated backup link, use the **revert** command. For example, **request interfaces revert ae0**. To configure a primary and a backup link, include the **primary** and **backup** statements at the **[edit interfaces ge-*fpc/pic/port* gigether-options 802.3ad ae*x*]** hierarchy level or the **[edit interfaces xe-*fpc/pic/port* fastether-options 802.3ad ae*x*]** hierarchy level. To disable link protection, delete the **link-protection** statement at **[edit interfaces ae*x* aggregated-ether-options link-protection]** hierarchy level. To display the active, primary, and backup link for an aggregated Ethernet interface, use the operational mode command **show interfaces redundancy ae*x***.

Figure 13 on page 175 shows how the flow of traffic changes from primary to backup when the primary link in an aggregate bundle fails.

Figure 13: Aggregated Ethernet Primary and Backup Links



This example configures two Gigabit Ethernet interfaces (**primary** and **backup**) as an aggregated Ethernet bundle (**ae0**) and enables link protection so that a shaping rate can be applied.

```
[edit class-of-service]
interface ae0 {
  shaping-rate 300m;
}
[edit interfaces]
ge-1/0/0 {
  gigether-options {
    802.3ad ae0 primary;
  }
}
ge-1/0/1 {
  gigether-options {
    802.3ad ae0 backup;
  }
}
ae0 {
  aggregated-ether-options {
    lacp {
      periodic slow;
    }
    link-protection {
      enable;
    }
  }
}
```

Applying Scheduler Maps and Shaping Rate to DLCIs 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 (DLCI) level, both transit and traffic statistics are obtained from the PFE 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 an output scheduler and shaping rate with the queues. You can configure logical interface scheduling on the following PICs:

- Adaptive Services PIC, on link services IQ (**lsq-**) interfaces
- Channelized E1 IQ PIC
- Channelized OC3 IQ PIC
- Channelized OC12 IQ PIC (Per-unit scheduling is not supported on T1 interfaces configured on this PIC.)
- Channelized STM1 IQ PIC
- Channelized T3 IQ PIC
- E3 IQ PIC
- Gigabit Ethernet IQ PIC
- Gigabit Ethernet IQ2 PIC
- IQE PICs
- Link services PIM (**ls-** interfaces) on J Series routers

For Juniper Networks J Series Services Routers only, you can configure per-unit scheduling for virtual channels. For more information, see the J Series router documentation.

For Channelized and Gigabit Ethernet IQ PICs only, you can configure a shaping rate for a VLAN or DLCI 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" on page 184.

Physical interfaces (for example, **t3-0/0/0**, **t3-0/0/0:0**, 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.

For Gigabit Ethernet IQ2 PIC PICs only, you can configure hierarchical traffic shaping, meaning the shaping is performed on both the physical interface and the logical interface. You can also configure input traffic scheduling and shared scheduling. For more information, see "CoS on Enhanced IQ2 PICs Overview" on page 325.

Logical interfaces (for example. **t3-0/0/0.0**, **ge-0/0/0.0**, and **t1-0/0/0:0.1**) support scheduling on DLCIs or VLANs only. Furthermore, logical interface scheduling is not supported on PICs that do not have IQ.

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 **ge-0/0/0.1** or **t1-0/0/0:0.1**, where the logical unit number is 1.

Although channelized interfaces are generally thought of as logical or virtual, the Junos OS sees T3, T1, and *N*xDS0 interfaces within a channelized IQ PIC as physical interfaces. For example, both **t3-0/0/0** and **t3-0/0/0:1** are treated as physical interfaces by the Junos OS. In contrast, **t3-0/0/0.2** and **t3-0/0/0:1.2** are considered logical interfaces because they have the **.2** at the end of the interface names.

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 t3-0/0/0 unit 0 scheduler-map map1
```

Table 29 on page 177 shows the interfaces that support transmission scheduling.

Table 29: Transmission Scheduling Support by Interfaces Type

Interface Type	PIC Type	Supported	Examples
IQ PICs			
Physical interfaces	ATM2 IQ	Yes	Example of supported configuration: [edit class-of-service interfaces at-0/0/0] scheduler-map map-1;
Channelized interfaces configured on IQ PICs	Channelized DS3 IQ	Yes	Example of supported configuration: [edit class-of-service interfaces t1-0/0/0:1] scheduler-map map-1;

Table 29: Transmission Scheduling Support by Interfaces Type *(continued)*

Interface Type	PIC Type	Supported	Examples
Logical interfaces (DLCIs and VLANs only) configured on IQ PICs	Gigabit Ethernet IQ with VLAN tagging enabled	Yes	Example of supported configuration: [edit class-of-service interfaces ge-0/0/0 unit 1] scheduler-map map-1;
	E3 IQ with Frame Relay encapsulation	Yes	Example of supported configuration: [edit class-of-service interfaces e3-0/0/0 unit 1] scheduler-map map-1;
	Channelized OC3 IQ with Frame Relay encapsulation	Yes	Example of supported configuration: [edit class-of-service interfaces t1-1/0/0:1:1 unit 0] scheduler-map map-1;
	Channelized STM1 IQ with Frame Relay encapsulation	Yes	Example of supported configuration: [edit class-of-service interfaces e1-0/0/0:1 unit 1] scheduler-map map-1;
	Channelized T3 IQ with Frame Relay encapsulation	Yes	Example of supported configuration: [edit class-of-service interfaces t1-0/0/0 unit 1] scheduler-map map-1;
Logical interfaces configured on IQ PICs (interfaces that are not DLCIs or VLANs)	E3 IQ PIC with Cisco HDLC encapsulation	No	Example of unsupported configuration: [edit class-of-service interfaces e3-0/0/0 unit 1] scheduler-map map-1;
	ATM2 IQ PIC with LLC/SNAP encapsulation	No	Example of unsupported configuration: [edit class-of-service interfaces at-0/0/0 unit 1] scheduler-map map-1;
	Channelized OC12 IQ PIC with PPP encapsulation	No	Example of unsupported configuration: [edit class-of-service interfaces t1-0/0/0:1 unit 1] scheduler-map map-1;
Non-IQ PICs			
Physical interfaces	T3	Yes	Example of supported configuration: [edit class-of-service interfaces t3-0/0/0] scheduler-map map-1;
Channelized OC12 PIC	Channelized OC12	Yes	Example of supported configuration: [edit class-of-service interfaces t3-0/0/0:1] scheduler-map map-1;

Table 29: Transmission Scheduling Support by Interfaces Type *(continued)*

Interface Type	PIC Type	Supported	Examples
Channelized interfaces (except the Channelized OC12 PIC)	Channelized STM1	No	Example of unsupported configuration: [edit class-of-service interfaces e1-0/0/0:1] scheduler-map map-1;
Logical interfaces	Fast Ethernet	No	Example of unsupported configuration: [edit class-of-service interfaces fe-0/0/0 unit 1] scheduler-map map-1;
	Gigabit Ethernet	No	Example of unsupported configuration: [edit class-of-service interfaces ge-0/0/0 unit 0] scheduler-map map-1;
	ATM1	No	Example of unsupported configuration: [edit class-of-service interfaces at-0/0/0 unit 2] scheduler-map map-1;
	Channelized OC12	No	Example of unsupported configuration: [edit class-of-service interfaces t3-0/0/0:0 unit 2] scheduler-map map-1;

To configure transmission scheduling on logical interfaces, perform the following steps:

1. Enable 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 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.

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*;

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*;

   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). The range is from 1000 through 32,000,000,000 bps. For the IQ2 Gigabit Ethernet PIC, the minimum is 80,000 bps, and for the IQ2 10 Gigabit Ethernet PIC, the minimum is 160,000 bps.

   For FRF.16 bundles on link services interfaces, only shaping rates based on percentage are supported.

   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 (DLCI) level, both transit and traffic statistics are obtained from the PFE and will not show any difference.

Example: Applying Scheduler Maps and Shaping Rate to DLCIs and VLANs

Associate the scheduler **sched-map-logical-0** with logical interface **unit 0** on physical interface **t3-1/0/0**, and allocate 10 Mbps of transmission bandwidth to the logical interface.

Associate the scheduler **sched-map-logical-1** with logical interface **unit 1** on physical interface **t3-1/0/0**, and allocate 20 Mbps of transmission bandwidth to the logical interface.

The allocated bandwidth is shared among the individual forwarding classes in the scheduler map. Although these schedulers are configured on a single physical interface, they are independent from each other. Traffic on one logical interface unit does not affect the transmission priority, bandwidth allocation, or drop behavior on the other logical interface unit.

For another example, see the *Junos OS Feature Guide*.

```
[edit interfaces]
t3-1/0/0:1 {
  encapsulation frame-relay;
  per-unit-scheduler;
}
```



```
[edit class-of-service]
interfaces {
   t3-1/0/0:1 {
      unit 0 {
         scheduler-map sched-map-logical-0;
         shaping-rate 10m;
      }
      unit 1 {
         scheduler-map sched-map-logical-1;
         shaping-rate 20m;
      }
   }
}
scheduler-maps {
   sched-map-logical-0 {
      forwarding-class best-effort scheduler sched-best-effort-0;
      forwarding-class assured-forwarding scheduler sched-bronze-0;
      forwarding-class expedited-forwarding scheduler sched-silver-0;
      forwarding-class network-control scheduler sched-gold-0;
   }
   sched-map-logical-1 {
      forwarding-class best-effort scheduler sched-best-effort-1;
      forwarding-class assured-forwarding scheduler sched-bronze-1;
      forwarding-class expedited-forwarding scheduler sched-silver-1;
      forwarding-class network-control scheduler sched-gold-1;
   }
}
schedulers {
   sched-best-effort-0 {
      transmit-rate 4m;
   }
   sched-bronze-0 {
      transmit-rate 3m;
   }
   sched-silver-0 {
      transmit-rate 2m;
   }
   sched-gold-0 {
      transmit-rate 1m;
   }
   sched-best-effort-1 {
      transmit-rate 8m;
   }
   sched-bronze-1 {
      transmit-rate 6m;
   }
   sched-silver-1 {
      transmit-rate 4m;
   }
   sched-gold-1 {
      transmit-rate 2m;
   }
}
```

Configuring Per-Unit Schedulers for Channelized Interfaces

You can configure per-unit scheduling on T1 and DS0 physical interfaces configured on channelized DS3 and STM1 IQ PICs. To enable per-unit scheduling, configure the **per-unit-scheduler** statements at the **[edit interfaces *interface-name*]** hierarchy level.

When per-unit scheduling is enabled on the channelized PICs, you can associate a scheduler map with the physical interface. For more information about configuring scheduler maps, see “Configuring Scheduler Maps” on page 167.

NOTE: If you configure the **per-unit-scheduler** statement on the physical interface of a 4-port channelized OC-12 IQ PIC and configure 975 logical interfaces or data link connection identifiers (DLCIs), some of the logical interfaces or DLCIs will drop all packets intermittently.

The following example configures per-unit scheduling on a channelized DS3 PIC and an STM1 IQ PIC.

```
[edit interfaces]
ct3-5/3/1 {
  partition 1 interface-type t1;
}
t1-5/3/1:1 {
  per-unit-scheduler; # This enables per-unit scheduling
  encapsulation frame-relay;
  unit 0 {
    dlci 1;
    family inet {
      address 10.0.0.2/32;
    }
  }
}
ct3-5/3/0 {
  partition 1 interface-type ct1;
}
ct1-5/3/0:1 {
  partition 1 timeslots 1 interface-type ds;
}
ds-5/3/0:1:1 {
  per-unit-scheduler; # This enables per-unit scheduling
  encapsulation frame-relay;
  unit 0 {
    dlci 1;
    family inet {
      address 10.0.0.1/32;
    }
  }
}
cau4-3/0/0 {
  partition 1 interface-type ce1;
}
```



```
cstm1-3/0/0 {
   no-partition 1 interface-type cau4;
}
ce1-3/0/0:1 {
   partition 1 timeslots 1 interface-type ds;
}
ds-3/0/0:1:1 {
   per-unit-scheduler; # This enables per-unit scheduling
   encapsulation frame-relay;
   unit 0 {
      dlci 1;
      family inet {
         address 10.1.1.1/32;
      }
   }
}

[edit class-of-service]
classifiers {
   dscp all-traffic-dscp {
      forwarding-class assured-forwarding {
         loss-priority low code-points 001010;
      }
      forwarding-class expedited-forwarding {
         loss-priority low code-points 101110;
      }
      forwarding-class best-effort {
         loss-priority low code-points 101010;
      }
      forwarding-class network-control {
         loss-priority low code-points 000110;
      }
   }
}
forwarding-classes {
   queue 0 best-effort;
   queue 1 assured-forwarding;
   queue 2 expedited-forwarding;
   queue 3 network-control;
}
interfaces {
   ds-3/0/0:1:1 {
      unit 0 {
         scheduler-map schedule-mlppp;
      }
   }
   ds-5/3/0:1:1 {
      unit 0 {
         scheduler-map schedule-mlppp;
      }
   }
   t1-5/3/1:1 {
      unit 0 {
         scheduler-map schedule-mlppp;
      }
```

```
    }
  }
  scheduler-maps {
    schedule-mlppp {
      forwarding-class expedited-forwarding scheduler expedited-forwarding;
      forwarding-class assured-forwarding scheduler assured-forwarding;
      forwarding-class best-effort scheduler best-effort;
      forwarding-class network-control scheduler network-control;
    }
  }
  schedulers {
    best-effort {
      transmit-rate percent 2;
      buffer-size percent 5;
      priority low;
    }
    assured-forwarding {
      transmit-rate percent 7;
      buffer-size percent 30;
      priority low;
    }
    expedited-forwarding {
      transmit-rate percent 90 exact;
      buffer-size percent 60;
      priority high;
    }
    network-control {
      transmit-rate percent 1;
      buffer-size percent 5;
      priority strict-high;
    }
  }
```

Oversubscribing Interface Bandwidth

The term *oversubscribing interface bandwidth* means configuring shaping rates (peak information rates [PIRs]) so that their sum exceeds the interface bandwidth.

On Channelized IQ PICs, Gigabit Ethernet IQ PICs, and FRF.16 link services IQ (LSQ) interfaces on AS PICs, you can oversubscribe interface bandwidth. This means that the logical interfaces (and DLCIs within an FRF.16 bundle) can be oversubscribed when there is leftover bandwidth. The oversubscription is capped to the configured PIR. Any unused bandwidth is distributed equally among oversubscribed logical interfaces or DLCIs.

For networks that are not likely to experience congestion, oversubscribing interface bandwidth improves network utilization, thereby allowing more customers to be provisioned on a single interface. If the actual data traffic does not exceed the interface bandwidth, oversubscription allows you to sell more bandwidth than the interface can support.

We recommend avoiding oversubscription in networks that are likely to experience congestion. Be cautious not to oversubscribe a service by too much, because this can cause degradation in the performance of the routing platform during congestion. When you configure oversubscription, starvation of some output queues can occur if the actual data traffic exceeds the physical interface bandwidth. You can prevent degradation by using statistical multiplexing to ensure that the actual data traffic does not exceed the interface bandwidth.

NOTE: You cannot oversubscribe interface bandwidth when you configure traffic shaping using the method described in "Applying Scheduler Maps and Shaping Rate to DLCIs and VLANs" on page 175.

To configure oversubscription of the interface, perform the following steps:

1. Include the **shaping-rate** statement at the **[edit class-of-service traffic-control-profiles *profile-name*]** hierarchy level:

   ```
   [edit class-of-service traffic-control-profiles profile-name]
   shaping-rate (percent percentage | rate);
   ```

   On LSQ interfaces, you can configure the shaping rate as a percentage from 1 through 100.

   On IQ and IQ2 interfaces, you can configure the shaping rate as an absolute rate from 1000 through 160,000,000,000 bps.

   For all MX series interfaces, the shaping rate can be from 65,535 through 160,000,000,000 bps.

   Alternatively, you can configure a shaping rate for a logical interface and oversubscribe the physical interface by including the **shaping-rate** statement at the **[edit class-of-service interfaces *interface-name* unit *logical-unit-number*]** hierarchy level. However, with this configuration approach, you cannot independently control the delay-buffer rate, as described in Step 2.

   NOTE: For channelized and Gigabit Ethernet IQ interfaces, the **shaping-rate** and **guaranteed-rate** statements are mutually exclusive. You cannot configure some logical interfaces to use a shaping rate and others to use a guaranteed rate. This means there are no service guarantees when you configure a PIR. For these interfaces, you can configure either a PIR or a committed information rate (CIR), but not both.

   This restriction does not apply to Gigabit Ethernet IQ2 PICs or LSQ interfaces on AS PICs. For LSQ and Gigabit Ethernet IQ2 interfaces, you can configure both a PIR and a CIR on an interface. For more information about CIRs, see "Providing a Guaranteed Minimum Rate" on page 191.

   For more information about Gigabit Ethernet IQ2 PICs, see "CoS on Enhanced IQ2 PICs Overview" on page 325.

2. Optionally, you can base the delay-buffer calculation on a delay-buffer rate. To do this, include the **delay-buffer-rate** statement at the **[edit class-of-service traffic-control-profiles *profile-name*]** hierarchy level:

   ```
   [edit class-of-service traffic-control-profiles profile-name]
   ```

delay-buffer-rate (percent *percentage* | *rate*);

The delay-buffer rate overrides the shaping rate as the basis for the delay-buffer calculation. In other words, the shaping rate or scaled shaping rate is used for delay-buffer calculations only when the delay-buffer rate is not configured.

For LSQ interfaces, if you do not configure a delay-buffer rate, the guaranteed rate (CIR) is used to assign buffers. If you do not configure a guaranteed rate, the shaping rate (PIR) is used in the undersubscribed case, and the scaled shaping rate is used in the oversubscribed case.

On LSQ interfaces, you can configure the delay-buffer rate as a percentage from 1 through 100.

On IQ and IQ2 interfaces, you can configure the delay-buffer rate as an absolute rate from 1000 through 160,000,000,000 bps.

The actual delay buffer is based on the calculations described in "Configuring Large Delay Buffers for Slower Interfaces" on page 150 and "Maximum Delay Buffer for NxDS0 Interfaces" on page 153. For an example showing how the delay-buffer rates are applied, see ***"Examples: Oversubscribing Interface Bandwidth" on page 189.

Configuring large buffers on relatively slow-speed links can cause packet aging. To help prevent this problem, the software requires that the sum of the delay-buffer rates be less than or equal to the port speed.

This restriction does not eliminate the possibility of packet aging, so you should be cautious when using the **delay-buffer-rate** statement. Though some amount of extra buffering might be desirable for burst absorption, delay-buffer rates should not far exceed the service rate of the logical interface.

If you configure delay-buffer rates so that the sum exceeds the port speed, the configured delay-buffer rate is not implemented for the last logical interface that you configure. Instead, that logical interface receives a delay-buffer rate of zero, and a warning message is displayed in the CLI. If bandwidth becomes available (because another logical interface is deleted or deactivated, or the port speed is increased), the configured delay-buffer-rate is reevaluated and implemented if possible.

If you do not configure a delay-buffer rate or a guaranteed rate, the logical interface receives a delay-buffer rate in proportion to the shaping rate and the remaining delay-buffer rate available. In other words, the delay-buffer rate for each logical interface with no configured delay-buffer rate is equal to:

(remaining delay-buffer rate * shaping rate) / (sum of shaping rates)

where the remaining delay-buffer rate is equal to:

(interface speed) - (sum of configured delay-buffer rates)

3. To assign a scheduler map to the logical interface, include the **scheduler-map** statement at the **[edit class-of-service traffic-control-profiles *profile-name*]** hierarchy level:

   [edit class-of-service traffic-control-profiles *profile-name*]
   scheduler-map *map-name*;

For information about configuring schedulers and scheduler maps, see “Configuring Schedulers” on page 148 and “Configuring Scheduler Maps” on page 167.

4. Optionally, you can enable large buffer sizes to be configured. To do this, include the **q-pic-large-buffer** statement at the **[edit chassis fpc *slot-number* pic *pic-number*]** hierarchy level:

   [edit chassis fpc *slot-number* pic *pic-number*]
   q-pic-large-buffer;

   If you do not include this statement, the delay-buffer size is more restricted. We recommend restricted buffers for delay-sensitive traffic, such as voice traffic. For more information, see “Configuring Large Delay Buffers for Slower Interfaces” on page 150.

5. To enable scheduling on logical interfaces, include the **per-unit-scheduler** statement at the **[edit interfaces *interface-name*]** hierarchy level:

   [edit interfaces *interface-name*]
   per-unit-scheduler;

   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.

6. To apply the traffic-scheduling profile to the logical interface, 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 interfaces *interface-name* unit *logical-unit-number*]
   output-traffic-control-profile *profile-name*;

   You cannot include the **output-traffic-control-profile** statement in the configuration if any of the following statements are included in the logical interface configuration: **scheduler-map**, **shaping-rate**, **adaptive-shaper**, or **virtual-channel-group** (the last two are valid on Juniper Networks J Series Services Routers only).

Table 30 on page 187 shows how the bandwidth and delay buffer are allocated in various configurations.

Table 30: Bandwidth and Delay Buffer Allocations by Configuration Scenario

Configuration Scenario	Delay Buffer Allocation
You do not oversubscribe the interface. You do not configure a guaranteed rate. You do not configure a shaping rate. You do not configure a delay-buffer rate.	Logical interface receives the remaining bandwidth and receives a delay buffer in proportion to the remaining bandwidth.
You do not oversubscribe the interface. You configure a shaping rate at the **[edit class-of-service interfaces *interface-name* unit *logical-unit-number*]** hierarchy level.	For backward compatibility, the shaped logical interface receives a delay buffer based on the shaping rate. The multiplicative factor depends on whether you include the **q-pic-large-buffer** statement. For more information, see “Configuring Large Delay Buffers for Slower Interfaces” on page 150. Unshaped logical interfaces receive the remaining bandwidth and a delay buffer in proportion to the remaining bandwidth.

Table 30: Bandwidth and Delay Buffer Allocations by Configuration Scenario *(continued)*

Configuration Scenario	Delay Buffer Allocation
You oversubscribe the interface. You do not configure a guaranteed rate. You do not configure a shaping rate. You do not configure a delay-buffer rate.	Logical interface receives minimal bandwidth with no guarantees and receives a minimal delay buffer equal to four MTU-sized packets.
You oversubscribe the interface. You configure a shaping rate. You do not configure a guaranteed rate. You do not configure a delay-buffer rate.	Logical interface receives a delay buffer based on the scaled shaping rate: scaled shaping rate = (shaping-rate * [physical interface bandwidth]) / SUM (shaping-rates of all logical interfaces on the physical interface) The logical interface receives variable bandwidth, depending on how much oversubscription and statistical multiplexing is present. If the amount of oversubscription is low enough that statistical multiplexing does not make all logical interfaces active at the same time and the physical interface bandwidth is not exceeded, the logical interface receives bandwidth equal to the shaping rate. Otherwise, the logical interface receives a smaller amount of bandwidth. In either case, the logical interface bandwidth does not exceed the shaping rate.
You oversubscribe the interface. You configure a shaping rate. You configure a delay-buffer rate.	Logical interface receives a delay buffer based on the delay-buffer rate. For example, on IQ and IQ2 interfaces: delay-buffer-rate <= 10 Mbps: 400-millisecond (ms) delay buffer delay-buffer-rate <= 20 Mbps: 300-ms delay buffer delay-buffer-rate <= 30 Mbps: 200-ms delay buffer delay-buffer-rate <= 40 Mbps: 150-ms delay buffer delay-buffer-rate > 40 Mbps: 100-ms delay buffer On LSQ DLCIs, if **total bundle bandwidth < T1 bandwidth**: delay-buffer-rate = 1 second On LSQ DLCIs, if **total bundle bandwidth >= T1 bandwidth**: delay-buffer-rate = 200 ms The multiplicative factor depends on whether you include the **q-pic-large-buffer** statement. For more information, see "Configuring Large Delay Buffers for Slower Interfaces" on page 150. The logical interface receives variable bandwidth, depending on how much oversubscription and statistical multiplexing is present. If the amount of oversubscription is low enough that statistical multiplexing does not make all logical interfaces active at the same time and the physical interface bandwidth is not exceeded, the logical interface receives bandwidth equal to the shaping rate. Otherwise, the logical interface receives a smaller amount of bandwidth. In either case, the logical interface bandwidth does not exceed the shaping rate.
You oversubscribe the interface. You do not configure a shaping rate. You configure a guaranteed rate. You configure a delay-buffer rate.	Logical interface receives a delay buffer based on the delay-buffer rate.
You oversubscribe the interface. You do not configure a shaping rate. You do not configure a guaranteed rate. You configure a delay-buffer rate.	This scenario is not allowed. If you configure a delay-buffer rate, the traffic-control profile must also include either a shaping rate or a guaranteed rate.

Table 30: Bandwidth and Delay Buffer Allocations by Configuration Scenario *(continued)*

Configuration Scenario	Delay Buffer Allocation
You oversubscribe the interface. You configure a shaping rate. You configure a guaranteed rate. You do not configure a delay-buffer rate.	Logical interface receives a delay buffer based on the guaranteed rate. This configuration is valid on LSQ interfaces and Gigabit Ethernet IQ2 interfaces only. On channelized interfaces, you cannot configure both a shaping rate (PIR) and a guaranteed rate (CIR).

Verifying Configuration of Bandwidth Oversubscription

To verify your configuration, you can issue this following operational mode commands:

- **show class-of-service interfaces**
- **show class-of-service traffic-control-profile** ***profile-name***

Examples: Oversubscribing Interface Bandwidth

This section provides two examples: oversubscription of a channelized interface and oversubscription of an LSQ interface.

Oversubscribing a Channelized Interface

Two logical interface units, **0** and **1**, are shaped to rates 2 Mbps and 3 Mbps, respectively. The delay-buffer rates are 750 Kbps and 500 Kbps, respectively. The actual delay buffers allocated to each logical interface are 1 second of 750 Kbps and 2 seconds of 500 Kbps, respectively. The 1-second and 2-second values are based on the following calculations:

```
delay-buffer-rate < [16 x 64 Kbps]): 1 second of delay-buffer-rate
delay-buffer-rate < [8 x 64 Kbps]): 2 seconds of delay-buffer-rate
```

For more information about these calculations, see “Maximum Delay Buffer for NxDS0 Interfaces” on page 153.

```
chassis {
    fpc 3 {
        pic 0 {
            q-pic-large-buffer;
        }
    }
}
interfaces {
    t1-3/0/0 {
        per-unit-scheduler;
    }
}
class-of-service {
    traffic-control-profiles {
        tc-profile1 {
            shaping-rate 2m;
            delay-buffer-rate 750k; # 750 Kbps is less than 16 x 64 Kbps
            scheduler-map sched-map1;
        }
        tc-profile2 {
            shaping-rate 3m;
            delay-buffer-rate 500k; # 500 Kbps is less than 8 x 64 Kbps
```

```
            scheduler-map sched-map2;
        }
    }
    interfaces {
        t1-3/0/0 {
            unit 0 {
                output-traffic-control-profile tc-profile1;
            }
            unit 1 {
                output-traffic-control-profile tc-profile2;
            }
        }
    }
}
```

Oversubscribing an LSQ Interface

Apply a traffic-control profile to a logical interface representing a DLCI on an FRF.16 bundle:

```
interfaces {
    lsq-1/3/0:0 {
        per-unit-scheduler;
        unit 0 {
            dlci 100;
        }
        unit 1 {
            dlci 200;
        }
    }
}

class-of-service {
    traffic-control-profiles {
        tc_0 {
            shaping-rate percent 100;
            guaranteed-rate percent 60;
            delay-buffer-rate percent 80;
        }
        tc_1 {
            shaping-rate percent 80;
            guaranteed-rate percent 40;
        }
    }
    interfaces {
        lsq-1/3/0 {
            unit 0 {
                output-traffic-control-profile tc_0;
            }
            unit 1 {
                output-traffic-control-profile tc_1;
            }
        }
    }
}
```

Providing a Guaranteed Minimum Rate

On Gigabit Ethernet IQ PICs, Channelized IQ PICs, and FRF.16 LSQ interfaces on AS PICs, you can configure guaranteed bandwidth, also known as a committed information rate (CIR). This allows you to specify a guaranteed rate for each logical interface. The guaranteed rate is a minimum. If excess physical interface bandwidth is available for use, the logical interface receives more than the guaranteed rate provisioned for the interface.

You cannot provision the sum of the guaranteed rates to be more than the physical interface bandwidth, or the bundle bandwidth for LSQ interfaces. If the sum of the guaranteed rates exceeds the interface or bundle bandwidth, the commit operation does not fail, but the software automatically decreases the rates so that the sum of the guaranteed rates is equal to the available bundle bandwidth.

To configure a guaranteed minimum rate, perform the following steps:

1. Include the **guaranteed-rate** statement at the **[edit class-of-service traffic-control-profile *profile-name*]** hierarchy level:

   [edit class-of-service traffic-control-profiles *profile-name*]
   guaranteed-rate (percent *percentage* | *rate*);

   On LSQ interfaces, you can configure the guaranteed rate as a percentage from 1 through 100.

   On IQ and IQ2 interfaces, you can configure the guaranteed rate as an absolute rate from 1000 through 160,000,000,000 bps.

   NOTE: For channelized and Gigabit Ethernet IQ interfaces, the **shaping-rate** and **guaranteed-rate** statements are mutually exclusive. You cannot configure some logical interfaces to use a shaping rate and others to use a guaranteed rate. This means there are no service guarantees when you configure a PIR. For these interfaces, you can configure either a PIR or a CIR, but not both.

   This restriction does not apply to Gigabit Ethernet IQ2 PICs or LSQ interfaces on AS PICs. For LSQ and Gigabit Ethernet IQ2 interfaces, you can configure both a PIR and a CIR on an interface.

   For more information about Gigabit Ethernet IQ2 PICs, see “CoS on Enhanced IQ2 PICs Overview” on page 325.

2. Optionally, you can base the delay-buffer calculation on a delay-buffer rate. To do this, include the **delay-buffer-rate** statement **[edit class-of-service traffic-control-profiles *profile-name*]** hierarchy level:

   [edit class-of-service traffic-control-profiles *profile-name*]
   delay-buffer-rate (percent *percentage* | *rate*);

   On LSQ interfaces, you can configure the delay-buffer rate as a percentage from 1 through 100.

On IQ and IQ2 interfaces, you can configure the delay-buffer rate as an absolute rate from 1000 through 160,000,000,000 bps.

The actual delay buffer is based on the calculations described in "Configuring Large Delay Buffers for Slower Interfaces" on page 150 and "Maximum Delay Buffer for NxDS0 Interfaces" on page 153. For an example showing how the delay-buffer rates are applied, see "Example: Providing a Guaranteed Minimum Rate" on page 194.

If you do not include the **delay-buffer-rate** statement, the delay-buffer calculation is based on the guaranteed rate, the shaping rate if no guaranteed rate is configured, or the scaled shaping rate if the interface is oversubscribed.

If you do not specify a shaping rate or a guaranteed rate, the logical interface receives a minimal delay-buffer rate and minimal bandwidth equal to four MTU-sized packets.

You can configure a rate for the delay buffer that is higher than the guaranteed rate. This can be useful when the traffic flow might not require much bandwidth in general, but in some cases traffic can be bursty and therefore needs a large buffer.

Configuring large buffers on relatively slow-speed links can cause packet aging. To help prevent this problem, the software requires that the sum of the delay-buffer rates be less than or equal to the port speed. This restriction does not eliminate the possibility of packet aging, so you should be cautious when using the **delay-buffer-rate** statement. Though some amount of extra buffering might be desirable for burst absorption, delay-buffer rates should not far exceed the service rate of the logical interface.

If you configure delay-buffer rates so that the sum exceeds the port speed, the configured delay-buffer rate is not implemented for the last logical interface that you configure. Instead, that logical interface receives a delay-buffer rate of 0, and a warning message is displayed in the CLI. If bandwidth becomes available (because another logical interface is deleted or deactivated, or the port speed is increased), the configured delay-buffer-rate is reevaluated and implemented if possible.

If the guaranteed rate of a logical interface cannot be implemented, that logical interface receives a delay-buffer rate of 0, even if the configured delay-buffer rate is within the interface speed. If at a later time the guaranteed rate of the logical interface can be met, the configured delay-buffer rate is reevaluated and if the delay-buffer rate is within the remaining bandwidth, it is implemented.

If any logical interface has a configured guaranteed rate, all other logical interfaces on that port that do not have a guaranteed rate configured receive a delay-buffer rate of 0. This is because the absence of a guaranteed rate configuration corresponds to a guaranteed rate of 0 and, consequently, a delay-buffer rate of 0.

3. To assign a scheduler map to the logical interface, include the **scheduler-map** statement at the **[edit class-of-service traffic-control-profiles *profile-name*]** hierarchy level:

   [edit class-of-service traffic-control-profiles *profile-name*]
   scheduler-map *map-name*;

   For information about configuring schedulers and scheduler maps, see "Configuring Schedulers" on page 148 and "Configuring Scheduler Maps" on page 167.

4. To enable large buffer sizes to be configured, include the **q-pic-large-buffer** statement at the **[edit chassis fpc *slot-number* pic *pic-number*]** hierarchy level:

   [edit chassis fpc *slot-number* pic *pic-number*]
   q-pic-large-buffer;

   If you do not include this statement, the delay-buffer size is more restricted. For more information, see "Configuring Large Delay Buffers for Slower Interfaces" on page 150.

5. To enable scheduling on logical interfaces, include the **per-unit-scheduler** statement at the **[edit interfaces *interface-name*]** hierarchy level:

   [edit interfaces *interface-name*]
   per-unit-scheduler;

   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.

6. To apply the traffic-scheduling profile to the logical interface, 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 interfaces *interface-name* unit *logical-unit-number*]
   output-traffic-control-profile *profile-name*;

Table 31 on page 193 shows how the bandwidth and delay buffer are allocated in various configurations.

Table 31: Bandwidth and Delay Buffer Allocations by Configuration Scenario

Configuration Scenario	Delay Buffer Allocation
You do not configure a guaranteed rate. You do not configure a delay-buffer rate.	Logical interface receives minimal bandwidth with no guarantees and receives a minimal delay buffer equal to 4 MTU-sized packets.
You configure a guaranteed rate. You do not configure a delay-buffer rate.	Logical interface receives bandwidth equal to the guaranteed rate and a delay buffer based on the guaranteed rate. The multiplicative factor depends on whether you include the **q-pic-large-buffer** statement. For more information, see "Configuring Large Delay Buffers for Slower Interfaces" on page 150.
You configure a guaranteed rate. You configure a delay-buffer rate.	Logical interface receives bandwidth equal to the guaranteed rate and a delay buffer based on the delay-buffer rate. The multiplicative factor depends on whether you include the **q-pic-large-buffer** statement. For more information, see "Configuring Large Delay Buffers for Slower Interfaces" on page 150.

Verifying Configuration of Guaranteed Minimum Rate

To verify your configuration, you can issue this following operational mode commands:

- **show class-of-service interfaces**
- **show class-of-service traffic-control-profile** ***profile-name***

Example: Providing a Guaranteed Minimum Rate

Two logical interface units, **0** and **1**, are provisioned with a guaranteed minimum of 750 Kbps and 500 Kbps, respectively. For logical unit **1**, the delay buffer is based on the guaranteed rate setting. For logical unit **0**, a delay-buffer rate of 500 Kbps is specified. The actual delay buffers allocated to each logical interface are 2 seconds of 500 Kbps. The 2-second value is based on the following calculation:

```
delay-buffer-rate < [8 x 64 Kbps]): 2 seconds of delay-buffer-rate
```

For more information about this calculation, see “Maximum Delay Buffer for NxDS0 Interfaces” on page 153.

```
chassis {
  fpc 3 {
    pic 0 {
      q-pic-large-buffer;
    }
  }
}
interfaces {
  t1-3/0/1 {
    per-unit-scheduler;
  }
}
class-of-service {
  traffic-control-profiles {
    tc-profile3 {
      guaranteed-rate 750k;
      scheduler-map sched-map3;
      delay-buffer-rate 500k; # 500 Kbps is less than 8 x 64 Kbps
    }
    tc-profile4 {
      guaranteed-rate 500k; # 500 Kbps is less than 8 x 64 Kbps
      scheduler-map sched-map4;
    }
  }
  interfaces {
    t1-3/0/1 {
    unit 0 {
      output-traffic-control-profile tc-profile3;
    }
    unit 1 {
      output-traffic-control-profile tc-profile4;
    }
  }
}
```

Applying Scheduler Maps to Packet Forwarding Component Queues

On Intelligent Queuing (IQ) and Intelligent Queuing 2 (IQ2) interfaces, the traffic that is fed from the packet forwarding components into the PIC uses low packet loss priority (PLP) by default and is distributed evenly across the four chassis queues (not PIC queues), regardless of the scheduling configuration for each logical interface. This default behavior can cause traffic congestion.

The default chassis scheduler allocates resources for queue 0 through queue 3, with 25 percent of the bandwidth allocated to each queue. When you configure the chassis to use more than four queues, you must configure and apply a custom chassis scheduler to override the default. To apply a custom chassis scheduler, include the **scheduler-map-chassis** statement at the **[edit class-of-service interfaces at-*fpc/pic*/***] hierarchy level.

To control the aggregated traffic transmitted from the chassis queues into the PIC, you can configure the chassis queues to derive their scheduling configuration from the associated logical interface's. Include the **scheduler-map-chassis derived** statement at the **[edit class-of-service interfaces *type-fpc/pic/**]** hierarchy level:

```
[edit class-of-service interfaces type-fpc/pic/*]
scheduler-map-chassis derived;
```

CAUTION: If you include the **scheduler-map-chassis derived** statement in the configuration, packet loss might occur when you subsequently add or remove logical interfaces at the **[edit interfaces *interface-name*]** hierarchy level.

When fragmentation occurs on the egress interface, the first set of packet counters displayed in the output of the **show interfaces queue** command show the post-fragmentation values. The second set of packet counters (under the **Packet Forwarding Engine Chassis Queues** field) show the pre-fragmentation values. For more information about the **show interfaces queue** command, see the *Junos OS Interfaces Command Reference*.

You can include both the **scheduler-map** and the **scheduler-map-chassis derived** statements in the same interface configuration. The **scheduler-map** statement controls the scheduler inside the PIC, while the **scheduler-map-chassis derived** statement controls the aggregated traffic transmitted into the entire PIC. For the Gigabit Ethernet IQ PIC, include both statements.

For more information about the **scheduler-map** statement, see "Applying Scheduler Maps to Physical Interfaces" on page 168. For information about logical interface scheduling configuration, see "Applying Scheduler Maps and Shaping Rate to DLCIs and VLANs" on page 175.

Generally, when you include the **scheduler-map-chassis** statement in the configuration, you must use an interface wildcard for the interface name, as in ***type-fpc/pic/****. The wildcard must use this format—for example. **so-1/2/***, which means all interfaces on FPC slot 1, PIC slot 2. There is one exception—you can apply the chassis scheduler map to a specific interface on the Gigabit Ethernet IQ PIC only.

According to Junos OS wildcard rules, specific interface configurations override wildcard configurations. For chassis scheduler map configuration, this rule does not apply; instead, specific interface CoS configurations are added to the chassis scheduler map configuration. For more information about how wildcards work with chassis scheduler maps, see "Examples: Scheduling Packet Forwarding Component Queues" on page 196. For general information about wildcards, see the *Junos OS System Basics Configuration Guide*.

For more information, see the following sections:

- Applying Custom Schedulers to Packet Forwarding Component Queues on page 196
- Examples: Scheduling Packet Forwarding Component Queues on page 196

Applying Custom Schedulers to Packet Forwarding Component Queues

Optionally, you can apply a custom scheduler to the chassis queues instead of configuring the chassis queues to automatically derive their scheduling configuration from the logical interfaces on the PIC.

To assign a custom scheduler to the packet forwarding component queues, include the **scheduler-map-chassis** statement at the **[edit class-of-service interfaces *type-fpc/ pic*]** hierarchy level:

```
[edit class-of-service interfaces type-fpc/pic/*]
scheduler-map-chassis map-name;
```

For information about defining the scheduler map referenced by ***map-name***, see “Configuring Scheduler Maps” on page 167.

Examples: Scheduling Packet Forwarding Component Queues

Applying a Chassis Scheduler Map to a 2-Port IQ PIC

Apply a chassis scheduler map to interfaces **so-0/1/0** and **so-0/1/1**.

According to customary wildcard rules, the **so-0/1/0** configuration overrides the **so-0/1/*** configuration, implying that the chassis scheduler map **MAP1** is not applied to **so-0/1/0**. However, the wildcard rule is not obeyed in this case; the chassis scheduler map applies to both interfaces **so-0/1/0** and **so-0/1/1**.

```
[edit]
class-of-service {
  interfaces {
    so-0/1/0 {
      unit 0 {
        classifiers {
          inet-precedence default;
        }
      }
    }
    so-0/1/* {
      scheduler-map-chassis derived;
    }
  }
}
```

Not Recommended: Using a Wildcard for Gigabit Ethernet IQ Interfaces When Applying a Chassis Scheduler Map

On a Gigabit Ethernet IQ PIC, you can apply the chassis scheduler map at both the specific interface level and the wildcard level. We do not recommend this because the wildcard chassis scheduler map takes precedence, which might not be the desired effect. For example, if you want to apply the chassis scheduler map MAP1 to port 0 and MAP2 to port 1, we do not recommend the following:

```
[edit class-of-service]
interfaces {
  ge-0/1/0 {
    scheduler-map-chassis MAP1;
```



```
    }
    ge-0/1/* {
      scheduler-map-chassis MAP2;
    }
  }
```

Recommended: Identifying Gigabit Ethernet IQ Interfaces Individually When Applying a Chassis Scheduler Map

Instead, we recommend this configuration:

```
[edit class-of-service]
interfaces {
  ge-0/1/0 {
    scheduler-map-chassis MAP1;
  }
  ge-0/1/1 {
    scheduler-map-chassis MAP2;
  }
}
```

Configuring ATM CoS with a Normal Scheduler and a Chassis Scheduler

For ATM2 IQ interfaces, the CoS configuration differs significantly from that of other interface types. For more information about ATM CoS, see “CoS on ATM Interfaces Overview” on page 411.

```
[edit class-of-service]
interfaces {
  at-1/2/* {
    scheduler-map-chassis derived;
  }
}

[edit interfaces]
at-1/2/0 {
  atm-options {
    vpi 0;
    linear-red-profiles red-profile-1 {
      queue-depth 35000 high-plp-threshold 75 low-plp-threshold 25;
    }
    scheduler-maps map-1 {
      vc-cos-mode strict;
      forwarding-class best-effort {
        priority low;
        transmit-weight percent 25;
        linear-red-profile red-profile-1;
      }
    }
  }
  unit 0 {
    vci 0.128;
    shaping {
      vbr peak 20m sustained 10m burst 20;
    }
    atm-scheduler-map map-1;
    family inet {
      address 192.168.0.100/32 {
        destination 192.168.0.101;
      }
    }
```

```
  }
}
```

Configuring Two T3 Interfaces on a Channelized DS3 IQ PIC

```
[edit interfaces]
ct3-3/0/0 {
  no-partition interface-type t3; # use entire port 0 as T3
}
ct3-3/0/1 {
  no-partition interface-type t3; # use entire port 1 as T3
}
t3-3/0/0 {
  unit 0 {
    family inet {
      address 10.0.100.1/30;
    }
  }
}
t3-3/0/1 {
  unit 0 {
    family inet {
      address 10.0.101.1/30;
    }
  }
}
```

Applying Normal Schedulers to Two T3 Interfaces

Configure a scheduler for the aggregated traffic transmitted into both T3 interfaces.

```
[edit class-of-service]
interfaces {
  t3-3/0/0 {
    scheduler-map sched-qct3-0;
  }
  t3-3/0/1 {
    scheduler-map sched-qct3-1;
  }
}
scheduler-maps {
  sched-qct3-0 {
    forwarding-class best-effort scheduler be-qct3-0;
    forwarding-class expedited-forwarding scheduler ef-qct3-0;
    forwarding-class assured-forwarding scheduler as-qct3-0;
    forwarding-class network-control scheduler nc-qct3-0;
  }
  sched-qct3-1 {
    forwarding-class best-effort scheduler be-qct3-1;
    forwarding-class expedited-forwarding scheduler ef-qct3-1;
    forwarding-class assured-forwarding scheduler as-qct3-1;
    forwarding-class network-control scheduler nc-qct3-1;
  }
  sched-chassis-to-q {
    forwarding-class best-effort scheduler be-chassis;
    forwarding-class expedited-forwarding scheduler ef-chassis;
    forwarding-class assured-forwarding scheduler as-chassis;
    forwarding-class network-control scheduler nc-chassis;
  }
}
```

```
schedulers {
   be-qct3-0 {
      transmit-rate percent 40;
   }
   ef-qct3-0 {
      transmit-rate percent 30;
   }
   as-qct3-0 {
      transmit-rate percent 20;
   }
   nc-qct3-0 {
      transmit-rate percent 10;
   }
   ...
}
```

Applying a Chassis Scheduler to Two T3 Interfaces

Bind a scheduler to the aggregated traffic transmitted into the entire PIC. The chassis scheduler controls the traffic from the packet forwarding components feeding the interface **t3-3/0/***.

```
[edit class-of-service]
interfaces {
   t3-3/0/* {
      scheduler-map-chassis derived;
   }
}
```

Not Recommended: Using a Wildcard for Logical Interfaces When Applying a Scheduler

Do not apply a scheduler to a logical interface using a wildcard. For example, if you configure a logical interface (unit) with one parameter, and apply a scheduler map to the interface using a wildcard, the logical interface will not apply the scheduler. The following configuration will commit correctly but will not apply the scheduler map to interface **so-3/0/0.0**:

```
[edit class-of-service]
interfaces {
   so-3/0/* {
      unit 0 {
         scheduler-map MY_SCHED_MAP;
      }
   }
   so-3/0/0 {
      unit 0 {
         shaping-rate 100m;
      }
   }
}
```

Recommended: Identifying Logical Interfaces Individually When Applying a Scheduler

Always apply the scheduler to a logical interface without the wildcard:

```
[edit class-of-service]
interfaces {
   so-3/0/0 {
      unit 0 {
         scheduler-map MY_SCHED_MAP;
         shaping-rate 100m;
      }
```



```
  }
}
```

NOTE: This same wildcard behavior applies to classifiers and rewrites as well as schedulers.

Default Fabric Priority Queuing

On Juniper Networks M320 Multiservice Edge Routers and T Series Core Routers, the default behavior is for fabric priority queuing on egress interfaces to match the scheduling priority you assign. High-priority egress traffic is automatically assigned to high-priority fabric queues. Likewise, low-priority egress traffic is automatically assigned to low-priority fabric queues.

For information about overriding automatic fabric priority queuing, see “Overriding Fabric Priority Queuing” on page 120 and “Associating Schedulers with Fabric Priorities” on page 200.

Associating Schedulers with Fabric Priorities

On Juniper Networks M320 Multiservice Edge Routers and T Series Core Routers only, you can associate a scheduler with a class of traffic that has a specific priority while transiting the fabric. Traffic transiting the fabric can have two priority values: **low** or **high**. To associate a scheduler with a fabric priority, include the **priority** and **scheduler** statements at the **[edit class-of-service fabric scheduler-map]** hierarchy level:

```
[edit class-of-service fabric scheduler-map]
priority (high | low) scheduler scheduler-name;
```

NOTE: For a scheduler that you associate with a fabric priority, include only the **drop-profile-map** statement at the **[edit class-of-service schedulers *scheduler-name*]** hierarchy level. You cannot include the **buffer-size**, **transmit-rate**, and **priority** statements at that hierarchy level.

For information about associating a forwarding class with a fabric priority, see “Overriding Fabric Priority Queuing” on page 120.

Example: Associating a Scheduler with a Fabric Priority

Associate a scheduler with a class of traffic that has a specific priority while transiting the fabric:

```
[edit class-of-service]
schedulers {
  fab-be-scheduler {
    drop-profile-map loss-priority low protocol any drop-profile fab-profile-1;
    drop-profile-map loss-priority high protocol any drop-profile fab-profile-2;
  }
  fab-ef-scheduler {
```

```
        drop-profile-map loss-priority low protocol any drop-profile fab-profile-3;
        drop-profile-map loss-priority high protocol any drop-profile fab-profile-4;
    }
}
drop-profiles {
    fab-profile-1 {
        fill-level 100 drop-probability 100;
        fill-level 85 drop-probability 50;
    }
    fab-profile-2 {
        fill-level 100 drop-probability 100;
        fill-level 95 drop-probability 50;
    }
    fab-profile-3 {
        fill-level 75 drop-probability 100;
        fill-level 95 drop-probability 50;
    }
    fab-profile-4 {
        fill-level 100 drop-probability 100;
        fill-level 80 drop-probability 50;
    }
}
fabric {
    scheduler-map {
        priority low scheduler fab-be-scheduler;
        priority high scheduler fab-ef-scheduler;
    }
}
```

Configuring the Number of Schedulers for Ethernet IQ2 PICs

You can oversubscribe the Ethernet IQ2 family of PICs. Because of the bursty nature of Ethernet use, traffic received by the PIC can be several orders of magnitude greater than the maximum bandwidth leaving the PIC and entering the router. Several configuration statements apply only to Ethernet IQ2 PICs and allow the PIC to intelligently handle the oversubscribed traffic.

NOTE: The total of the input guaranteed rates for oversubscribed IQ2 PICs is limited to the FPC or PIC bandwidth.

This section discusses the following topics:

Ethernet IQ2 PIC Schedulers

By default, each Ethernet IQ2 PIC is allocated a fixed number of the 1024 available schedulers for each port during PIC initialization. For example, the 8-port Gigabit Ethernet IQ2 PIC is allocated 128 schedulers for each port. This number cannot be changed after the PIC is operational and can limit the utilization of shapers among the ports. Each of the 1024 schedulers is mapped at the logical interface (unit) level, and each scheduler can support up to eight forwarding classes.

Schedulers are allocated in multiples of four. Three schedulers are reserved on each port. One is for control traffic, one is for port-level shaping, and the last is for unshaped logical interface traffic. These are allocated internally and automatically. The fourth scheduler is added when VLANs are configured.

When you configure schedulers for a port on an Ethernet IQ2 PIC:

- The three reserved schedulers are added to the configured value, which yields four schedulers per port.
- The configured value is adjusted upward to the nearest multiple of 4 (schedulers are allocated in multiples of 4).
- After all configured schedulers are allocated, any remaining unallocated schedulers are partitioned equally across the other ports.
- Any remaining schedulers that cannot be allocated meaningfully across the ports are allocated to the last port.

If the configured scheduler number is changed, the Ethernet IQ2 PIC is restarted when the configuration is committed.

NOTE: If you deactivate and reactivate a port configured with a non-default number of schedulers then the whole Ethernet IQ2 PIC restarts.

To configure the number of schedulers assigned to a port on an Ethernet IQ2 PIC, include the **schedulers** statement for the Ethernet IQ2 PIC interface at the **[edit interfaces ge-*fpc/pic/port*]** hierarchy level:

```
[edit interfaces ge-fpc/pic/port]
schedulers number;
```

You can configure between 1 and 1024 schedulers on a port.

Example: Configuring a Scheduler Number for an Ethernet IQ2 PIC Port

This example allocates 100 schedulers to port 1 on an 8-port Gigabit Ethernet IQ2 PIC. The example shows the final scheduler allocation numbers for each port on the PIC. By default, each port would have been allocated 1024 / 8 = 128 schedulers.

```
[edit interfaces]
ge-1/2/1 {
  schedulers 100;
}
```

This configuration results in the port and scheduler configuration shown in Table 32 on page 203.

Table 32: Scheduler Allocation for an Ethernet IQ2 PIC

Ethernet IQ2 PIC Port	Number of Allocated Schedulers
0	128
1	104 (100 configured, plus 3 reserved, rounded up to multiple of 4: 100 + 3 +1= 104)
2	128
3	128
4	128
5	128
6	128
7	152 (128 plus the 24 remaining that cannot be meaningfully allocated to other ports)

Ethernet IQ2 PIC RTT Delay Buffer Values

The following table shows the round-trip time (RTT) delay buffer values for IQ2 PICs, which are nonstandard and vary by PIC type and direction. The values are rounded up slightly to account for oversubscription.

Table 33: RTT Delay Buffers for IQ2 PICs

IQ2 PIC Type	Ingress Buffer (ms)	Egress Buffer (ms)
4–port Gigabit Ethernet (Type 1)	200	300
8–port Gigabit Ethernet (Type 2)	175	200
8–port Gigabit Ethernet (Type 3)	35	225
1–port 10–Gigabit Ethernet (Type 3)	25	190

Configuring Rate Limiting and Sharing of Excess Bandwidth on Multiservices PICs

On Multiservices PICs, you can limit the transmit rate of a logical interface (**lsq-**) in the same way as other types of queuing PICs. You can also assign a percentage of the excess bandwidth to the logical interfaces. As with other types of PICs, the strict-high queue (voice) can “starve” low and medium priority queues. To prevent the strict-high queue from starving other queues, rate-limit the queue.

To rate-limit logical interfaces on a Multiservices PIC, include the **transmit-rate** statement with the **rate-limit** option at the **[edit class-of-service schedulers *scheduler-name*]** hierarchy level:

```
[edit class-of-service schedulers scheduler-name]
transmit-rate (rate | percent percentage | remainder) rate-limit;
```

You can also make the excess strict-high bandwidth available for other queues. You can split the excess bandwidth among multiple queues, but the total excess bandwidth assigned to these queues can only add up to 100 percent. The excess-bandwidth **priority** statement option is not supported on the Multiservices PIC. For more information about excess bandwidth sharing, see "Configuring Excess Bandwidth Sharing on IQE PICs" on page 295.

To share excess bandwidth among Multiservices PICs, include the **excess-rate** statement at the **[edit class-of-service schedulers *scheduler-name*]** hierarchy level.

```
[edit class-of-service schedulers scheduler-name]
excess-rate percent percentage;
```

Both of these rate-limiting and excess bandwidth sharing features apply to egress traffic only, and only for per-unit schedulers. Hierarchical schedulers and shared schedulers are not supported.

You must still complete the configuration by configuring the scheduler map and applying it to the Multiservices PIC interface.

This example configures a rate limit and excess bandwidth sharing for a Multiservices PIC interface.

```
[edit class-of-service schedulers]
scheduler0 {
  transmit-rate percent 10 rate-limit;
  priority strict-high;
  excess-rate percent 30;
}
scheduler1 {
  transmit-rate percent 1m rate-limit;
  priority high;
  excess-rate percent 70;
}

[edit class-of-service scheduler-maps]
scheduler0 {
  forwarding-class ef scheduler scheduler0;
  forwarding-class af scheduler scheduler1;
}

[edit class-of-service interfaces lsq-1/3/0]
unit 0 {
  scheduler-map scheduler0;
}
unit 1 {
  scheduler-map scheduler1;
}
```


CHAPTER 11

Configuring Hierarchical Schedulers

This topic discusses the following:

Hierarchical Schedulers Terminology

Hierarchical schedulers introduce some new terms into a discussion of CoS capabilities. They also use some familiar terms it different contexts. This section presents a complete overview of the terms used with hierarchical schedulers.

The following terms are important for hierarchical schedulers:

- Customer VLAN (C-VLAN)—A C-VLAN, defined by IEEE 802.1ad. A stacked VLAN contains an outer tag corresponding to the S-VLAN, and an inner tag corresponding to the C-VLAN. A C-VLAN often corresponds to CPE. Scheduling and shaping is often used on a C-VLAN to establish minimum and maximum bandwidth limits for a customer. See also *S-VLAN*.
- Interface set—A logical group of interfaces that describe the characteristics of set of service VLANs, logical interfaces, or customer VLANs. Interface sets establish the set and name the traffic control profiles. See also *Service VLAN*.
- Scheduler— A scheduler defines the scheduling and queuing characteristics of a queue. Transmit rate, scheduler priority, and buffer size can be specified. In addition, a drop

profile may be referenced to describe WRED congestion control aspects of the queue. See also *Scheduler map*.

- Scheduler map—A scheduler map is referenced by traffic control profiles to define queues. The scheduler map establishes the queues that comprise a scheduler node and associates a forwarding class with a scheduler. See also *Scheduler*.
- Stacked VLAN—An encapsulation on an S-VLAN with an outer tag corresponding to the S-VLAN, and an inner tag corresponding to the C-VLAN. See also *Service VLAN* and *Customer VLAN*.
- Service VLAN (S-VLAN)—An S-VLAN, defined by IEEE 802.1ad, often corresponds to a network aggregation device such as a DSLAM. Scheduling and shaping is often established for an S-VLAN to provide CoS for downstream devices with little buffering and simple schedulers. See also *Customer VLAN*.
- Traffic control profile—Defines the characteristics of a scheduler node. Traffic control profiles are used at several levels of the CLI, including the physical interface, interface set, and logical interface levels. Scheduling and queuing characteristics can be defined for the scheduler node using the **shaping-rate**, **guaranteed-rate**, and **delay-buffer-rate** statements. Queues over these scheduler nodes are defined by referencing a scheduler map. See also *Scheduler* and *Scheduler map*.
- VLAN—Virtual LAN, defined on an Ethernet logical interface.

These terms are especially important when applied to a scheduler hierarchy. Scheduler hierarchies are composed 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 34 on page 206 shows how the configuration of an interface set or logical interface affects the terminology of hierarchical scheduler nodes.

Table 34: Hierarchical Scheduler Nodes

Root Node (Level 1)	Level 2	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

Scheduler hierarchies consist of levels, starting with Level 1 at the physical port. This chapter establishes a four-level scheduler hierarchy which, when fully configured, consists of the physical interface (Level 1), the interface set (Level 2), one or more logical interfaces (Level 3), and one or more queues (Level 4).

Configuring Hierarchical Schedulers for CoS

In metro Ethernet environments, a virtual LAN (VLAN) typically corresponds to a customer premises equipment (CPE) device and the VLANs are identified by an inner VLAN tag on Ethernet frames (called the customer VLAN, or C-VLAN, tag). A set of VLANs can be grouped at the DSL access multiplexer (DSLAM) and identified by using the same outer VLAN tag (called the service VLAN, or S-VLAN, tag). The service VLANs are typically gathered at the Broadband Remote Access Server (B-RAS) level. Hierarchical schedulers let you provide shaping and scheduling at the service VLAN level as well as other levels, such as the physical interface. In other words, you can group a set of logical interfaces and then apply scheduling and shaping parameters to the logical interface set as well as to other levels.

On Juniper Networks MX Series Ethernet Services Routers and systems with Enhanced IQ2 (IQ2E) PICs, you can apply CoS shaping and scheduling at one of four different levels, including the VLAN set level. You can only use this configuration on MX Series routers or IQ2E PICs. For more information about configuring CoS on IQ2E PICs, see "CoS on Enhanced IQ2 PICs Overview" on page 325.

The supported scheduler hierarchy is as follows:

- The physical interface (level 1)
- The service VLAN (level 2 is unique to MX Series routers)
- The logical interface or customer VLAN (level 3)
- The queue (level 4)

Users can specify a traffic control profile (**output-traffic-control-profile** that can specify a shaping rate, a guaranteed rate, and a scheduler map with transmit rate and buffer delay. The scheduler map contains the mapping of queues (forwarding classes) to their respective schedulers (schedulers define the properties for the queue). Queue properties can specify a transmit rate and buffer management parameters such as buffer size and drop profile.

To configure CoS hierarchical schedulers, include the following statements at the **[edit class-of-service interfaces]** and **[edit interfaces]** hierarchy levels:

```
[edit class-of-service interfaces]
interface-set interface-set-name {
  excess-bandwith-share (proportional value | equal);
  internal-node;
  output-traffic-control-profile profile-name;
  output-traffic-control-profile-remaining profile-name;
}

[edit interfaces]
hierarchical-scheduler;
interface-set interface-set-name {
  ethernet-interface-name {
    (interface-parameters);
  }
```

```
}
```

Configuring Interface Sets

To configure an interface set, include the **interface-set** statement at the **[edit class-of-service interfaces]** hierarchy level:

```
[edit class-of-service interfaces]
interface-set interface-set-name {
  ...interface-cos-configuration-statements ...
}
```

To apply the interface set to interfaces, include the **interface-set** statement at the **[edit interfaces]** hierarchy level:

```
[edit interfaces]
interface-set interface-set-name {
  interface ethernet-interface-name {
    ... interface-cos-configuration-statements ...
  }
}
```

Interface sets can be defined in two major ways: as a list of logical interfaces (**unit 100**, **unit 200**, and so on), or at the stacked VLAN level using a list of outer VLAN IDs (**vlan-tags-outer 210**, **vlan-tags-outer 220**, and so on). In addition, the **svlan** ***number*** listing option with a single outer VLAN tag is a convenient way for specifying a set of VLAN members having the same outer VLAN tags. 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.

Whether using the logical interface listing option for a group of customer VLANs or the S-VLAN set listing option for a group of VLAN outer tags, 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.

Regardless of listing convention, you can only use one of the types in an interface set. Examples of this limitation appear later in this section.

Interface sets are currently used only by CoS, but they are applied at the **[edit interfaces]** hierarchy level to make them available to other services that might use them in future.

```
[edit interfaces]
interface-set interface-set-name {
  interface ethernet-interface-name {
    (unit logical-unit-number | vlan-tags-outer vlan-tag) {
      ...
    }
  }
}
```

The logical interface naming option lists Ethernet interfaces:

```
[edit interfaces]
interface-set unitl-set-ge-0 {
  interface ge-0/0/0 {
```



```
        unit 0;
        unit 1;
        ...
    }
}
```

The S-VLAN option lists only one S-VLAN (outer) tag value:

```
[edit interfaces]
interface-set svlan-set {
    interface ge-1/0/0 {
        vlan-tags-outer 2000;
    }
}
```

The S-VLAN naming option lists S-VLAN (outer) tag values:

```
[edit interfaces]
interface-set svlan-set-tags {
    interface ge-2/0/0 {
        vlan-tags-outer 2000;
        vlan-tags-outer 2001;
        vlan-tags-outer 2002;
        ...
    }
}
```

NOTE: Ranges are not supported: you must list each VLAN or logical interface separately.

Applying Interface Sets

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.

This example applies a traffic control profile called **tcp-set1** to an interface set called **set-ge-0**:

```
[edit class-of-service interfaces]
interface-set set-ge-0 {
    output-traffic-control-profile tcp-set1;
}
```

Interface Set Caveats

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 operation fails.

This example generates a commit error:

```
[edit interfaces]
interface-set set-one {
   interface ge-2/0/0 {
      unit 0;
      unit 2;
   }
}
interface-set set-two {
   interface 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 {
   interface ge-0/0/1 {
      unit 0;
      unit 1;
   }
   interface ge-0/0/2 { # This is NOT supported in the same interface set!
      unit 0;
      unit 1;
   }
}
```

Hierarchical Schedulers and Traffic Control Profiles

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.

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), scheduler maps (assigning queues and resources to traffic), and so on.

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

For more information on traffic control profiles see "Oversubscribing Interface Bandwidth" on page 184 and "Providing a Guaranteed Minimum Rate" on page 191. For more information on scheduler maps, see "Configuring Scheduler Maps" on page 167.

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.

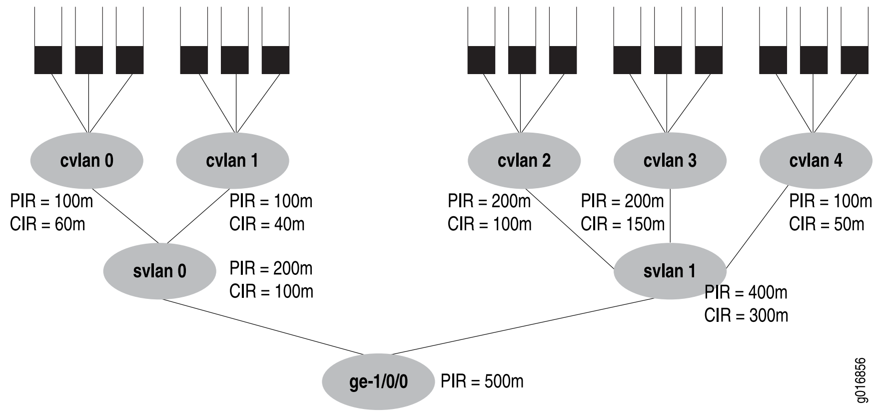
```
[edit class-of-service interfaces]
interface-set level-2 {
  output-traffic-control-profile tcp-interface-level-2;
}
ge-0/1/0 {
  output-traffic-control-profile tcp-port-level-1;
  unit 0 {
    output-traffic-control-profile tcp-unit-level-3;
  }
}
```

In all cases, the properties for level 4 of the hierarchical schedulers are determined by the scheduler map.

Example: Four-Level Hierarchy of Schedulers

This section provides a more complete example of building a 4-level hierarchy of schedulers. The configuration parameters are shown in Figure 14 on page 212. The queues are shown at the top of the figure with the other three levels of the hierarchy below.

Figure 14: Building a Scheduler Hierarchy



The figure's PIR values are configured as the shaping rates and the CIRs are 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).

This configuration example presents all details of the CoS configuration for the interface in the figure (**ge-1/0/0**), including:

- Configuring the Interface Sets on page 212
- Configuring the Interfaces on page 213
- Configuring the Traffic Control Profiles on page 213
- Configuring the Schedulers on page 214
- Configuring the Drop Profiles on page 215
- Configuring the Scheduler Maps on page 215
- Applying the Traffic Control Profiles on page 215

Configuring the Interface Sets

```
[edit 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;
  }
}
```

Configuring the Interfaces

The keyword to configure hierarchical schedulers is at the physical interface level, as is 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.

```
[edit 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;
}
```

Configuring the Traffic Control Profiles

The traffic control profiles hold parameters for levels above the queue level of the scheduler hierarchy. This section defines traffic control profiles for both the service VLAN level (logical interfaces) and the customer VLAN (VLAN tag) level.

```
[edit 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; # Applies scheduler maps to customer VLANs.
```

```
}
tcp-cvlan1 {
  shaping-rate 100m;
  guaranteed-rate 40m;
  scheduler-map tcp-map-cvlan1; # Applies scheduler maps to customer VLANs.
}
tcp-cvlan2 {
  shaping-rate 200m;
  guaranteed-rate 100m;
  scheduler-map tcp-map-cvlanx; # Applies scheduler maps to customer VLANs.
}
tcp-cvlan3 {
  shaping-rate 200m;
  guaranteed-rate 150m;
  scheduler-map tcp-map-cvlanx; # Applies scheduler maps to customer VLANs
}
tcp-cvlan4 {
  shaping-rate 100m;
  guaranteed-rate 50m;
  scheduler-map tcp-map-cvlanx; # Applies scheduler maps to customer VLANs
}
```

Configuring the Schedulers

The schedulers hold the information about the queues, the last level of the hierarchy. Note the consistent naming schemes applied to repetitive elements in all parts of this example.

```
[edit class-of-service schedulers]
sched-cvlan0-qx {
  priority low;
  transmit-rate 20m;
  buffer-size temporal 100ms;
  drop-profile loss-priority low dp-low;
  drop-profile loss-priority high dp-high;
}
sched-cvlan1-q0 {
  priority high;
  transmit-rate 20m;
  buffer-size percent 40;
  drop-profile loss-priority low dp-low;
  drop-profile loss-priority high dp-high;
}
sched-cvlanx-qx {
  transmit-rate percent 30;
  buffer-size percent 30;
  drop-profile loss-priority low dp-low;
  drop-profile loss-priority high dp-high;
}
sched-cvlan1-qx {
  transmit-rate 10m;
  buffer-size temporal 100ms;
  drop-profile loss-priority low dp-low;
  drop-profile loss-priority high dp-high;
}
```

Configuring the Drop Profiles

This section configures the drop profiles for the example. For more information about interpolated drop profiles, see “RED Drop Profiles Overview” on page 231.

```
[edit class-of-service drop-profiles]
dp-low {
   interpolate fill-level 80 drop-probability 80;
   interpolate fill-level 100 drop-probability 100;
}
dp-high {
   interpolate fill-level 60 drop-probability 80;
   interpolate fill-level 80 drop-probability 100;
}
```

Configuring the Scheduler Maps

This section configures the scheduler maps for the example. Each one references a scheduler configured in “Configuring the Schedulers” on page 214.

```
[edit 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;
}
```

Applying the Traffic Control Profiles

This section applies the traffic control profiles to the proper levels of the hierarchy.

NOTE: Although a shaping rate can be applied directly to the physical interface, hierarchical schedulers must use a traffic control profile to hold this parameter.

```
[edit 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;
  }
```

NOTE: You should be careful when using a **show interfaces queue** command that references nonexistent class-of-service logical interfaces. When multiple logical interfaces (units) but are not configured under the same interface set or physical interface, but are referenced by a command such as **show interfaces queue ge-10/0/1.12 forwarding-class be** or **show interfaces queue ge-10/0/1.13 forwarding-class be** (where logical units 12 and 13 are not configured as a class-of-service interfaces), these interfaces display the same traffic statistics for each logical interface. In other words, even if there is no traffic passing through a particular unconfigured logical interface, as long as one or more of the other unconfigured logical interfaces under the same interface set or physical interface is passing traffic, this particular logical interface displays statistics counters showing the total amount of traffic passed through all other unconfigured logical interfaces together.

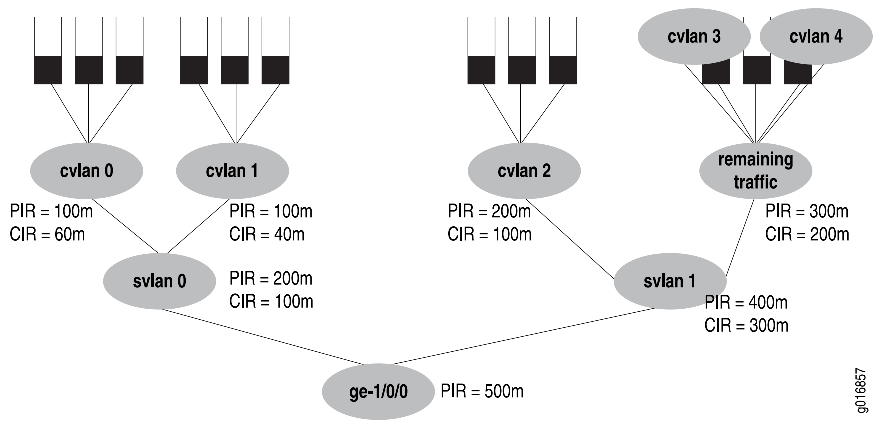
Controlling Remaining Traffic

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 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*. 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. In the same way, the **shaping-rate** and **delay-buffer-rate** statements can be specified in the traffic control profile referenced with the **output-traffic-control-profile-remaining** statement in order to shape and provide buffering for remaining traffic.

Consider the interface shown in Figure 15 on page 217. 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 customer

VLANs. The solution in to configure and apply a traffic control profile for all remaining traffic on the interface.

Figure 15: Handling Remaining Traffic

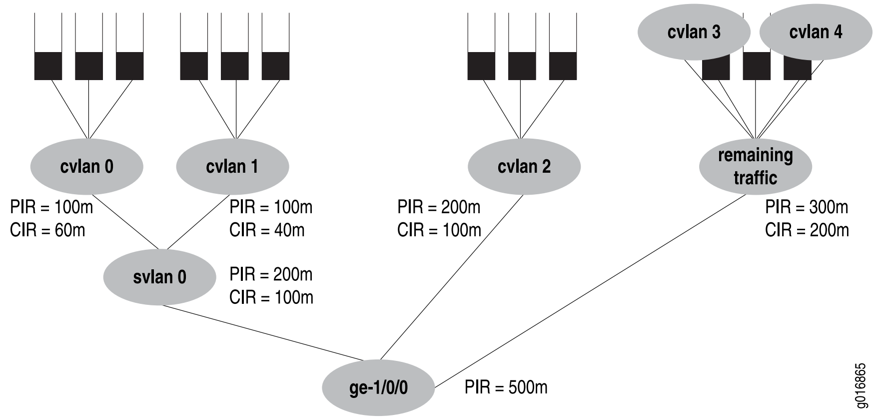


This example considers the case where customer VLANs 3 and 4 have no explicit traffic control profile, yet need to establish a shaping and guaranteed transmit rate for traffic heading for those customer VLANs. The solution is to add a traffic control profile to the **svlan1** interface set. This example builds on the earlier example and so does not repeat all configuration details, only those at the service VLAN level.

```
[edit class-of-service interfaces]
interface-set svlan0 {
  output-traffic-control-profile tcp-svlan0;
}
interface-set svlan1 {
  output-traffic-control-profile tcp-svlan1; # For explicitly shaped traffic.
  output-traffic-control-profile-remaining tcp-svlan1-remaining; # For all remaining traffic.
}

[edit 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
}
```

Next, consider the example shown in Figure 16 on page 218.

Figure 16: Another Example of Handling Remaining Traffic



In this example, **ge-1/0/0** has three logical interfaces (unit 1, unit 2, and unit 3), and SVLAN 2000, 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 this example, **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**.

This example does not include the **[edit interfaces]** configuration.

```
[edit 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.
   }
}
```


Here is how the traffic control profiles for this example are configured:

```
[edit 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;
}
```

Finally, here are the scheduler maps and queues for the example:

```
[edit 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;
}
```

The configuration for the referenced schedulers are not given for this example.

Configuring Internal Scheduler Nodes

A node in the hierarchy is considered internal if either of the following conditions apply:

- Any one of its children nodes has a traffic control profile configured and applied.
- You include the **internal-node** statement at the **[edit class-of-service interfaces interface-set *set-name*]** hierarchy level.

Why would it be important to make a certain node internal? Generally, there are more resources available at the logical interface (unit) level than at the interface set level. Also, 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.

The **internal-node** statement can be used 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.

In summary, using the **internal-node** statement allows statements to all be scheduled at the same level with or without children.

The following example makes the interfaces 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.

PIR-Only and CIR Mode

The actual behavior of many CoS parameters, especially the shaping rate and guaranteed rate, depend on whether the physical interface is operating in PIR-only or CIR mode.

In PIR-only mode, one or more nodes perform shaping. The physical interface is in the 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 CIR 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 CIR mode, one or more nodes applies the guaranteed rates. 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).

Priority Propagation

Juniper Networks MX Series Ethernet Services Routers with Enhanced Queuing DPCs and M Series and T Series routers with IQ2E PIC 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 due to the data traffic of another customer. Nodes and queues are always services 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 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 only relevant if the physical interface is in CIR mode).

Each queue has a configured priority and a hardware priority. The usual mapping between the configured priority and the hardware priority is shown in Table 35 on page 221.

Table 35: Queue Priority

Configured Priority	Hardware Priority
Strict-high	0
High	0
Medium-high	1
Medium-low	1
Low	2

In CIR mode, the priority for each internal node depends on whether the highest active child node is above or below the guaranteed rate. The mapping between the highest active child's priority and the hardware priority below and above the guaranteed rate is shown in Table 36 on page 221.

Table 36: 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
High	0	3
Medium-high	1	3
Medium-low	1	3
Low	2	3

In PIR-only mode, nodes cannot send if they are above the configured shaping rate. The mapping between the configured priority and the hardware priority is for PIR-only mode is shown in Table 37 on page 221.

Table 37: Internal Node Queue Priority for PIR-Only Mode

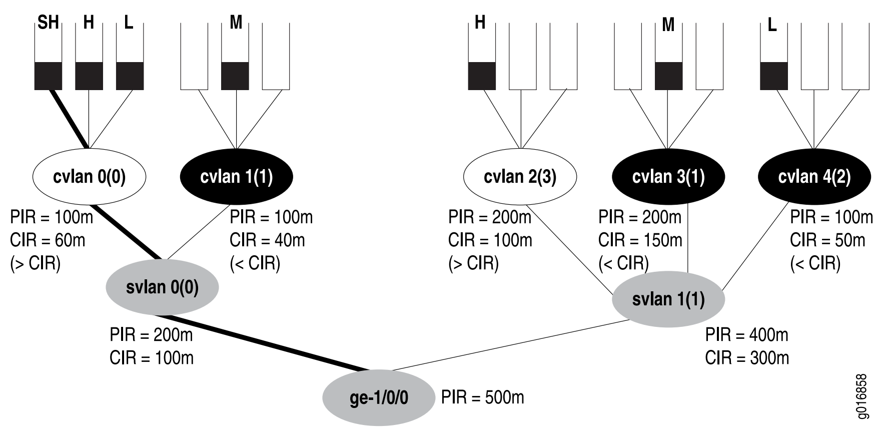
Configured Priority	Hardware Priority
Strict-high	0
High	0

Table 37: Internal Node Queue Priority for PIR-Only Mode *(continued)*

Configured Priority	Hardware Priority
Medium-high	1
Medium-low	1
Low	2

A physical interface with hierarchical schedulers configured is shown in Figure 17 on page 222. 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 is above or below the CIR. The nodes are shown in one of three states: above the CIR (clear), below the CIR (dark), or in a condition where the CIR does not matter (gray).

Figure 17: Hierarchical Schedulers and Priorities



In the figure, the strict-high queue for customer VLAN 0 (cvlan 0) receives service first, even though the customer VLAN is above the configured CIR (see Table 36 on page 221 for the reason: strict-high always has hardware priority 0 regardless of CIR state). Once that queue has been drained, and the priority of the node has become 3 instead of 0 (due to the lack of strict-high traffic), the system moves on to the medium queues next (cvlan 1 and cvlan 3), draining them in a round robin fashion (empty queue 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 (thanks to svlan 0 having a priority of 3).

CHAPTER 12

Configuring Queue-Level Bandwidth Sharing

This topic includes the following:

Bandwidth Sharing on Nonqueuing Packet Forwarding Engines Overview

You can configure bandwidth sharing rate limits, excess rate, and excess priority at the queue level on the following Juniper Networks routers:

- M120 Multiservice Edge Router (rate limit and excess priority only; excess rate is not configured by the user)
- M320 router with Enhanced FPCs (rate limit, excess rate, and excess priority)
- MX Series Ethernet Services Router with nonqueuing DPCs (rate limit, excess rate, and excess priority)

You configure rate limits when you have a concern that low-latency packets (such as high or strict-high priority packets for voice) might starve low-priority and medium-priority packets. In Junos OS, the low latency queue is implemented by rate-limiting packets to the transmit bandwidth. The rate-limiting is performed immediately before queueing the packet for transmission. All packets that exceed the rate limit not queued but dropped.

By default, if the excess priority is not configured for a queue, the excess priority will be the same as the normal queue priority. If none of the queues have an excess rate configured, then the excess rate will be the same as the transmit rate percentage. If at least one of the queues has an excess rate configured, then the excess rate for the queues that do not have an excess rate configured will be set to zero.

When the physical interface is on queuing hardware such as the IQ, IQ2, or IQE PICs, or MX Series routers queuing DPCs, these features are dependent on the PIC (or queuing DPC in the case of the MX Series router) configuration.

You cannot configure both rate limits and buffer sizes on these Packet Forwarding Engines.

Four levels of excess priorities are supported: low, medium-low, medium-high, and high.

Configuring Rate Limits on Nonqueuing Packet Forwarding Engines

To configure rate limits for nonqueuing Packet Forwarding Engines, include the **transmit-rate** statement at the **[edit class-of-service schedulers *scheduler-name*]** hierarchy level.

Configuring the Schedulers

The following example configures schedulers, forwarding classes, and a scheduler map for a rate-limited interface.

```
[edit class-of-service schedulers]
scheduler-1 {
  transmit-rate percent 20 rate-limit;
  priority high;
}
scheduler-2 {
  transmit-rate percent 10 rate-limit;
  priority strict-high;
}
scheduler-3 {
  transmit-rate percent 40;
  priority medium-high;
}
scheduler-4 {
  transmit-rate percent 30;
  priority medium-high;
}
```

Configuring the Forwarding Classes

```
[edit class-of-service]
forwarding-classes {
  class cp_000 queue-num 0;
  class cp_001 queue-num 1;
  class cp_010 queue-num 2;
  class cp_011 queue-num 3;
  class cp_100 queue-num 4;
  class cp_101 queue-num 5;
  class cp_110 queue-num 6;
  class cp_111 queue-num 7;
}
```

Configuring the Scheduler Map

```
[edit class-of-service scheduler-maps]
scheduler-map-1 {
  forwarding-class cp_000 scheduler scheduler-1;
  forwarding-class cp_001 scheduler scheduler-2;
  forwarding-class cp_010 scheduler scheduler-3;
  forwarding-class cp_011 scheduler scheduler-4;
}
```

Applying the Scheduler Map to the Interface

```
[edit interfaces]
ge-1/0/0 {
  scheduler-map scheduler-map-1;
  unit 0 {
    family inet {
      address 192.168.1.1/32;
```



```
            }
        }
    }
```

Excess Rate and Excess Priority Configuration Examples

To configure the excess rate for nonqueuing Packet Forwarding Engines, include the **excess-rate** statement at the **[edit class-of-service schedulers *scheduler-name*]** hierarchy level.

To configure the excess priority for nonqueuing Packet Forwarding Engines, include the **excess-priority** statement at the **[edit class-of-service schedulers *scheduler-name*]** hierarchy level.

The relationship between the configured guaranteed rate, excess rate, guaranteed priority, excess priority, and offered load is not always obvious. The following tables show the expected throughput of a Gigabit Ethernet port with various bandwidth-sharing parameters configured on the queues.

The default behavior of a nonqueuing Gigabit Ethernet interface with multiple priority levels is shown in Table 38 on page 225. All queues in the table get their guaranteed rate. The excess bandwidth is first offered to the excess high-priority queues. Because these use all available bandwidth, these is no remaining excess bandwidth for the low-priority queues.

Table 38: Current Behavior with Multiple Priority Levels

Queue	Guaranteed (Transmit) Rate	Guaranteed Priority	Excess Priority	Offered Load	Expected Throughput
Q0	20%	high	high	600 Mbps	200 + 366.67 = 566.67 Mbps
Q1	10%	high	high	500 Mbps	100 + 183.33 = 283.33 Mbps
Q2	10%	low	low	500 Mbps	100 + 0 = 100 Mbps
Q3	5%	low	low	500 Mbps	50 + 0 = 50 Mbps

The default behavior of a nonqueuing Gigabit Ethernet interface with the same priority levels is shown in Table 39 on page 225. All queues in the table get their guaranteed rate. Because all queues have the same excess priority, they share the excess bandwidth and each queue gets excess bandwidth in proportion to the transmit rate.

Table 39: Current Behavior with Same Priority Levels

Queue	Guaranteed (Transmit) Rate	Guaranteed Priority	Excess Priority	Offered Load	Expected Throughput
Q0	20%	high	high	500 Mbps	200 + 244.44 = 444.44 Mbps
Q1	10%	high	high	500 Mbps	100 + 122.22 = 222.22 Mbps

Table 39: Current Behavior with Same Priority Levels *(continued)*

Queue	Guaranteed (Transmit) Rate	Guaranteed Priority	Excess Priority	Offered Load	Expected Throughput
Q2	10%	high	high	500 Mbps	100 + 122.22 = 222.22 Mbps
Q3	5%	high	high	500 Mbps	50 + 61.11= 111.11 Mbps

The default behavior of a nonqueuing Gigabit Ethernet interface with the at least one strict-high priority level is shown in Table 40 on page 226. First the high priority and strict-high are serviced in a weighted round-robin fashion. The high priority queue gets its guaranteed bandwidth and the strict-high queue gets what remains. The high excess priority queue gets all the excess bandwidth.

Table 40: Current Behavior with Strict-High Priority

Queue	Guaranteed (Transmit) Rate	Guaranteed Priority	Excess Priority	Offered Load	Expected Throughput
Q0	20%	strict-high	X	500 Mbps	500 Mbps
Q1	10%	high	high	500 Mbps	100 + 250 = 350 Mbps
Q2	10%	low	low	500 Mbps	100 + 0 = 100 Mbps
Q3	5%	low	low	500 Mbps	50 + 0= 50 Mbps

The default behavior of a nonqueuing Gigabit Ethernet interface with the at least one strict-high priority level and a higher offered load on Q0 is shown in Table 41 on page 226. First the high priority and strict-high are serviced in a weighted round-robin fashion. The high priority queue gets its guaranteed bandwidth and the strict-high queue gets what remains. There is no excess bandwidth.

Table 41: Strict-High Priority with Higher Load

Queue	Guaranteed (Transmit) Rate	Guaranteed Priority	Excess Priority	Offered Load	Expected Throughput
Q0	20%	strict-high	X	1 Gbps	900 Mbps
Q1	10%	high	high	500 Mbps	100 + 0 = 100 Mbps
Q2	10%	low	low	500 Mbps	0 + 0 = 0 Mbps
Q3	5%	low	low	500 Mbps	0 + 0= 0 Mbps

Now consider the behavior of the queues with configured excess rates and excess priorities.

The behavior with multiple priority levels is shown in Table 42 on page 227. All queues get the guaranteed rate. The excess bandwidth is first offered to the excess high priority queues and these consume all the bandwidth. There is no remaining excess bandwidth for low priority queues.

Table 42: Sharing with Multiple Priority Levels

Queue	Guaranteed (Transmit) Rate	Excess Rate	Guaranteed Priority	Excess Priority	Offered Load	Expected Throughput
Q0	20%	10%	high	high	500 Mbps	200 + 275 = 475 Mbps
Q1	10%	20%	high	low	500 Mbps	100 + 0 = 100 Mbps
Q2	10%	10%	low	high	500 Mbps	100 + 275 = 275 Mbps
Q3	5%	20%	low	low	500 Mbps	50 + 0= 50 Mbps

The behavior with the same (high) priority levels is shown in Table 43 on page 227. All queues get the guaranteed rate. Because all queues have the same excess priority, they share the excess bandwidth in proportion to their transmit rate.

Table 43: Sharing with the Same Priority Levels

Queue	Guaranteed (Transmit) Rate	Excess Rate	Guaranteed Priority	Excess Priority	Offered Load	Expected Throughput
Q0	20%	10%	high	high	500 Mbps	200 + 91.67 = 291.67 Mbps
Q1	10%	20%	high	high	500 Mbps	100 + 183.33 = 283.33 Mbps
Q2	10%	10%	high	high	500 Mbps	100 + 91.67 = 191.67 Mbps
Q3	5%	20%	high	high	500 Mbps	50 + 183.33 = 233.33 Mbps

The behavior with at least one strict-high priority level is shown in Table 44 on page 227. The high priority and strict-high queues are serviced in a weighted round-robin fashion. The high priority queue gets its guaranteed rate and the strict-high queue gets the rest. The excess high-priority queue get all the excess bandwidth.

Table 44: Sharing with at Least One Strict-High Priority

Queue	Guaranteed (Transmit) Rate	Excess Rate	Guaranteed Priority	Excess Priority	Offered Load	Expected Throughput
Q0	20%	X	strict-high	X	500 Mbps	500 Mbps
Q1	10%	20%	high	low	500 Mbps	100 + 0 = 100 Mbps

Table 44: Sharing with at Least One Strict-High Priority *(continued)*

Queue	Guaranteed (Transmit) Rate	Excess Rate	Guaranteed Priority	Excess Priority	Offered Load	Expected Throughput
Q2	10%	10%	low	high	500 Mbps	100 + 250 = 350 Mbps
Q3	5%	20%	low	low	500 Mbps	50 + 0 = 50 Mbps

The behavior with at least one strict-high priority level and a higher offered load is shown in Table 45 on page 228. The high priority and strict-high queues are serviced in a weighted round-robin fashion. The high priority queue gets its guaranteed rate and the strict-high queue gets the rest. There is no excess bandwidth.

Table 45: Sharing with at Least One Strict-High Priority and Higher Load

Queue	Guaranteed (Transmit) Rate	Excess Rate	Guaranteed Priority	Excess Priority	Offered Load	Expected Throughput
Q0	20%	X	strict-high	X	900 Mbps	900 Mbps
Q1	10%	20%	high	low	500 Mbps	100 + 0 = 100 Mbps
Q2	10%	10%	low	high	500 Mbps	0 + 0 = 0 Mbps
Q3	5%	20%	low	low	500 Mbps	0 + 0 = 0 Mbps

The behavior with at least one strict-high priority level and a rate limit is shown in Table 46 on page 228. Queue 0 and Queue 2 are rate limited, so the maximum bandwidth they are offered is the transmit bandwidth and they will not be offered any excess bandwidth. All other queues are offered the guaranteed bandwidth and the excess is shared by the non-rate-limited queues.

Table 46: Sharing with at Least One Strict-High Priority and Rate Limit

Queue	Guaranteed (Transmit) Rate	Rate Limit	Excess Rate	Guaranteed Priority	Excess Priority	Offered Load	Expected Throughput
Q0	20%	Yes	X	strict-high	X	500 Mbps	200 + 0 = 200 Mbps
Q1	10%	No	20%	high	low	500 Mbps	100 + 275 = 375 Mbps
Q2	10%	Yes	10%	low	high	500 Mbps	100 + 0 = 100 Mbps
Q3	5%	No	20%	low	low	500 Mbps	50 + 275 = 325 Mbps

Configuring the Schedulers

The following example configures schedulers, forwarding classes, and a scheduler map for an interface with excess rates and excess priorities.

```
[edit class-of-service schedulers]
scheduler-1 {
   transmit-rate percent 20;
   priority high;
   excess-rate percent 10;
   excess-priority low;
}
scheduler-2 {
   transmit-rate percent 10;
   priority strict-high;
}
scheduler-3 {
   transmit-rate percent 10;
   priority medium-high;
   excess-rate percent 20;
   excess-priority high;
}
scheduler-4 {
   transmit-rate percent 5;
   priority medium-high;
   excess-rate percent 30;
   excess-priority low;
}
```

Configuring the Forwarding Classes

```
[edit class-of-service]
forwarding-classes {
   class cp_000 queue-num 0;
   class cp_001 queue-num 1;
   class cp_010 queue-num 2;
   class cp_011 queue-num 3;
   class cp_100 queue-num 4;
   class cp_101 queue-num 5;
   class cp_110 queue-num 6;
   class cp_111 queue-num 7;
}
```

Configuring the Scheduler Map

```
[edit class-of-service scheduler-maps]
scheduler-map-1 {
   forwarding-class cp_000 scheduler scheduler-1;
   forwarding-class cp_001 scheduler scheduler-2;
   forwarding-class cp_010 scheduler scheduler-3;
   forwarding-class cp_011 scheduler scheduler-4;
}
```

Applying the Scheduler Map to the Interface

```
[edit interfaces]
ge-1/1/0 {
   scheduler-map scheduler-map-1;
   unit 0 {
      family inet {
         address 192.168.1.2/32;
      }
   }
}
```


CHAPTER 13

Configuring RED Drop Profiles

This topic discusses the following:

RED Drop Profiles Overview

You can configure two parameters to control congestion at the output stage. The first parameter defines the delay-buffer bandwidth, which provides packet buffer space to absorb burst traffic up to the specified duration of delay. Once the specified delay buffer becomes full, packets with 100 percent drop probability are dropped from the head of the buffer. For more information, see "Configuring the Scheduler Buffer Size" on page 148.

The second parameter defines the drop probabilities across the range of delay-buffer occupancy, supporting the random early detection (RED) process. When the number of packets queued is greater than the ability of the router to empty a queue, the queue requires a method for determining which packets to drop from the network. To address this, the Junos OS provides the option of enabling RED on individual queues.

Depending on the drop probabilities, RED might drop many packets long before the buffer becomes full, or it might drop only a few packets even if the buffer is almost full.

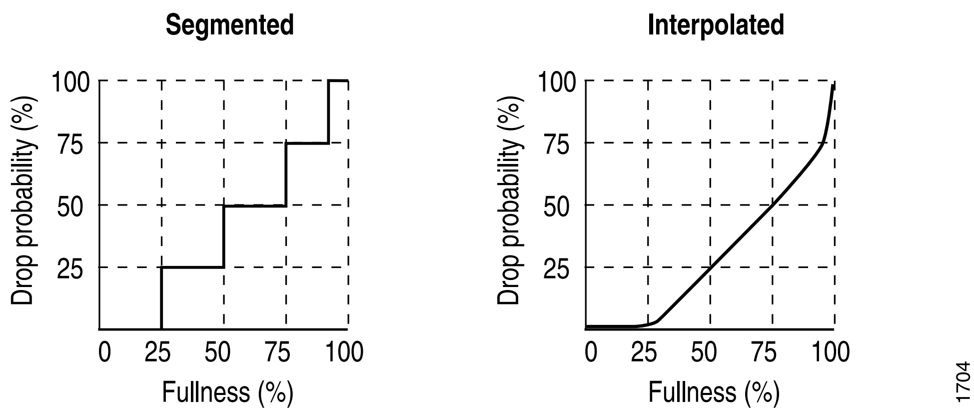
A *drop profile* is a mechanism of RED that defines parameters that allow packets to be dropped from the network. Drop profiles define the meanings of the loss priorities.

When you configure drop profiles, there are two important values: the queue fullness and the drop probability. The *queue fullness* represents a percentage of the memory used to store packets in relation to the total amount that has been allocated for that specific queue. Similarly, 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, as shown in Figure 18 on page 232.

NOTE: You can only specify two fill levels for interpolated drop profiles on the Enhanced Queuing DPC for Juniper Network MX Series Ethernet Services Routers. For more information about interpolated drop profiles on the Enhanced Queuing DPC for MX Series routers, see "Configuring WRED on Enhanced Queuing DPCs" on page 360.

Figure 18 on page 232 shows both a segmented and an interpolated graph. Although the formation of these graph lines is different, the application of the profile is the same. When a packet reaches the head of the queue, a random number between 0 and 100 is calculated by the router. This random number is plotted against the drop profile using the current queue fullness of that particular queue. When the random number falls above the graph line, the packet is transmitted onto the physical media. When the number falls below graph the line, the packet is dropped from the network.

Figure 18: Segmented and Interpolated Drop Profiles



By defining multiple fill levels and drop probabilities, you create a segmented drop profile. The line segments are defined in terms of the following graphical model: in the first quadrant, the x axis represents the fill level, and the y axis represents the drop probability. The initial line segment spans from the origin (0,0) to the point (<l1>, <p1>); a second line runs from (<l1>, <p1>) to (<l2>, <p2>) and so forth, until a final line segment connects (100, 100). The software automatically constructs a drop profile containing 64 fill levels at drop probabilities that approximate the calculated line segments.

NOTE: If you configure the **interpolate** statement, you can specify more than 64 pairs, but the system generates only 64 discrete entries.

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 protocol.

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:

```
[edit class-of-service]
drop-profiles {
  profile-name {
    fill-level percentage drop-probability percentage;
    interpolate {
      drop-probability [ values ];
      fill-level [ values ];
    }
  }
}
```

Default Drop Profile

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.

As a backup method for managing congestion, tail dropping takes effect when congestion of small packets occurs. On Juniper Networks M320 Multiservice Edge Routers and T Series Core Routers, the software supports *tail-RED*, which means that when tail dropping occurs, the software uses RED to execute intelligent tail drops. On other routers, the software executes tail drops unconditionally.

Configuring RED Drop Profiles

You enable RED by applying a drop profile to a scheduler. When RED is operational on an interface, the queue no longer drops packets from the tail of the queue. Rather, packets are dropped after they reach the head of the queue.

To configure a drop profile, include the **drop-profiles** statement at the **[edit class-of-service]** hierarchy level:

```
[edit class-of-service]
drop-profiles {
  profile-name {
    fill-level percentage drop-probability percentage;
    interpolate {
      drop-probability [ values ];
      fill-level [ values ];
    }
  }
}
```

In this configuration, include either the **interpolate** statement and its options, or the fill-level and drop-probability ***percentage*** values. These two alternatives enable you to configure either each drop probability at up to 64 fill-level/drop-probability paired values, or a profile represented as a series of line segments, as discussed in “RED Drop Profiles Overview” on page 231.

After you configure a drop profile, you must assign the drop profile to a drop-profile map, and assign the drop-profile map to a scheduler, as discussed in “Configuring Drop Profile Maps for Schedulers” on page 159.

Packet Loss Priority Configuration Overview

Loss priority settings help determine which packets are dropped from the network during periods of congestion. The software supports multiple packet loss priority (PLP) designations: **low** and **high**. (In addition, **medium-low** and **medium-high** PLPs are supported when you configure tricolor marking, as discussed in “Configuring Tricolor Marking” on page 89.) You can set PLP by configuring a behavior aggregate or multifield classifier, as discussed in “Setting Packet Loss Priority” on page 60 and “Configuring Multifield Classifiers” on page 72.

NOTE: On T Series routers with different Packet Forwarding Engines (non-Enhanced Scaling and Enhanced Scaling FPCs), you can configure PLP bit copying for ingress and egress unicast and multicast traffic. To configure, include the **copy-plp-all** statement at the **[edit class-of-service]** hierarchy level.

A drop-profile map examines the loss priority setting of an outgoing packet: **high**, **medium-high**, **medium-low**, **low**, or any.

Obviously, *low*, *medium-low*, *medium-high*, and *high* are relative terms, which by themselves have no meaning. Drop profiles define the meanings of the loss priorities. In the following example, the **low-drop** drop profile defines the meaning of **low** PLP as a 10 percent drop probability when the fill level is 75 percent and a 40 percent drop probability when the fill level is 95 percent. The **high-drop** drop profile defines the meaning of **high** PLP as a 50 percent drop probability when the fill level is 25 percent and a 90 percent drop probability when the fill level is 50 percent.

In this example, the scheduler includes two drop-profile maps, which specify that packets are evaluated by the **low-drop** drop profile if they have a **low** loss priority and are from any protocol. Packets are evaluated by the **high-drop** drop profile if they have a **high** loss priority and are from any protocol.

```
[edit class-of-service]
drop-profiles {
  low-drop {
    interpolate {
      drop-probability [ 10 40];
      fill-level [ 75 95];
    }
  }
  high-drop {
    interpolate {
      drop-probability [ 50 90];
      fill-level [ 25 50];
    }
  }
```



```
}
schedulers {
  best-effort {
    drop-profile-map loss-priority low protocol any drop-profile low-drop;
    drop-profile-map loss-priority high protocol any drop-profile high-drop;
  }
}
```

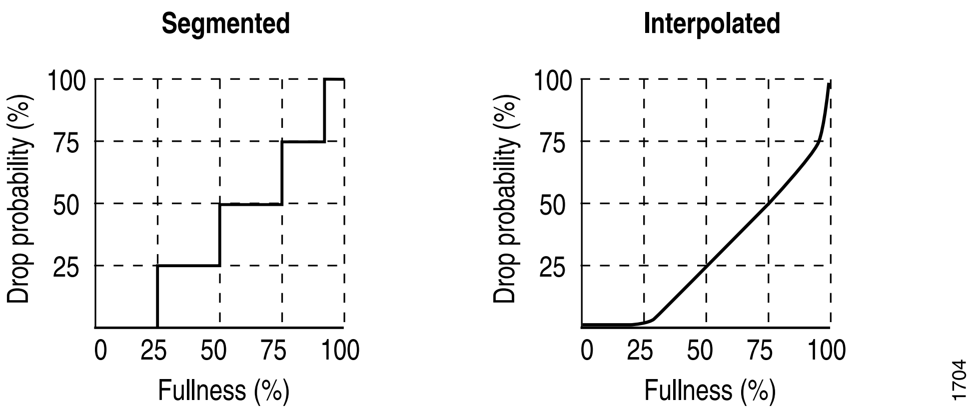
Related Documentation

- Configuring Schedulers on page 148
- Setting Packet Loss Priority on page 60
- Configuring Multifield Classifiers on page 72

Example: Configuring RED Drop Profiles

Create a segmented configuration and an interpolated configuration that correspond to the graphs in Figure 19 on page 235. 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 19: Segmented and Interpolated Drop Profiles



Creating a Segmented Configuration

```
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:

Creating an Interpolated Configuration

```
class-of-service {
  drop-profiles {
    interpolated-style-profile {
      interpolate {
        fill-level [ 50 75 ];
        drop-probability [ 25 50 ];
      }
    }
  }
}
```

Configuring Weighted RED Buffer Occupancy

By default, RED is performed based on instantaneous buffer occupancy information. However, IQ-PICs can be configured to use *weighted average* buffer occupancy information. This option is configured on a per-PIC basis and applies to the following IQ-PICs:

- Channelized T1/T3
- Channelized E1/E3
- Channelized OC3/STM1
- Channelized OC12

If you configure this feature on an unsupported PIC, you see an error message.

When weighted average buffer occupancy is configured, you configure a weight value for averaged buffer occupancy calculations. This weight value is expressed as a negative exponential value of 2 in a fractional expression. For example, a configured weight value of 2 would be expressed as $1/(2^2) = 1/4$. If a configured weight value was configured as 1 (the default), the value would be expressed as $1/(2^1) = 1/2$.

This calculated weight value is applied to the instantaneous buffer occupancy value to determine the new value of the weighted average buffer occupancy. The formula to derive the new weighted average buffer occupancy is:

new average buffer occupancy = weight value * instantaneous buffer occupancy + (1 – weight value) * current average buffer occupancy

For example, if the weight exponent value is configured as 3 (giving a weight value of $1/2^3 = 1/8$), the formula used to determine the new average buffer occupancy based on the instant buffer usage is:

new average buffer occupancy = 1/8 * instantaneous buffer occupancy + (7/8) * current average buffer occupancy

The valid operational range for the weight value on IQ-PICs is 0 through 31. A value of 0 results in the average buffer occupancy being the same as the instantaneous buffer occupancy calculations. Values higher than 31 can be configured, but in these cases the current maximum *operational* value of 31 is used for buffer occupancy calculations.

NOTE: The **show interfaces** command with the **extensive** option displays the *configured* value for the **RED buffer occupancy** weight exponent. However, in all such cases, the current *operational* maximum value of 31 is used internally.

To configure a Q-PIC for RED weighted average buffer occupancy calculations, include the **red-buffer-occupancy** statement with the **weighted-averaged** option at the **[edit chassis fpc *slot-number* pic *pic-number*]** hierarchy level:

```
[edit chassis]
fpc slot-number {
  pic pic-number {
    red-buffer-occupancy {
      weighted-averaged [ instant-usage-weight-exponent ] weight-value;
    }
  }
}
```

Example: Configuring Weighted RED Buffer Occupancy

Configure the Q-PIC to use a weight value of 1/2 in average buffer occupancy calculations.

```
[edit chassis]
fpc 0 {
  pic 1 {
    red-buffer-occupancy {
      weighted-averaged instant-usage-weight-exponent 1;
    }
  }
}
```

or

```
[edit chassis]
fpc 0 {
  pic 1 {
    red-buffer-occupancy {
      weighted-averaged; # the default value is 1 if not specified
    }
  }
}
```

Configure the Q-PIC to use a weight value of 1/4 in average buffer occupancy calculations.

```
[edit chassis]
fpc 0 {
   pic 1 {
      red-buffer-occupancy {
         weighted-averaged instant-usage-weight-exponent 2;
      }
   }
}
```


CHAPTER 14

Rewriting Packet Header Information

This topic discusses the following:

Rewriting Packet Header Information Overview

As packets enter or exit a network, edge routers might be required to alter the class-of-service (CoS) settings of the packets. Rewrite rules set the value of the CoS bits within the packet's header. Each rewrite rule reads the current forwarding class and loss priority information associated with the packet, locates the chosen CoS value from a table, and writes this CoS value into the packet header.

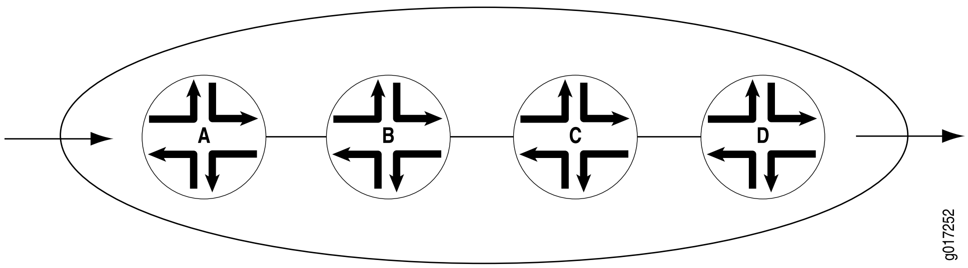
In effect, the rewrite rule performs the opposite function of the behavior aggregate (BA) classifier used when the packet enters the router. As the packet leaves the routing platform, the final CoS action is generally the application of a rewrite rule.

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

In addition, you often need to rewrite a given marker (IP precedence, Differentiated Services code point [DSCP], IEEE 802.1p, or MPLS EXP settings) at the inbound interfaces of an edge router to accommodate BA classification by core devices.

Figure 20 on page 240 shows a flow of packets through four routers. Router A rewrites the CoS bits in incoming packet to accommodate the BA classification performed by Routers B and C. Router D alters the CoS bits of the packets before transmitting them to the neighboring network.

Figure 20: Packet Flow Across the Network



To configure CoS rewrite rules, you define the rewrite rule and apply it to an interface. Include the following statements at the **[edit class-of-service]** hierarchy level:

```
[edit class-of-service]
interfaces {
  interface-name {
    unit logical-unit-number {
      rewrite-rules {
        dscp (rewrite-name | default)protocol protocol-types;
        dscp-ipv6 (rewrite-name | default);
        exp (rewrite-name | default)protocol protocol-types;
        exp-push-push-push default;
        exp-swap-push-push default;
        ieee-802.1 (rewrite-name | default) vlan-tag (outer | outer-and-inner);
        ieee-802.1ad (rewrite-name | default) vlan-tag (outer | outer-and-inner);
        inet-precedence (rewrite-name | default)protocol protocol-types;
      }
    }
  }
}
rewrite-rules {
  (dscp | dscp-ipv6 | exp | frame-relay-de | ieee-802.1 | inet-precedence) rewrite-name {
    import (rewrite-name | default);
    forwarding-class class-name {
      loss-priority level code-point (alias | bits);
    }
  }
}
```

Applying Default Rewrite Rules

By default, rewrite rules are not usually applied to interfaces. The exceptions are MPLS interfaces: all MPLS-enabled interfaces use the default EXP rewrite rule, even if not configured. Except for MPLS interfaces, if you want to apply a rewrite rule, you can either design your own rule and apply it to an interface, or you can apply a default rewrite rule. To apply default rewrite rules, include one or more of the following statements at the **[edit class-of-service interfaces *interface-name* unit *logical-unit-number* rewrite-rules]** hierarchy level:

```
[edit class-of-service interfaces interface-name unit logical-unit-number rewrite-rules]
dscp default;
dscp-ipv6 default;
exp default;
ieee-802.1 default vlan-tag (outer | outer-and-inner);
inet-precedence default;
```

Table 47 on page 241 shows the default rewrite rule mappings. These are based on the default bit definitions of DSCP, DSCP IPv6, EXP, IEEE, and IP CoS values, as shown in "Default CoS Values" on page 66, and the default forwarding classes shown in "Default Forwarding Classes" on page 112.

When the software detects packets whose CoS values match the forwarding class and PLP values in the first two columns in Table 47 on page 241, the software maps the header bits of those packets to the code-point aliases in the last column in Table 47 on page 241. The code-point aliases in the last column map to the CoS bits shown in "Default CoS Values" on page 66.

Table 47: Default Packet Header Rewrite Mappings

Map from Forwarding Class	PLP Value	Map to DSCP/DSCP IPv6/ EXP/IEEE/IP
expedited-forwarding	low	ef
expedited-forwarding	high	ef
assured-forwarding	low	af11
assured-forwarding	high	af12 (DSCP/DSCP IPv6/EXP)
best-effort	low	be
best-effort	high	be
network-control	low	nc1/cs6
network-control	high	nc2/cs7

In the following example, the **so-1/2/3.0** interface is assigned the default DSCP rewrite rule. One result of this configuration is that each packet exiting the interface with the **expedited-forwarding** forwarding class and the **high** or **low** loss priority has its DSCP bits rewritten to the DSCP **ef** code-point alias. "Default CoS Values" on page 66 shows that this code-point alias maps to the **101110** bits.

NOTE: DSCP rewrite is not supported on the 10-port 10-Gigabit Oversubscribed Ethernet PIC.

Another result of this configuration is that all packets exiting the interface with the **best-effort** forwarding class and the **high** or **low** loss priority have their EXP bits rewritten to the EXP **be** code-point alias. "Default CoS Values" on page 66 shows that this code-point alias maps to the **000** bits.

To evaluate all the implications of this example, see "Default CoS Values" on page 66 and Table 47 on page 241.

```
class-of-service {
  interfaces {
    so-1/2/3 {
      unit 0 {
        rewrite-rules {
          dscp default;
        }
      }
    }
  }
}
```

Configuring Rewrite Rules

You define markers in the rewrite rules section of the CoS configuration hierarchy and reference them in the logical interface configuration. This model supports marking on the DSCP, DSCP IPv6, IP precedence, IEEE 802.1, and MPLS EXP CoS values.

To configure a rewrite-rules mapping and associate it with the appropriate forwarding class and code-point alias or bit set, include the **rewrite-rules** statement at the **[edit class-of-service]** hierarchy level:

```
[edit class-of-service]
rewrite-rules {
  (dscp | dscp-ipv6 | exp | ieee-802.1 | inet-precedence) rewrite-name {
    import (rewrite-name | default);
    forwarding-class class-name {
      loss-priority level code-point (alias | bits);
    }
  }
}
```

The rewrite rule sets the code-point aliases and bit patterns for a specific forwarding class and PLP. The inputs for the map are the forwarding class and the PLP. The output of the map is the code-point alias or bit pattern. For more information about how CoS maps work, see "CoS Inputs and Outputs Overview" on page 9.

By default, IP precedence rewrite rules alter the first three bits on the type-of-service (ToS) byte while leaving the last three bits unchanged. This default behavior is not configurable. The default behavior applies to rules you configure by including the **inet-precedence** statement at the **[edit class-of-service rewrite-rules]** hierarchy level. The default behavior also applies to rewrite rules you configure for MPLS packets with IPv4 payloads. You configure these types of rewrite rules by including the **mpls-inet-both** or **mpls-inet-both-non-vpn** option at the **[edit class-of-service interfaces *interface-name* unit *logical-unit-number* rewrite-rules exp *rewrite-rule-name* protocol]** hierarchy level.

On the M320, T1600, and MX960 routers, if you configure **vlan-vpls** encapsulation and add an IEEE 802.1 header on a Gigabit Ethernet or 10 Gigabit Ethernet interface to output traffic, but do not apply an IEEE 802.1 rewrite rule, then the default IEEE 802.1 rewrite rule is ignored and the IEEE 802.1p bits are set to match the leftmost three bits of the DSCP field in the IP header. This occurs even when the packet is transported inside an MPLS LSP.

Header Bits Preserved, Cleared, and Rewritten

For every incoming packet, the ingress classifier decodes the ingress CoS bits into a forwarding class and packet loss priority (PLP) combination.

The egress CoS information depends on which type of rewrite marker is active, as follows:

- For Multiprotocol Label Switching (MPLS) EXP and IEEE 802.1 rewrite markers, values are derived from the forwarding class and PLP values in rewrite rules. MPLS EXP and IEEE 802.1 markers are not preserved because they are part of the Layer 2 encapsulation.
- For IP precedence and DiffServ code point (DSCP) rewrite markers, the marker alters the first three bits on the type-of-service (ToS) byte while leaving the last three bits unchanged.

Applying Rewrite Rules to Output Logical Interfaces

To assign the rewrite-rules configuration to the output logical interface, include the **rewrite-rules** 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]
rewrite-rules {
  dscp (rewrite-name | default)protocol protocol-types;
  dscp-ipv6 (rewrite-name | default);
  exp (rewrite-name | default)protocol protocol-types;
  exp-push-push-push default;
  exp-swap-push-push default;
  ieee-802.1 (rewrite-name | default) inet-prec vlan-tag (outer | outer-and-inner);
  inet-precedence (rewrite-name | default)protocol protocol-types;
}
```

On M120, M320 routers with an Enhanced III FPC, and MX Series routers, you can combine the **dscp** or **inet-prec** and **exp** options to set the DSCP or IP precedence bits and MPLS EXP bits independently on IP packets entering an MPLS tunnel.

For IQ PICs, you can only configure one IEEE 802.1 rewrite rule on a physical port. All logical ports (units) on that physical port should apply the same IEEE 802.1 rewrite rule. if you configure more than one IEEE 802.1 rewrite rule for the IQ PIC, the configuration check fails.

The following example sets the DSCP bits to the bit configuration specified in **zf-dscp** on packets entering the MPLS tunnel on **so-0/0/1** and sets the EXP bits to the bit configuration specified in **zf-exp**:

```
[edit class-of-service interfaces]
so-0/0/1 {
  unit 0 {
    rewrite-rules {
      dscp zf-dscp protocol mpls; # Applies to IPv4 packets entering MPLS tunnel
      exp zf-exp; # Sets label EXP bits independently
    }
  }
}
```

You can use interface wildcards for ***interface-name*** and ***logical-unit-number***. You can also include Layer 2 and Layer 3 rewrite information in the same configuration.

NOTE: On M Series routers only, if you include the **control-word** statement at the **[edit protocols l2circuit neighbor *address* interface *interface-name*]** hierarchy level, the software cannot rewrite MPLS EXP bits.

DSCP and DSCP IPv6 rewrite rules are supported on M Series and T Series routers when non-queuing PICs are installed, but are disabled when queuing PICs are installed with the following exception:

- On M320 routers, DSCP rewrite is supported on IQ, IQ2, IQE, and IQ2E PICs when used with the Enhanced III FPC.

DSCP rewrite rules are not supported on T Series routers when IQ, IQ2, IQE, IQ2E, or PD-5-10XGE-SFPP PICs are installed.

For IQ PICs, you can only configure one IEEE 802.1 rewrite rule on a physical port. All logical ports (units) on that physical port should apply the same IEEE 802.1 rewrite rule.

On M320 routers and T Series routers, for a single interface, you cannot enable a rewrite rule on a subset of forwarding classes. You must assign a rewrite rule to either none of the forwarding classes or all of the forwarding classes. When you assign a rewrite rule to a subset of forwarding classes, the commit does not fail, and the subset of forwarding classes work as expected. However, the forwarding classes to which the rewrite rule is not assigned are rewritten to all zeros.

For example, if you configure a Differentiated Services code point (DSCP) rewrite rule, the bits in the forwarding classes to which you do not assign the rewrite rule are rewritten to 000000; if you configure an IP precedence rewrite rule, the bits in the forwarding classes to which you do not assign the rewrite rule are rewritten to 000.

Applying IEEE 802.1p Rewrite Rules to Dual VLAN Tags

By default, when you apply an IEEE 802.1p rewrite rule to an output logical interface, the software rewrites the IEEE bits in the outer VLAN tag only.

For Gigabit Ethernet IQ2 PICs, 10-port 10-Gigabit OSE PICs, and 10-Gigabit Ethernet IQ2 PICs only, you can rewrite the IEEE bits in both the outer and inner VLAN tags of the tagged Ethernet frames. When you enable class of service (CoS) rewrite for both tags, the same IEEE 802.1p rewrite table is used for the inner and outer VLAN tag.

For IQ PICs, you can only configure one IEEE 802.1 rewrite rule on a physical port. All logical ports (units) on that physical port should apply the same IEEE 802.1 rewrite rule.

To rewrite both the outer and inner VLAN tags, include the **vlan-tag outer-and-inner** statement at the **[edit class-of-service interfaces *interface-name* unit *logical-unit-number* rewrite-rules ieee-802.1 (*rewrite-name* | default)]** hierarchy level:

```
[edit class-of-service interfaces interface-name unit logical-unit-number rewrite-rules
  ieee-802.1 (rewrite-name | default)]
vlan-tag outer-and-inner;
```

To explicitly specify the default behavior, include the **vlan-tag outer** statement at the **[edit class-of-service interfaces *interface-name* unit *logical-unit-number* rewrite-rules ieee-802.1 (*rewrite-name* | default)]** hierarchy level:

```
[edit class-of-service interfaces interface-name unit logical-unit-number rewrite-rules
  ieee-802.1 (rewrite-name | default)]
vlan-tag outer;
```

For more information about VLAN tags, see the *Junos OS Network Interfaces Configuration Guide*.

On MX routers, you can perform IEEE 802.1p and DEI rewriting based on forwarding class and PLP at the VPLS ingress PE. You rewrite (mark) the IEEE 802.1p or DEI bits on frames at the VPLS ingress PE based on the value of the forwarding class and PLP established for the traffic. You can rewrite either the outer tag only or the outer and inner tag. When both tags are rewritten, both get the same value. To configure these rewrite rules, include the **ieee-802.1** statement at the [**edit class-of-services routing-instance *routing-instance-name* rewrite-rules**] hierarchy level.

On routers with IQ2 or IQ2-E PICs, you can perform IEEE 802.1p and DEI rewriting based on forwarding-class and packet loss priority (PLP) at the VPLS ingress provider edge (PE) router. You rewrite (mark) the IEEE 802.1p or DEI bits on frames at the VPLS ingress PE based on the value of the forwarding-class and PLP established for the traffic. You can rewrite either the outer tag only or both the outer and inner tags. When both tags are rewritten, both get the same value.

NOTE: The 10-port 10-Gigabit OSE PIC does not support DEI rewriting based on forwarding class and PLP at the VPLS ingress PE.

To configure these rewrite rules, include the **ieee-802.1** statement at the [**edit class-of-services routing-instance *routing-instance-name* rewrite-rules**] hierarchy level.

Example: Applying an IEEE 802.1p Rewrite Rule to Dual VLAN Tags

Apply the **ieee8021p-rwrule1** rewrite rule to both inner and outer VLAN tags of Ethernet-tagged frames exiting the **ge-0/0/0.0** interface:

```
class-of-service {
  interfaces {
    ge-0/0/0 {
      unit 0 {
        rewrite-rules {
          ieee-802.1 ieee8021p-rwrule1 vlan-tag outer-and-inner;
        }
      }
    }
  }
}
```

Applying IEEE 802.1ad Rewrite Rules to Dual VLAN Tags

By default, when you apply an IEEE 802.1ad rewrite rule to an output logical interface, the software rewrites the IEEE bits in the outer VLAN tag only.

For MX Series routers and IQ2 PICs, you can rewrite the IEEE 802.1ad bits in both the outer and inner VLAN tags of the tagged Ethernet frames. When you enable the CoS rewrite for both tags, the same IEEE 802.1ad rewrite table is used for the inner and outer VLAN tag.

To rewrite both the outer and inner VLAN tags, include the **vlan-tag outer-and-inner** statement at the **[edit class-of-service interfaces *interface-name* unit *logical-unit-number* rewrite-rules ieee-802.1ad (*rewrite-name* | default)]** hierarchy level:

```
[edit class-of-service interfaces interface-name unit logical-unit-number rewrite-rules
  ieee-802.1ad (rewrite-name | default)]
vlan-tag outer-and-inner;
```

To explicitly specify the default behavior, include the **vlan-tag outer** statement at the **[edit class-of-service interfaces *interface-name* unit *logical-unit-number* rewrite-rules ieee-802.1ad (*rewrite-name* | default)]** hierarchy level:

```
[edit class-of-service interfaces interface-name unit logical-unit-number rewrite-rules
  ieee-802.1ad (rewrite-name | default)]
vlan-tag outer;
```

For more information about VLAN tags, see the *Junos OS Network Interfaces Configuration Guide*.

Example: Applying an IEEE 802.1ad Rewrite Rule to Dual VLAN Tags

Apply the **dot1p_dei_rw** rewrite rule to both inner and outer VLAN tags of Ethernet-tagged frames exiting the **ge-2/0/0.0** interface:

```
class-of-service {
```

```
  interfaces {
    ge-2/0/0 {
      unit 0 {
        rewrite-rules {
          ieee-802.1ad dot1p_dei_rw vlan-tag outer-and-inner;
        }
      }
    }
  }
}
```

Example: Per-Node Rewriting of EXP Bits

To configure a custom table to rewrite the EXP bits, also known as CoS bits, on a particular node, the classifier table and the rewrite table must specify exactly the same CoS values.

In addition, the least significant bit of the CoS value itself must represent the PLP value. For example, CoS value **000** must be associated with PLP **low**, **001** must be associated with PLP **high**, and so forth.

This example configures a custom table to rewrite the EXP bits on a particular node:

```
[edit class-of-service]
classifiers {
  exp exp-class {
    forwarding-class be {
      loss-priority low code-points 000;
      loss-priority high code-points 001;
    }
    forwarding-class af {
      loss-priority low code-points 010;
      loss-priority high code-points 011;
    }
    forwarding-class ef {
      loss-priority low code-points 100;
      loss-priority high code-points 101;
    }
    forwarding-class nc {
      loss-priority low code-points 110;
      loss-priority high code-points 111;
    }
  }
}
rewrite-rules {
  exp exp-rw {
    forwarding-class be {
      loss-priority low code-point 000;
      loss-priority high code-point 001;
    }
    forwarding-class af {
      loss-priority low code-point 010;
      loss-priority high code-point 011;
    }
    forwarding-class ef {
      loss-priority low code-point 100;
```

```
            loss-priority high code-point 101;
        }
        forwarding-class nc {
            loss-priority low code-point 110;
            loss-priority high code-point 111;
        }
    }
}
```

Rewriting MPLS and IPv4 Packet Headers

You can apply a rewrite rule to MPLS and IPv4 packet headers simultaneously. This allows you to initialize MPLS EXP and IP precedence bits at LSP ingress. You can configure different rewrite rules depending on whether the traffic is VPN or non-VPN.

The default MPLS EXP rewrite table contents are shown in Table 48 on page 248.

Table 48: Default MPLS EXP Rewrite Table

Forwarding Class	Loss Priority	CoS Value
best-effort	low	000
best-effort	high	001
expedited-forwarding	low	010
expedited-forwarding	high	011
assured-forwarding	low	100
assured-forwarding	high	101
network-control	low	110
network-control	high	111

By default, IP precedence rewrite rules alter the first three bits on the type-of-service (ToS) byte while leaving the last three bits unchanged. This default behavior applies to rewrite rules you configure for MPLS packets with IPv4 payloads.

To override the default MPLS EXP rewrite table and rewrite MPLS and IPv4 packet headers simultaneously, include the **protocol** statement at the **[edit class-of-service interfaces *interface-name* unit *logical-unit-number* rewrite-rules exp *rewrite-rule-name*]** hierarchy level:

[edit class-of-service interfaces *interface-name* unit *logical-unit-number* rewrite-rules exp *rewrite-rule-name*]
protocol *protocol-types*;

The **protocol** statement defines the types of MPLS packets and packet headers to which the specified rewrite rule is applied. The MPLS packet can be a standard MPLS packet or an MPLS packet with an IPv4 payload. Specify the type of MPLS packet using the following options:

- **mpls**—Applies the rewrite rule to MPLS packets and writes the CoS value to MPLS headers.
- **mpls-inet-both**—Applies the rewrite rule to VPN MPLS packets with IPv4 payloads. On Juniper Networks M120 Multiservice Edge Routers, M320 Multiservice Edge Routers, and T Series Core Routers, writes the CoS value to the MPLS and IPv4 headers. On other M Series Multiservice Edge Router routers, causes all ingress MPLS LSP packets with IPv4 payloads to be initialized with **000** code points for the MPLS EXP value, and the configured rewrite code point for IP precedence.
- **mpls-inet-both-non-vpn**—Applies the rewrite rule to non-VPN MPLS packets with IPv4 payloads. On Juniper Networks M120 Multiservice Edge Routers, M320 Multiservice Edge Routers, and T Series Core Routers, writes the CoS value to the MPLS and IPv4 headers. On other M Series Multiservice Edge Routers, causes all ingress MPLS LSP packets with IPv4 payloads to be initialized with **000** code points for the MPLS EXP value, and the configured rewrite code point for IP precedence.

An alternative to overwriting the default with a rewrite-rules mapping is to configure the default packet header rewrite mappings, as discussed in “Applying Default Rewrite Rules” on page 240.

By default, IP precedence rewrite rules alter the first three bits on the ToS byte while leaving the last three bits unchanged. This default behavior is not configurable. The default behavior applies to rules you configure by including the **inet-precedence** statement at the **[edit class-of-service rewrite-rules]** hierarchy level. The default behavior also applies to rewrite rules you configure for MPLS packets with IPv4 payloads. You configure these types of rewrite rules by including the **mpls-inet-both** or **mpls-inet-both-non-vpn** option at the **[edit class-of-service interfaces *interface-name* unit *logical-unit-number* rewrite-rules exp *rewrite-rule-name* protocol]** hierarchy level.

Example: Rewriting MPLS and IPv4 Packet Headers

On M320 and T Series routers, configure rewrite tables and apply them in various ways to achieve the following results:

- For interface **so-3/1/0**, the three EXP rewrite tables are applied to packets, depending on the protocol of the payload:
  - IPv4 packets (VPN) that enter the LSPs on interface **so-3/1/0** are initialized with values from rewrite table **exp-inet-table**. An identical 3-bit value is written into the IP precedence and MPLS EXP bit fields.
  - IPv4 packets (non-VPN) that enter the LSPs on interface **so-3/1/0** are initialized with values from rewrite table **rule-non-vpn**. An identical 3-bit value is written into the IP precedence and MPLS EXP bit fields.

- Non-IPv4 packets that enter the LSPs on interface **so-3/1/0** are initialized with values from rewrite table **rule1**, and written into the MPLS EXP header field only. The statement **exp rule1** has the same result as **exp rule1 protocol mpls**.

- For interface **so-3/1/0**, IPv4 packets transmitted over a non-LSP layer are initialized with values from IP precedence rewrite table **rule2**.
- For interface **so-3/1/1**, IPv4 packets that enter the LSPs are initialized with values from EXP rewrite table **exp-inet-table**. An identical 3-bit value is written into the IP precedence and MPLS EXP bit fields.
- For interface **so-3/1/1**, MPLS packets other than IPv4 Layer 3 types are also initialized with values from table **exp-inet-table**. For VPN MPLS packets with IPv4 payloads, the CoS value is written to MPLS and IPv4 headers. For VPN MPLS packets without IPv4 payloads, the CoS value is written to MPLS headers only.

```
[edit class-of-service]
rewrite-rules {
  exp exp-inet-table {
    forwarding-class best-effort {
      loss-priority low code-point 000;
      loss-priority high code-point 001;
    }
    forwarding-class assured-forwarding {
      loss-priority low code-point 010;
      loss-priority high code-point 011;
    }
    forwarding-class expedited-forwarding {
      loss-priority low code-point 111;
      loss-priority high code-point 110;
    }
    forwarding-class network-control {
      loss-priority low code-point 100;
      loss-priority high code-point 101;
    }
  }
  exp rule1 {
    ...
  }
  inet-precedence rule2 {
    ...
  }
}
exp rule_non_vpn {
  ...
}

interfaces {
  so-3/1/0 {
    unit 0 {
      rewrite-rules {
        exp rule1;
        inet-precedence rule2;
        exp exp-inet-table protocol mpls-inet-both; # For all VPN traffic.
        exp rule_non_vpn protocol mpls-inet-both-non-vpn; # For all non-VPN
```



```
                # traffic.
            }
        }
    }
    so-3/1/1 {
        unit 0 {
            rewrite-rules {
                exp exp-inet-table protocol [mpls mpls-inet-both];
            }
        }
    }
}
```

Rewriting the EXP Bits of All Three Labels of an Outgoing Packet

In interprovider, carrier-of-carrier, and complex traffic engineering scenarios, it is sometimes necessary to push three labels on the next hop, using a swap-push-push or triple-push operation.

By default, on M Series routers, the top MPLS EXP label of an outgoing packet is not rewritten when you configure swap-push-push and triple-push operations. On M Series routers, you can rewrite the EXP bits of all three labels of an outgoing packet, thereby maintaining the CoS of an incoming MPLS or non-MPLS packet.

When the software performs a swap-push-push operation and no rewriting is configured, the EXP fields of all three labels are the same as in the old label. If there is EXP rewriting configured, the EXP bits of the bottom two labels are overwritten with the table entry. The EXP setting of the top label is retained even with rewriting.

To push three labels on all incoming MPLS packets, include the **exp-swap-push-push default** statement at the **[edit class-of-service interfaces *interface-name* unit *logical-unit-number* rewrite-rules]** hierarchy level:

[edit class-of-service interfaces *interface-name* unit *logical-unit-number* rewrite-rules]
exp-swap-push-push default;

When the software performs a push-push-push operation and if no rewriting is configured, the EXP fields of the bottom two labels are zero. If EXP rewriting is configured, the EXP fields of the bottom two labels are rewritten with the table entry's rewrite value. The EXP field of the top label is inserted with the Qn+PLP value. This Qn reflects the final classification by a multifield classifier if one exists, regardless of whether rewriting is configured.

NOTE: The **exp-push-push-push** and **exp-swap-push-push** configuration on the egress interface does not rewrite the top label's EXP field with the Qn+PLP value on an IQ or IQ2 PIC.

To push three labels on incoming non-MPLS packets, include the **exp-push-push-push default** statement at the **[edit class-of-service interfaces *interface-name* unit *logical-unit-number* rewrite-rules]** hierarchy level:

[edit class-of-service interfaces *interface-name* unit *logical-unit-number* **rewrite-rules**]
exp-push-push-push default;

These configurations apply the default MPLS EXP rewrite table, as described in "Rewriting MPLS and IPv4 Packet Headers" on page 248. You can configure these operations and override the default MPLS EXP rewrite table with a custom table. For more information about writing and applying a custom rewrite table, see "Configuring Rewrite Rules" on page 242 and "Applying Rewrite Rules to Output Logical Interfaces" on page 243.

NOTE: With a three-label stack, if you do not include the **exp-swap-push-push default** or **exp-push-push-push default** statement in the configuration, the top label's EXP bits are set to zero.

Example: Rewriting the EXP Bits of All Three Labels of an Outgoing Packet

Configure a swap-push-push operation, and override the default rewrite table with a custom table:

```
[edit class-of-service]
forwarding-classes {
   queue 0 be;
   queue 1 ef;
   queue 2 af;
   queue 3 nc;
}
interfaces {
   so-1/1/3 {
      unit 0 {
         rewrite-rules {
            exp exp_rew; # Apply custom rewrite table
            exp-swap-push-push default;
         }
      }
   }
}
rewrite-rules {
   exp exp_rew {
      forwarding-class be {
         loss-priority low code-point 000;
         loss-priority high code-point 100;
      }
      forwarding-class ef {
         loss-priority low code-point 001;
         loss-priority high code-point 101;
      }
      forwarding-class af {
         loss-priority low code-point 010;
         loss-priority high code-point 110;
      }
      forwarding-class nc {
         loss-priority low code-point 011;
         loss-priority high code-point 111;
      }
```

```
    }
  }
```

Rewriting IEEE 802.1p Packet Headers with an MPLS EXP Value

For Ethernet interfaces on Juniper Networks M320 Multiservice Edge Routers, MX Series Ethernet Service Routers, and T Series Core Routers that have a peer connection to an M Series Multiservice Edge Router, MX Series, or T Series router, you can rewrite both MPLS EXP and IEEE 802.1p bits to a configured value. This enables you to pass the configured value to the Layer 2 VLAN path. For IQ PICs, you can only configure one IEEE 802.1 rewrite rule on a physical port. All logical ports (units) on that physical port should apply the same IEEE 802.1 rewrite rule.

To rewrite both the MPLS EXP and IEEE 802.1p bits, you must include EXP and IEEE 802.1p rewrite rules in the interface configuration. To configure EXP and IEEE 802.1p rewrite rules, include the **rewrite-rules** statement at the **[edit class-of-service interfaces *interface-name* unit *logical-unit-number*]** hierarchy level, specifying the **exp** and **ieee-802.1** options:

```
[edit class-of-service interfaces interface-name unit logical-unit-number]
rewrite-rules {
  exp rewrite-rule-name;
  ieee-802.1 default;
}
```

When you combine these two rewrite rules, only the EXP rewrite table is used for rewriting packet headers. If you do not configure a VLAN on the interface, only the EXP rewriting is in effect. If you do not configure an LSP on the interface or if the MPLS EXP rewrite rule mapping is removed, the IEEE 802.1p default rewrite rules mapping takes effect.

NOTE: You can also combine other rewrite rules. IP, DSCP, DSCP IPv6, and MPLS EXP are associated with Layer 3 packet headers, and IEEE 802.1p is associated with Layer 2 packet headers.

For IQ PICs, you can only configure one IEEE 802.1 rewrite rule on a physical port. All logical ports (units) on that physical port should apply the same IEEE 802.1 rewrite rule.

If you combine IEEE 802.1p with IP rewrite rules, the Layer 3 packets and Layer 2 headers are rewritten with the IP rewrite rule.

If you combine IEEE 802.1p with DSCP or DSCP IPv6 rewrite rules, three bits of the Layer 2 header and six bits of the Layer 3 packet header are rewritten with the DSCP or DSCP IPv6 rewrite rule.

The following example shows how to configure an EXP rewrite rule and apply it to both MPLS EXP and IEEE 802.1p bits:

```
[edit class-of-service]
rewrite-rules {
  exp exp-ieee-table {
```



```
            forwarding-class best-effort {
                loss-priority low code-point 000;
                loss-priority high code-point 001;
            }
            forwarding-class assured-forwarding {
                loss-priority low code-point 010;
                loss-priority high code-point 011;
            }
            forwarding-class expedited-forwarding {
                loss-priority low code-point 111;
                loss-priority high code-point 110;
            }
            forwarding-class network-control {
                loss-priority low code-point 100;
                loss-priority high code-point 101;
            }
        }
    }
    interfaces {
        so-3/1/0 {
            unit 0 {
                rewrite-rules {
                    exp exp-ieee-table;
                    ieee-802.1 default;
                }
            }
        }
    }
```

Setting Ingress DSCP Bits for Multicast Traffic over Layer 3 VPNs

By default, the DSCP bits on outer IP headers arriving at an ingress PE router using generic routing encapsulation (GRE) are not set for multicast traffic sent over an Layer 3 virtual private network (VPN) provider network. However, you can configure a type-of-service (ToS) rewrite rule so the router sets the DSCP bits of GRE packets to be consistent with the service provider's overall core network CoS policy. The bits are set at the core-facing interface of the ingress provider edge (PE) router. For more information about rewriting IP header bits, see "Rewriting Packet Header Information Overview" on page 239.

This section describes this configuration from a CoS perspective. The examples are not complete multicast or VPN configurations. For more information about multicast, see the *Junos OS Multicast Protocols Configuration Guide*. For more information about Layer 3 VPNs, see the *Junos OS VPNs Configuration Guide*.

To configure the rewrite rules on the core-facing interface of the ingress PE, include the **rewrite-rules** statement at the **[edit class-of-service]** hierarchy level. You apply the rule to the proper ingress interface at the **[edit class-of-service interfaces]** hierarchy level to complete the configuration. This ingress DSCP rewrite is independent of classifiers placed on ingress traffic arriving on the customer-facing interface of the PE router.

The rewrite rules are applied to all unicast packets and multicast groups. You cannot configure different rewrite rules for different multicast groups. The use of DSCPv6 bits is not supported because IPv6 multicast is not supported. You can configure another rewrite rule for the EXP bits on MPLS CE-CE unicast traffic.

This example defines a rewrite rule called **dscp-rule** that establishes a value of **000000** for best-effort traffic. The rule is applied to the outgoing, core-facing PE interface **ge-2/3/0**.

```
[edit class-of-service]
rewrite-rules {
  dscp dscp-rule {
    forwarding-class best-effort {
      loss-priority low code-point 000000;
    }
  }
}

[edit class-of-service interfaces]
ge-2/3/0 {
  unit 0 {
    rewrite-rules {
      dscp dscp-rule;
    }
  }
}
```

PART 3

CoS Configuration on Various PIC Types

CHAPTER 15

Hardware Capabilities and Routing Engine Protocol Queue Assignments

This topic discusses the following:

Hardware Capabilities and Limitations

Juniper Networks J Series Services Routers, M320 Multiservice Edge Routers, and T Series Core Routers, as well as M Series Multiservice Edge Routers with enhanced Flexible PIC Concentrators (FPCs), have more CoS capabilities than M Series routers that use other FPC models. Table 49 on page 260 lists some of these the differences. Basic MX Series router architecture information is presented in "Packet Flow on MX Series Ethernet Services Routers" on page 13.

To determine whether your M Series router is equipped with an enhanced FPC, issue the **show chassis hardware** command. The presence of an enhanced FPC is designated by the **E-FPC** description in the output.

```
user@host> show chassis hardware
Hardware inventory:
Item              Version  Part number Serial number     Description
Chassis                                31959             M7i
Midplane         REV 02   710-008761   CA0209            M7i Midplane
Power Supply 0   REV 04   740-008537   PD10272           AC Power Supply
Routing Engine   REV 01   740-008846   1000396803        RE-5.0
CFEB             REV 02   750-009492   CA0166            Internet Processor IIv1
FPC 0                                                    E-FPC
  PIC 0          REV 04   750-003163   HJ6416            1x G/E, 1000 BASE-SX
  PIC 1          REV 04   750-003163   HJ6423            1x G/E, 1000 BASE-SX
  PIC 2          REV 04   750-003163   HJ6421            1x G/E, 1000 BASE-SX
  PIC 3          REV 02   750-003163   HJ0425            1x G/E, 1000 BASE-SX
FPC 1                                                    E-FPC
```

```
PIC 2          REV 01    750-009487    HM2275           ASP - Integrated
PIC 3          REV 01    750-009098    CA0142           2x F/E, 100 BASE-TX
```

J Series Services Routers do not use FPCs. Instead, they use Physical Interface Modules (PIMs), which are architecturally like FPCs but functionally like PICs. Both PIMs and PICs provide the interfaces to the routers.

In Table 49 on page 260, the information in the column titled “M320 and T Series FPCs” is valid for all M320 and T Series router FPCs, including Enhanced II FPCs.

Table 49: CoS Hardware Capabilities and Limitations

Feature	J Series PIMs	M Series FPCs	M Series Enhanced FPCs	M320 and T Series FPCs	Comments
Classifiers					
Maximum number per FPC, PIC, or PIM	64	1	8	64	For M Series router FPCs, the one-classifier limit includes the default IP precedence classifier. If you create a new classifier and apply it to an interface, the new classifier does not override the default classifier for other interfaces on the same FPC. In general, the first classifier associated with a logical interface is used. The default classifier can be replaced only when a single interface is associated with the default classifier. For more information, see “Applying Classifiers to Logical Interfaces” on page 50.
dscp	Yes	No	Yes	Yes	On all routers, you cannot configure IP precedence and DiffServ code point (DSCP) classifiers on a single logical interface, because both apply to IPv4 packets. For more information, see “Applying Classifiers to Logical Interfaces” on page 50.
dscp-ipv6	Yes	No	Yes	Yes	For T Series routers, you can apply separate classifiers for IPv4 and IPv6 packets per logical interface. For M Series router enhanced FPCs, you cannot apply separate classifiers for IPv4 and IPv6 packets. Classifier assignment works as follows: • If you assign a DSCP classifier only, IPv4 and IPv6 packets are classified using the DSCP classifier. • If you assign an IP precedence classifier only, IPv4 and IPv6 packets are classified using the IP precedence classifier. The lower three bits of the DSCP field are ignored because IP precedence mapping requires the upper three bits only. • If you assign either the DSCP or the IP precedence classifier in conjunction with the DSCP IPv6 classifier, the commit fails. • If you assign a DSCP IPv6 classifier only, IPv4 and IPv6 packets are classified using the DSCP IPv6 classifier, but the commit displays a warning message. For more information, see “Applying Classifiers to Logical Interfaces” on page 50.

Table 49: CoS Hardware Capabilities and Limitations *(continued)*

Feature	J Series PIMs	M Series FPCs	M Series Enhanced FPCs	M320 and T Series FPCs	Comments
ieee-802.1p	Yes	No	Yes	Yes	On M Series router enhanced FPCs and T Series routers, if you associate an IEEE 802.1p classifier with a logical interface, you cannot associate any other classifier with that logical interface. For more information, see "Applying Classifiers to Logical Interfaces" on page 50. For most PICs, if you apply an IEEE 802.1p classifier to a logical interface, you cannot apply non-IEEE classifiers on other logical interfaces on the same physical interface. This restriction does not apply to Gigabit Ethernet IQ2 PICs.
inet-precedence	Yes	Yes	Yes	Yes	On all routers, you cannot assign IP precedence and DSCP classifiers to a single logical interface, because both apply to IPv4 packets. For more information, see "Applying Classifiers to Logical Interfaces" on page 50.
mpls-exp	Yes	Yes	Yes	Yes	For M Series router FPCs, only the default MPLS EXP classifier is supported; the default MPLS EXP classifier takes the EXP bits 1 and 2 as the output queue number.
Loss priorities based on the Frame Relay discard eligible (DE) bit	Yes	No	No	No	–

Table 49: CoS Hardware Capabilities and Limitations *(continued)*

Feature	J Series PIMs	M Series FPCs	M Series Enhanced FPCs	M320 and T Series FPCs	Comments
Drop Profiles					
Maximum number per FPC, PIC, or PIM	32	2	16	32	–
Per queue	Yes	No	Yes	Yes	–
Per loss priority	Yes	Yes	Yes	Yes	–
Per Transmission Control Protocol (TCP) bit	Yes	No	Yes	Yes	–
Policing					
Adaptive shaping for Frame Relay traffic	Yes	No	No	No	–
Traffic policing	Yes	Yes	Yes	Yes	–
Two-rate tricolor marking (TCM)	No	No	No	Yes	Allows you to configure up to four loss priorities. Two-rate TCM is supported on T Series routers with Enhanced II FPCs and the T640 Core Router with Enhanced Scaling FPC4.
Virtual channels	Yes	No	No	No	–
Queuing					Gigabit Ethernet IQ2 PICs support only one queue in the scheduler map with **medium-high**, **high**, or **strict-high** priority. If more than one queue is configured with **high** or **strict-high** priority, the one that appears first in the configuration is implemented as **strict-high** priority. This queue receives unlimited transmission bandwidth. The remaining queues are implemented as **low** priority, which means they might be starved. On the IQE PIC, you can rate-limit the strict-high and high queues. Without this limiting, traffic that requires low latency (delay) such as voice can block the transmission of medium-priority and low-priority packets. Unless limited, high and strict-high traffic is always sent before lower priority traffic.

Table 49: CoS Hardware Capabilities and Limitations *(continued)*

Feature	J Series PIMs	M Series FPCs	M Series Enhanced FPCs	M320 and T Series FPCs	Comments
Priority	Yes	No	Yes	Yes	Support for the **medium-low** and **medium-high** queuing priority mappings varies by FPC type. For more information, see "Platform Support for Priority Scheduling" on page 164.
Per-queue output statistics	Yes	No	Yes	Yes	Per-queue output statistics are shown in the output of the **show interfaces queue** command.
Rewrite Markers					
Maximum number per FPC, PIC, or PIM	64	No maximum	No maximum	64	–
dscp	Yes	No	Yes	Yes	For J Series router PIMs and M Series Enhanced FPCs, bits 0 through 5 are rewritten, and bits 6 through 7 are preserved. For M320 and T Series router non-IQ FPCs, bits 0 through 5 are rewritten, and bits 6 through 7 are preserved. For M320 and T Series router FPCs, you must decode the loss priority using the firewall filter before you can use loss priority to select the rewrite CoS value. For more information, see "Setting Packet Loss Priority" on page 60. For M320 and T Series router FPCs, Adaptive Services PIC link services IQ interfaces (**lsq-**) do not support DSCP rewrite markers.
dscp-ipv6	Yes	No	Yes	Yes	For J Series router PIMs, M Series router Enhanced FPCs, and M320 and T Series router FPCs, bits 0 through 5 are rewritten, and bits 6 through 7 are preserved. For M320 and T Series routers FPCs, you must decode the loss priority using the firewall filter before you can use loss priority to select the rewrite CoS value. For more information, see "Setting Packet Loss Priority" on page 60. For M320 and T Series router FPCs, Adaptive Services PIC link services IQ interfaces (**lsq-**) do not support DSCP rewrite markers.
frame-relay-de	Yes	No	No	No	–
ieee-802.1	Yes	No	Yes	Yes	For M Series router enhanced FPCs and T Series router FPCs, fixed rewrite loss priority determines the value for bit 0; queue number (forwarding class) determines bits 1 and 2. For IQ PICs, you can only configure one IEEE 802.1 rewrite rule on a physical port. All logical ports (units) on that physical port should apply the same IEEE 802.1 rewrite rule.

Table 49: CoS Hardware Capabilities and Limitations *(continued)*

Feature	J Series PIMs	M Series FPCs	M Series Enhanced FPCs	M320 and T Series FPCs	Comments
inet-precedence	Yes	Yes	Yes	Yes	For J Series router PIMs, bits 0 through 2 are rewritten, and bits 3 through 7 are preserved. For M Series router FPCs, bits 0 through 2 are rewritten, and bits 3 through 7 are preserved. For M Series router Enhanced FPCs, bits 0 through 2 are rewritten, bits 3 through 5 are cleared, and bits 6 through 7 are preserved. For M320 and T Series routers FPCs, bits 0 through 2 are rewritten and bits 3 through 7 are preserved. For M320 and T Series router FPCs, you must decode the loss priority using the firewall filter before you can use loss priority to select the rewrite CoS value. For more information, see "Setting Packet Loss Priority" on page 60.
mpls-exp	Yes	Yes	Yes	Yes	For M320 and T Series router FPCs, you must decode the loss priority using the firewall filter before you can use loss priority to select the rewrite CoS value. For more information, see "Setting Packet Loss Priority" on page 60. For M Series routers FPCs, fixed rewrite loss priority determines the value for bit 0; queue number (forwarding class) determines bits 1 and 2.

Many operations involving the DSCP bits depend on the router and PIC type. For example, some DSCP classification configurations for MPLS and Internet can only be performed on MX, M120, and M320 routers with Enhanced Type III FPCs only. For examples of these possibilities, see "Applying Classifiers to Logical Interfaces" on page 50.

M320 Routers FPCs and CoS

On Juniper Networks M320 Multiservice Edge Routers, CoS is supported with two types of FPCs: the Enhanced II FPC and the Enhanced III FPC. The Enhanced III FPC provides different CoS functionality than the standard and Enhanced III FPCs. You can mix the FPC types in a single M320 router, but CoS processing for packets traveling between the Enhanced II and Enhanced III FPCs differ from the processing of packets traveling between FPCs of the same type. In cases of mixed FPC types, only the least common denominator of CoS functions is supported.

In particular, the drop priority classification behavior is different for packets traveling between Enhanced II and Enhanced III FPCs in an M320 router chassis. In the Enhanced III FPC, the packet is always classified into one of four packet drop priorities whether the **tri-color** statement is configured or not. However, depending on the presence or absence of the **tri-color** statement, the four colors might have a different meaning to the Enhanced

II FPC. For more information about the **tri-color** statement, see “Enabling Tricolor Marking” on page 97.

When packets flow from an Enhanced II FPC to an Enhanced III FPC, the drop priority classification behavior is shown in Table 50 on page 265.

Table 50: Drop Priority Classification for Packet Sent from Enhanced III to Enhanced II FPC on M320 Routers

Enhanced III FPC Drop Priority	Enhanced II FPC Drop Priority (Without Tricolor Marking Enabled)	Enhanced II FPC Drop Priority (with Tricolor Marking Enabled)
low	low	low
medium-low	low	medium-low
medium-high	high	medium-high
high	high	high

When packets flow from an Enhanced II FPC without tricolor marking enabled to an Enhanced III FPC, the drop priority classification behavior is shown in Table 51 on page 265.

Table 51: Drop Priority Classification for Packet Sent from Enhanced II FPC Without Tricolor Marking to Enhanced III FPC on M320 Routers

Enhanced II FPC (Without Tricolor Marking Enabled)	Enhanced III FPC
low	low
high	medium-high

When packets flow from an Enhanced II FPC with tricolor marking enabled to an Enhanced III FPC, the drop priority classification behavior is shown in Table 52 on page 265.

Table 52: Drop Priority Classification for Packet Sent from Enhanced II FPC with Tricolor Marking to Enhanced III FPC on M320 Routers

Enhanced II FPC (With Tricolor Marking Enabled)	Enhanced III FPC
low	low
medium-low	medium-low
medium-high	medium-high
high	high

MX Series Router CoS Hardware Capabilities and Limitations

Generally, the Layer 3 CoS hardware capabilities and limitations for Juniper Networks MX Series Ethernet Service Routers are the same as for M Series Multiservice Edge Routers (M120 routers in particular).

In particular, the following scaling and performance parameters apply to MX Series routers:

- 32 classifiers of each type
- 32 rewrite tables of each type
- Eight queues per port
- 64 WRED profiles
- 100-ms queue buffering for interfaces 1 Gbps and above; 500 ms for all others
- Line-rate CoS features

For more information about MX Series router CoS capabilities, including software configuration, see "Configuring Hierarchical Schedulers for CoS" on page 207 and "Enhanced Queuing DPC Hardware Properties" on page 355.

On MX Series routers, you can apply classifiers or rewrite rules to an integrated bridging and routing (IRB) interface at the **[edit class-of-service interfaces irb unit *logical-unit-number*]** level of the hierarchy. All types of classifiers and rewrite rules are allowed. These classifiers and rewrite rules are independent of others configured on an MX Series router.

```
[edit class-of-service interfaces]
irb {
  unit logical-unit-number {
    classifiers {
      type (classifier-name | default) family (mpls | all);
    }
    rewrite-rules {
      dscp (rewrite-name | default);
      dscp-ipv6 (rewrite-name | default);
      exp (rewrite-name | default)protocol protocol-types;
      ieee-802.1 (rewrite-name | default) vlan-tag (outer | outer-and-inner);
      inet-precedence (rewrite-name | default);
    }
  }
}
```

For IQ PICs, you can only configure one IEEE 802.1 rewrite rule on a physical port. All logical ports (units) on that physical port should apply the same IEEE 802.1 rewrite rule.

The IRB classifiers and rewrite rules are applied only to the "routed" packets. For logical interfaces that are part of a bridge domain, only IEEE classifiers and IEEE rewrite rules are allowed. Only the listed options are available for rewrite rules on an IRB.

For dual-tagged bridge domain logical interfaces, you can configure classification based on the inner or outer VLAN tag's IEEE 802.1p bits using the **vlan-tag** statement with the **inner** or **outer** option:

```
[edit class-of-service interfaces interface-name unit logical-unit-number]
classifiers {
  ieee-802.1 (classifier-name | default) vlan-tag (inner | outer);
}
```

Also, for dual-tagged bridge domain logical interfaces, you can configure rewrite rules to rewrite the outer or both outer and inner VLAN tag's IEEE 802.1p bits using the **vlan-tag** statement with the **outer** or **outer-and-inner** option:

```
[edit class-of-service interfaces interface-name unit logical-unit-number]
rewrite-rules {
  ieee-802.1 (rewrite-rule-name | default) vlan-tag (outer | outer-and-inner);
}
```

Default Routing Engine Protocol Queue Assignments

Table 53 on page 267 lists (in alphabetical order) how Routing Engine-sourced traffic is mapped to output queues. The follow caveats apply to Table 53 on page 267:

- For all packets sent to queue 3 over a VLAN-tagged interface, the software sets the 802.1p bit to 110.
- 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 53: Routing Engine Protocol Queue Assignments

Routing Engine Protocol	Queue Assignment
Adaptive Services PIC	TCP tickle (keepalive packets for idle session generated with stateful firewall to probe idle TCP sessions) are sent from queue 0.
ATM Operation, Administration, and Maintenance (OAM)	Queue 3
Bidirectional Forwarding Detection (BFD) Protocol	Queue 3
Border Gateway Protocol (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
Frame Relay Local Management Interface (LMI)	Queue 3

Table 53: Routing Engine Protocol Queue Assignments *(continued)*

Routing Engine Protocol	Queue Assignment
Frame Relay Asynchronization permanent virtual circuit (PVC)/data link connection identifier (DLCI) status messages	Queue 3
FTP	Queue 0
Intermediate System-to-Intermediate System (IS-IS) Open Systems Interconnection (OSI)	Queue 3
Internet Group Management Protocol (IGMP) query	Queue 3
IGMP Report	Queue 0
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 then 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
Multilink Frame Relay Link Integrity Protocol (LIP)	Queue 3
Open Shortest Path First (OSPF) protocol data unit (PDU)	Queue 3
Point-to-Point Protocol (PPP)	Queue 3
Protocol Independent Multicast (PIM)	Queue 3

Table 53: Routing Engine Protocol Queue Assignments *(continued)*

Routing Engine Protocol	Queue Assignment
Real-time performance monitoring (RPM) probe packets	Queue 3
Resource Reservation Protocol (RSVP)	Queue 3
Routing Information Protocol (RIP)	Queue 3
Simple Network Management Protocol (SNMP)	Queue 0
SSH	Queue 0
Telnet	Queue 0
Virtual Router Redundancy Protocol (VRRP)	Queue 3
xnm-clear-text	Queue 0
xnm-ssl	Queue 0

Changing the Routing Engine Outbound Traffic Defaults

You can modify the default queue assignment (forwarding class) and DSCP bits used in the ToS field of packets generated by the Routing Engine. By default, the forwarding class (queue) and packet loss priority (PLP) bits are set according to the values given in “Default DSCP and DSCP IPv6 Classifier” on page 45.

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

This feature is not available on Juniper Networks J Series Services Routers.

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

The queue selected must be properly configured on all interfaces. For more information about configuring queues and forwarding classes, see “Overview of Forwarding Classes” on page 111.

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

```
[edit class-of-service]
host-outbound-traffic {
```

```
    forwarding-class class-name;
    dscp-code-point value;
}
```

The following example places all Routing Engine-sourced traffic into queue 3 (network control) with a DSCP code point value of 101010:

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

CoS Features of Router Hardware and Interface Families

CoS Features of the Router Hardware and PIC Families

Table 54 on page 270 compares the PIC families with regard to major CoS features. Note that this table reflects the ability to perform the CoS function *at the PIC level* and not on the system as a whole.

Table 54: CoS Features of the Router Hardware and Interface Families Compared

Feature:	M320 and T Series	Trio DPC/MPCs	IQ PICs	IQ2 PICs	IQ2E PICs	Enhanced IQ PICs
BA classification	Yes	Yes	–	–	–	Yes
ToS bit rewrites	Yes	Yes	Yes, for IEEE bits only	Yes, for IEEE bits only	Yes, for IEEE bits only	–
Ingress ToS bit rewrites	–	Yes, with firewall filter	–	–	–	Yes
Hierarchical policers	–	Yes	–	–	–	Yes

Scheduling on the Router Hardware and PIC Families

Table 55 on page 271 compares the PIC families with regard to scheduling abilities or features. Note that this table reflects the ability to perform the function *at the PIC level* and not necessarily on the system as a whole. In this table, the OSE PICs refer to the 10-port 10-Gigabit OSE PICs.

Table 55: Scheduling on Router Hardware and Interface Families Compared

Scheduling Feature:	M320 and T Series	Trio DPC/MPCs	IQ PICs	IQ2 PICs	IQ2E PICs	OSE PICs on T Series	Enhanced IQ PICs
Per–unit scheduling	–	Yes, for EQ MPC	Yes	Yes	Yes	–	Yes
Physical port and logical unit shaping	–	Yes	–	Yes	Yes	–	Yes
Guaranteed rate or peak rate support	–	Yes	–	Yes	Yes	Yes, at the queue level	Yes, at the logical unit
Excess rate support	–	Yes	–	–	–	Yes	Yes, at the logical unit
Shared scheduler support	–	–	–	Yes	Yes	–	–

Schedulers on the Router Hardware and PIC Families

Table 56 on page 271 compares the PIC families with regard to scheduler statements or features. Note that this table reflects the ability to perform the scheduler function *at the PIC level* and not necessarily on the system as a whole. In this table, the OSE PICs refer to the 10-port 10-Gigabit OSE PICs.

Table 56: Schedulers on the Router Hardware and Interface Families Compared

Scheduler Statement or Feature:	M320 and T Series	Trio DPC/MPCs	IQ PICs	IQ2 PICs	IQ2E PICs	OSE PICs on T Series	Enhanced IQ PICs
Exact	Yes	Yes	Yes	–	–	Yes	Yes
Rate-limit	–	–	–	Yes	Yes	Yes	Yes
Traffic shaping	–	Yes	–	–	Yes	Yes	Yes

Table 56: Schedulers on the Router Hardware and Interface Families Compared *(continued)*

Scheduler Statement or Feature:	M320 and T Series	Trio DPC/MPCs	IQ PICs	IQ2 PICs	IQ2E PICs	OSE PICs on T Series	Enhanced IQ PICs
More than one high-priority queue	Yes	Yes	Yes	–	Yes	–	Yes
Excess priority or sharing	–	Yes	–	–	–	–	Yes
Hierarchical Scheduling	–	Yes, for EQ MPC	–	–	Yes	–	–

Queuing Parameters for the Router Hardware and PIC Families

Table 57 on page 272 compares the PIC families with regard to queuing parameters and features. In this table, the OSE PICs refer to the 10-port 10-Gigabit OSE PICs.

Table 57: Queue Parameters on the Router Hardware and Interface Families Compared

Queuing Statement or Feature:	M320 and T Series	Trio DPC/MPCs	IQ PICs	IQ2 PICs	IQ2E PICs	OSE PICs on T Series	Enhanced IQ PICs
Maximum number of queues	8	8	8 on M320 or T Series routers, 4 on M7, M10, M20 routers	8	8	4 ingress, 8 egress	8
Maximum delay buffer bandwidth	80 ms: Type 1 and 2 FPC, 50 ms: Type 3 FPC	100 ms for 1 Gbps and up; 500 ms for others	100 ms	200 ms	200 ms	–	up to 4000 ms
Packet transmit priority level	3 and 3	3 and 2	2 and 2	2	3	2	3 and 2

Table 57: Queue Parameters on the Router Hardware and Interface Families Compared *(continued)*

Queuing Statement or Feature:	M320 and T Series	Trio DPC/MPCs	IQ PICs	IQ2 PICs	IQ2E PICs	OSE PICs on T Series	Enhanced IQ PICs
Maximum number of drop profiles	32 (32 samples)	64	32 (32 samples)	32	32	–	64
Packet loss priority level	4	4	4	4	4	4	4

CHAPTER 16

Configuring CoS for Tunnels

This topic discusses the following:

CoS for Tunnels Overview

For Adaptive Services, Link Services, and Tunnel PICs installed on Juniper Networks M Series Multiservice Edge Routers and T Series Core Routers with enhanced Flexible PIC Concentrators (FPCs), class-of-service (CoS) information is preserved inside generic routing encapsulation (GRE) and IP-IP tunnels.

For the ES PIC installed on M Series and T Series routers with enhanced FPCs, class-of-service information is preserved inside IP Security (IPsec) tunnels. For IPsec tunnels, you do not need to configure CoS, because the ES PIC copies the type-of-service (ToS) byte from the inner IP header to the GRE or IP-IP header.

For IPsec tunnels, the IP header type-of-service (ToS) bits are copied to the outer IPsec header at encryption side of the tunnel. You can rewrite the outer ToS bits in the IPsec header using a rewrite rule. On the decryption side of the IPsec tunnel, the ToS bits in the IPsec header are not written back to the original IP header field. You can still apply a firewall filter to the ToS bits to apply a packet action on egress. For more information about ToS bits and the Multiservices PICs, see “Multiservices PIC ToS Translation” on page 289. For more information about IPsec and Multiservices PICs, see the *Junos OS Services Interfaces Configuration Guide*.

To configure CoS for tunnels, include the following statements at the **[edit class-of-service]** and **[edit interfaces]** hierarchy level:

```
[edit class-of-service]
interfaces {
  interface-name {
    unit logical-unit-number {
      rewrite-rules {
        dscp (rewrite-name | default);
        dscp-ipv6 (rewrite-name | default);
```

```
                exp (rewrite-name | default)protocol protocol-types;
                exp-push-push-push default;
                exp-swap-push-push default;
                ieee-802.1 (rewrite-name | default);
                inet-precedence (rewrite-name | default);
            }
        }
    }
}
rewrite-rules {
    (dscp | dscp-ipv6 | exp | ieee-802.1 | inet-precedence) rewrite-name {
        import (rewrite-name | default);
        forwarding-class class-name {
            loss-priority level code-point (alias | bits);
        }
    }
}
[edit interfaces]
gre-interface-name {
    unit logical-unit-number;
    copy-tos-to-outer-ip-header;
}
```

Configuring CoS for Tunnels

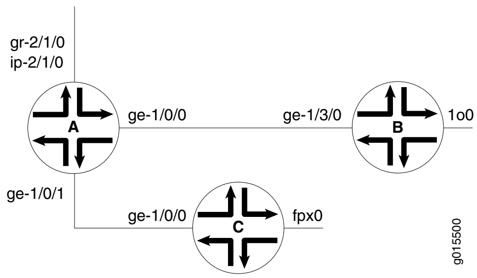
To configure CoS for GRE and IP-IP tunnels, perform the following configuration tasks:

1. To configure the tunnel, include the **tunnel** statement at the **[edit interfaces ip-*fpc/pic/port* unit *logical-unit-number*]** or **[edit interfaces gr-*fpc/pic/port* unit *logical-unit-number*]** hierarchy level.
2. To rewrite traffic on the outbound interface, include the **rewrite-rules** statement at the **[edit class-of-service]** and **[edit class-of-service interfaces *interface-name* unit *logical-unit-number*]** hierarchy levels. For GRE and IP-IP tunnels, you can configure IP precedence and DSCP rewrite rules.
3. To classify traffic on the inbound interface, you can configure a behavior aggregate (BA) classifier or firewall filter. Include the **loss-priority** and **forwarding-class** statements at the **[edit firewall filter *filter-name* term *term-name* then]** hierarchy level, or the **classifiers** statement at the **[edit class-of-service]** hierarchy level.
4. For a GRE tunnel, the default is to set the ToS bits in the outer IP header to all 0s. To copy the ToS bits from the inner IP header to the outer, include the **copy-tos-to-outer-ip-header** statement at the **[edit interfaces gr-*fpc/pic/port* unit *logical-unit-number*]** hierarchy level. (This inner-to-outer ToS bits copying is already the default behavior for IP-IP tunnels.)

Example: Configuring CoS for Tunnels

In Figure 21 on page 277, Router A acts as a tunnel ingress device. The link between interfaces **ge-1/0/0** in Router A and **ge-1/3/0** in Router B is the GRE or IP-IP tunnel. Router A monitors the traffic received from interface **ge-1/3/0**. By way of interface **ge-1/0/0**, Router C generates traffic to Router B.

Figure 21: CoS with a Tunnel Configuration



Router A

```
[edit interfaces]
ge-1/0/0 {
   unit 0 {
      family inet {
         address 10.80.0.2/24;
      }
   }
}
ge-1/0/1 {
   unit 0 {
      family inet {
         filter {
            input zf-catch-all;
         }
         address 10.90.0.2/24;
      }
   }
}
gr-2/1/0 {
   unit 0 {
      tunnel {
         source 11.11.11.11;
         destination 10.255.245.46;
      }
      family inet {
         address 21.21.21.21/24;
      }
   }
}
ip-2/1/0 {
   unit 0 {
      tunnel {
         source 12.12.12.12;
         destination 10.255.245.46;
      }
      family inet {
         address 22.22.22.22/24;
      }
   }
}

[edit routing-options]
static {
   route 1.1.1.1/32 next-hop gr-2/1/0.0;
```

```
    route 2.2.2.2/32 next-hop ip-2/1/0.0;
}

[edit class-of-service]
interfaces {
    ge-1/0/0 {
        unit 0 {
            rewrite-rules {
                inet-precedence zf-tun-rw-ipprec-00;
            }
        }
    }
}
rewrite-rules {
    inet-precedence zf-tun-rw-ipprec-00 {
        forwarding-class best-effort {
            loss-priority low code-point 000;
            loss-priority high code-point 001;
        }
        forwarding-class expedited-forwarding {
            loss-priority low code-point 010;
            loss-priority high code-point 011;
        }
        forwarding-class assured-forwarding {
            loss-priority low code-point 100;
            loss-priority high code-point 101;
        }
        forwarding-class network-control {
            loss-priority low code-point 110;
            loss-priority high code-point 111;
        }
    }
}
dscp zf-tun-rw-dscp-00 {
    forwarding-class best-effort {
        loss-priority low code-point 000000;
        loss-priority high code-point 001001;
    }
    forwarding-class expedited-forwarding {
        loss-priority low code-point 010010;
        loss-priority high code-point 011011;
    }
    forwarding-class assured-forwarding {
        loss-priority low code-point 100100;
        loss-priority high code-point 101101;
    }
    forwarding-class network-control {
        loss-priority low code-point 110110;
        loss-priority high code-point 111111;
    }
}

[edit firewall]
filter zf-catch-all {
    term term1 {
        then {
```



```
            loss-priority high;
            forwarding-class network-control;
        }
    }
}
```

Router B

```
[edit interfaces]
ge-1/3/0 {
    unit 0 {
        family inet {
            address 10.80.0.1/24;
        }
    }
}
lo0 {
    unit 0 {
        family inet {
            address 10.255.245.46/32;
        }
    }
}
```

Router C

```
[edit interfaces]
ge-1/0/0 {
    unit 0 {
        family inet {
            address 10.90.0.1/24;
        }
    }
}

[edit routing-options]
static {
    route 1.1.1.1/32 next-hop 10.90.0.2;
    route 2.2.2.2/32 next-hop 10.90.0.2;
}
```

Example: Configuring a GRE Tunnel to Copy ToS Bits to the Outer IP Header

Unlike IP-IP tunnels, GRE tunnels do not copy the ToS bits to the outer IP header by default. 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. 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;
    }
}
```

CHAPTER 17

Configuring CoS on Services PICs

This topic discusses the following:

Overview of CoS on Services PICs

On Adaptive Services (AS) PICs and Multiservices PICs with **lsq-** interfaces, there are additional features you can configure. One such feature is an additional method of classifying traffic flows based on applications, for example stateful firewalls and network address translation (NAT).

Application-based traffic flow classification enables you to configure a rule-based service that provides DiffServ code point (DSCP) marking and forwarding-class assignments for traffic transiting the AS PIC. The service enables you to specify matching by application, application set, source, destination address, and match direction, and uses a similar structure to other rule-based services such as stateful firewall. The service actions allow you to associate the DSCP alias or value, forwarding-class name, system log activity, or a preconfigured application profile with the matched packet flows.

NOTE: If you configure a forwarding class map associating a forwarding class with a queue number, these maps are not supported on MultiServices link services intelligent queuing (lsq-) interfaces.

To configure class-of-service (CoS) features on the Adaptive Services PIC or Multiservices PIC, include the **cos** statement at the **[edit services]** hierarchy level:

```
[edit services]
cos {
```

```
    application-profile profile-name {
        ftp {
            data {
                dscp (alias | bits);
                forwarding-class class-name;
            }
        }
        sip {
            video {
                dscp (alias | bits);
                forwarding-class class-name;
            }
            voice {
                dscp (alias | bits);
                forwarding-class class-name;
            }
        }
    }
    rule rule-name {
        match-direction (input | output | input-output);
        term term-name {
            from {
                applications [ application-names ];
                application-sets [ set-names ];
                destination-address address;
                source-address address;
            }
            then {
                application-profile profile-name;
                dscp (alias | bits);
                forwarding-class class-name;
                syslog;
                (reflexive | reverse) {
                    application-profile profile-name;
                    dscp (alias | bits);
                    forwarding-class class-name;
                    syslog;
                }
            }
        }
    }
    rule-set rule-set-name {
        [ rule rule-names ];
    }
}
```

Configuring CoS Rules

To configure a CoS rule, include the **rule** ***rule-name*** statement at the **[edit services cos]** hierarchy level:

```
[edit services cos]
rule rule-name {
    match-direction (input | output | input-output);
    term term-name {
```



```
        from {
            applications [ application-names ];
            application-sets [ set-names ];
            destination-address address;
            source-address address;
        }
        then {
            application-profile profile-name;
            dscp (alias | bits);
            forwarding-class class-name;
            syslog;
            (reflexive | reverse) {
                application-profile profile-name;
                dscp (alias | bits);
                forwarding-class class-name;
                syslog;
            }
        }
    }
}
```

Each CoS rule consists of a set of terms, similar to a filter configured at the **[edit firewall]** hierarchy level. A term consists of the following:

- **from** statement—Specifies the match conditions and applications that are included and excluded.
- **then** statement—Specifies the actions and action modifiers to be performed by the router software.

In addition, each rule must include a **match-direction** statement that specifies the direction in which the rule match is applied. To configure where the match is applied, include the **match-direction** statement at the **[edit services cos rule *rule-name*]** hierarchy level:

```
match-direction (input | output | input-output);
```

If you configure **match-direction input-output**, bidirectional rule creation is allowed.

The match direction is used with respect to the traffic flow through the Services PIC. When a packet is sent to the Services PIC, direction information is carried along with it.

With an interface service set, packet direction is determined by whether a packet is entering or leaving the interface on which the service set is applied.

With a next-hop service set, packet direction is determined by the interface used to route the packet to the Services PIC. If the inside interface is used to route the packet, the packet direction is **input**. If the outside interface is used to direct the packet to the Services PIC, the packet direction is **output**. For more information on inside and outside interfaces, see the *Junos OS Services Interfaces Configuration Guide*.

On the Services PIC, a flow lookup is performed. If no flow is found, rule processing is performed. All rules in the service set are considered. During rule processing, the packet direction is compared against rule directions. Only rules with direction information that matches the packet direction are considered.

The following sections describe CoS rule content in more detail:

- Configuring Match Conditions in a CoS Rule on page 284
- Configuring Actions in a CoS Rule on page 285

Configuring Match Conditions in a CoS Rule

To configure CoS match conditions, include the **from** statement at the **[edit services cos rule *rule-name* term *term-name*]** hierarchy level:

```
[edit services cos rule rule-name term term-name]
from {
  applications [ application-names ];
  application-sets [ set-names ];
  destination-address address;
  source-address address;
}
```

You can use either the source address or the destination address as a match condition, in the same way that you would configure a firewall filter; for more information, see the *Junos OS Policy Framework Configuration Guide*.

If you omit the **from** term, the router accepts all traffic and the default protocol handlers take effect:

- User Datagram Protocol (UDP), Transmission Control Protocol (TCP), and Internet Control Message Protocol (ICMP) create a bidirectional flow with a predicted reverse flow.
- IP creates a unidirectional flow.

You can also include application protocol definitions that you have configured at the **[edit applications]** hierarchy level; for more information, see the *Junos OS Services Interfaces Configuration Guide*.

- To apply one or more specific application protocol definitions, include the **applications** statement at the **[edit services cos rule *rule-name* term *term-name* from]** hierarchy level.
- To apply one or more sets of application protocol definitions you have defined, include the **application-sets** statement at the **[edit services cos rule *rule-name* term *term-name* from]** hierarchy level.

NOTE: If you include a statement that specifies application protocols, the router derives port and protocol information from the corresponding configuration at the **[edit applications]** hierarchy level; you cannot specify these properties as match conditions.

Configuring Actions in a CoS Rule

To configure CoS actions, include the **then** statement at the **[edit services cos rule *rule-name* term *term-name*]** hierarchy level:

```
[edit services cos rule rule-name term term-name]
then {
  application-profile profile-name;
  dscp (alias | bits);
  forwarding-class class-name;
  syslog;
  (reflexive | reverse) {
    application-profile profile-name;
    dscp (alias | bits);
    forwarding-class class-name;
    syslog;
  }
}
```

The principal CoS actions are as follows:

- **dscp**—Marks the packet with the specified DiffServ code point (DSCP) value or alias.
- **forwarding-class**—Assigns the packet to the specified forwarding class.

You can optionally set the configuration to record information in the system logging facility by including the **syslog** statement at the **[edit services cos rule *rule-name* term *term-name* then]** hierarchy level. This statement overrides any **syslog** setting included in the service set or interface default configuration.

For information about some additional CoS actions, see the following sections:

- Configuring Application Profiles on page 285
- Configuring Reflexive and Reverse CoS Actions on page 286

Configuring Application Profiles

You can optionally define one or more application profiles for inclusion in CoS actions. To configure, include the **application-profile** statement at the **[edit services cos]** hierarchy level:

```
[edit services cos]
application-profile profile-name {
  ftp {
    data {
      dscp (alias | bits);
      forwarding-class class-name;
    }
  }
  sip {
    video {
      dscp (alias | bits);
      forwarding-class class-name;
    }
    voice {
```

```
            dscp (alias | bits);
            forwarding-class class-name;
          }
        }
      }
```

The **application-profile** statement includes two main components and three traffic types: **ftp** with the **data** traffic type and **sip** with the **video** and **voice** traffic types. You can set the appropriate **dscp** and **forwarding-class** values for each component within the application profile.

NOTE: The **ftp** and **sip** statements are not supported on Juniper Network MX Series Ethernet Services Routers.

You can apply the application profile to a CoS configuration by including it at the **[edit services cos rule *rule-name* term *term-name* then]** hierarchy level.

Configuring Reflexive and Reverse CoS Actions

It is important to understand that CoS services are unidirectional. It might be necessary to specify different treatments for flows in opposite directions.

Regardless of whether a packet matches the input, output, or input-output direction, flows in both directions are created. The difference is that a forward, reverse, or forward-and-reverse CoS action is associated with each flow. You should bear in mind that the flow in the opposite direction might end up having a CoS action associated with it, which you have not specifically configured.

To control the direction in which service is applied, separate from the direction in which the rule match is applied, you can configure the **reflexive** or **reverse** statement at the **[edit services cos rule *rule-name* term *term-name* then]** hierarchy level:

```
[edit services cos rule rule-name term term-name then]
(reflexive | reverse) {
  application-profile profile-name;
  dscp (alias | bits);
  forwarding-class class-name;
  syslog;
}
```

The two actions are mutually exclusive. If nothing is specified, data flows inherit the CoS behavior of the forward control flow.

- **reflexive** causes the equivalent reverse CoS action to be applied to flows in the opposite direction.
- **reverse** allows you to define the CoS behavior for flows in the reverse direction.

Configuring CoS Rule Sets

The **rule-set** statement defines a collection of CoS rules that determine what actions the router software performs on packets in the data stream. You define each rule by specifying a rule name and configuring terms. You then specify the order of the rules by including the **rule-set** statement at the **[edit services cos]** hierarchy level:

```
[edit services cos]
rule-set rule-set-name {
  rule rule-name1;
  rule rule-name2;
  rule rule-name3;
  ...
}
```

The router software processes the rules in the order in which you specify them in the configuration. If a term in a rule matches the packet, the router performs the corresponding action and the rule processing stops. If no term in a rule matches the packet, processing continues to the next rule in the rule set. If none of the rules match the packet, the packet is dropped by default.

Output Packet Rewriting

On M Series routers, you can configure rewrite rules to change packet header information and attach it to an output interface. Because these rules can possibly overwrite the DSCP marking configured on the AS PIC, it is important to create system-wide configurations carefully.

For example, knowing that the AS PIC or Multiservices PIC can mark packets with any ToS or DSCP value and the output interface is restricted to only eight DSCP values, rewrite rules on the output interface condense the mapping from 64 to 8 values with overall loss of granularity. In this case, you have the following options:

- Remove rewrite rules in the output interface.
- Configure the output interface to include the most important mappings.

Allocating Excess Bandwidth Among Frame Relay DLCIs on Multiservices PICs

By default, all logical (**lsq-**) interfaces on a Multiservices PIC share bandwidth equally in the excess region (that is, bandwidth available once these interfaces have exhausted their committed information rate (CIR).

However, you can include the **excess-rate** statement to control an independent set of parameters for bandwidth sharing in the excess region of a frame relay data-link connection identifier (DLCI) on a Multiservices PIC. Include the **excess-rate** statement at the **[edit class-of-service traffic-control-profile *traffic-control-profile-name*]** hierarchy level.

```
[edit class-of-service traffic-control-profile traffic-control-profile-name]
excess-rate percent percentage;
```


There are several limitations to this feature:

- The excess bandwidth comes from bandwidth not used by any DLCIs (that is, bandwidth above the CIR). Therefore, only FRF.16 is supported.
- Only CIR mode is supported (you must configure a CIR on at least one DLCI).
- Only the **percent** option is supported for **lsq-** interfaces. The **priority** option is not supported for DLCIs.
- You cannot configure this feature if you also include one of the following statements in the configuration:
  - **scheduler-map**
  - **shaping-rate**
  - **adaptive-shaper** (valid on J Series Services Routers only)
  - **virtual-channel-group** (valid on J Series Services Routers only)
- If you oversubscribe the DLCIs, then the bandwidth can only be distributed equally.
- The **excess-priority** statement is not supported. However, for consistency, this statement will not result in a commit error.
- This feature is only supported on the Multiservices 100, Multiservices 400, and Multiservices 500 PICs.

This example configures excess bandwidth sharing in the ratio of 70 to 30 percent for two frame relay DLCIs. Only FRF.16 interfaces are supported.

Configuring the Frame Relay DLCIs

You must configure the per-unit scheduler.

```
[edit interfaces]
lsq-1/3/0:0 {
   per-unit-scheduler;
   unit 0 {
      dlci 100;
   }
   unit 1 {
      dlci 200;
   }
}
```

Configuring the Traffic Control Profile

Only the **percent** option is supported.

```
[edit class-of-service]
traffic-control-profiles {
   tc_70 {
      excess-rate percent 70;
   }
   tc_30 {
      excess-rate percent 30;
   }
}
```


Applying the Traffic Control Profiles

Only FRF.16 is supported.

```
[edit interfaces]
lsq-1/3/0 {
  unit 0 {
    output-traffic-control-profile tc_70;
  }
  unit 1 {
    output-traffic-control-profile tc_30;
  }
}
```

Multiservices PIC ToS Translation

By default, all logical (**lsq-**) interfaces on a Multiservices PIC preserve the type-of-service (ToS) bits in an incoming packet header.

However, you can use the **translation-table** statement at the **[edit class-of-service]** hierarchy level to replace the arriving ToS bit pattern with a user-defined value.

This feature follows exactly the same configuration rules as the Enhanced IQ PIC. For configuration details, see “Configuring ToS Translation Tables” on page 291.

Example: Configuring CoS Rules

The following example show a CoS configuration containing two rules, one for input matching on a specified application set and the other for output matching on a specified source address:

```
[edit services]
cos {
  application-profile cosprofile {
    ftp {
      data {
        dscp af11;
        forwarding-class 1;
      }
    }
  }
  application-profile cosrevprofile {
    ftp {
      data {
        dscp af22;
      }
    }
  }
  rule cosrule {
    match-direction input;
    term costerm {
      from {
        source-address {
          any-unicast;
        }
        applications junos-ftp;
```

```
            }
            then {
               dscp af33;
               forwarding-class 3;
               application-profile cosprofile;
               reverse {
                  dscp af43;
                  application-profile cosrevprofile;
               }
            }
         }
      }
   }
stateful-firewall {
   rule r1 {
      match-direction input;
      term t1 {
         from {
            application-sets junos-algs-outbound;
         }
         then {
            accept;
         }
      }
      term t2 {
         then {
            accept;
         }
      }
   }
   service-set test {
      stateful-firewall-rules r1;
      cos-rules cosrule;
      interface-service {
         service-interface sp-1/3/0;
      }
   }
}
```

CHAPTER 18

Configuring CoS on Enhanced IQ PICs

This topic discusses the following:

CoS on Enhanced IQ PICs Overview

The Enhanced IQ (IQE) PIC family supports a series of non-channelized and channelized interfaces that run at a large variety of speeds. Sophisticated Class-of-Service (CoS) techniques are available for the IQE PICs at the channel level. These techniques include policing based on type-of-service (ToS) bits, five priority levels, two shaping rates (the guaranteed rate and shaping rate), a shared scheduling option, Diffserv code point (DSCP) rewrite on egress, and configurable delay buffers for queuing. All of these features, with numerous examples, are discussed in this chapter. For a comparison of the capabilities of IQE PICs with other types of PICs, see "Hardware Capabilities and Limitations" on page 259.

For information about CoS components that apply generally to all interfaces, see "CoS Overview" on page 3. For general information about configuring interfaces, see the *Junos OS Network Interfaces Configuration Guide*.

IQE PICs can be used in Juniper Networks M40e, M120, M320 Multiservice Edge Routers and T Series Core Routers to supply enhanced CoS capabilities for edge aggregation. The same interface configuration syntax is used for basic configuration, and other CoS statements are applied at channel levels. Some configuration statements are available only in Junos OS Release 9.3 and later, as noted in this chapter.

Configuring ToS Translation Tables

On the IQE PICs, the behavior aggregate (BA) translation tables are included for every logical interface (unit) protocol family configured on the logical interface. The proper default translation table is active even if you do not include any explicit translation tables.

You can display the current translation table values with the **show class-of-service classifiers** command.

On M40e, M120, M320, and T Series routers with IQE PICs, or on any system with IQ2 or Enhanced IQ2 PICs, you can replace the ToS bit value on the incoming packet header on a logical interface with a user-defined value. The new ToS value is used for all class-of-service processing and is applied before any other class-of-service or firewall treatment of the packet. On the IQE PIC, the values configured with the **translation-table** statement determines the new ToS bit values.

Four types of translation tables are supported: IP precedence, IPv4 DSCP, IPv6 DSCP, and MPLS EXP. You can configure a maximum of eight tables for each supported type. If a translation table is enabled for a particular type of traffic, then behavior aggregate (BA) classification of the same type must be configured for that logical interface. In other words, if you configure an IPv4 translation table, you must configure IPv4 BA classification on the same logical interface.

To configure ToS translation on the IQE PIC, include the **translation-table** statement at the **[edit class-of-service]** hierarchy level:

```
[edit class-of-service]
translation-table {
  (to-dscp-from-dscp | to-dscp-ipv6-from-dscp-ipv6 | to-exp-from-exp |
    to-inet-precedence-from-inet-precedence) table-name {
    to-code-point value from-code-points (* | [ values ]);
  }
}
```

The **from-code-points** statement establishes the values to match on the incoming packets. The **default** option is used to match all values not explicitly listed, and, as a single entry in the translation table, to mark all incoming packets on an interface the same way. The **to-code-point** statement establishes the target values for the translation. If an incoming packet header ToS bit configuration is not covered by the translation table list and a * option is not specified, the ToS bits in the incoming packet header are left unchanged.

You can define many translation tables, as long as they have distinct names. You apply a translation table to a logical interface at the **[edit class-of-service interfaces]** hierarchy level. Translation tables always translate “like to like.” For example, a translation table applied to MPLS traffic can only translate from received EXP bit values to new EXP bit values. That is, translation tables cannot translate (for instance) from DSCP bits to INET precedence code points.

On the IQE PIC, incoming ToS bit translation is subject to the following rules:

- Locally generated traffic is not subject to translation.
- The **to-dscp-from-dscp** translation table type is not supported if an Internet precedence classifier is configured.
- The **to-inet-precedence-from-inet-precedence** translation table type is not supported if a DSCP classifier is configured.
- The **to-dscp-from-dscp** and **to-inet-precedence-from-inet-precendence** translation table types cannot be configured on the same unit.

- The **to-dscp-from-dscp** and **to-inet-precedence-from-inet-precendence** translation table types are supported for IPv4 packets.
- Only the **to-dscp-ipv6-from-dscp-ipv6** translation table type is supported for IPv6 packets.
- Only the **to-exp-from-exp** translation table type is supported for MPLS packets.

NOTE: Translation tables are not supported if fixed classification is configured on the logical interface.

The following example translates incoming DSCP values to the new values listed in the table. All incoming DSCP values other than **111111**, **111110**, **000111**, and **100111** are translated to **000111**.

```
[edit class-of-service]
translation-table {
    to-dscp-from-dscp dscp-trans-table {
        to-code-point 000000 from-code-points 111111;
        to-code-point 000001 from-code-points 111110;
        to-code-point 111000 from-code-points [ 000111 100111 ];
        to-code-point 000111 from-code-points *;
    }
}
```

You must apply the translation table to the logical interface input on the Enhanced IQ PIC:

```
[edit class-of-service interfaces so-1/0/0 unit 0]
translation-table to-dscp-from-dscp dscp-trans-table;
```

A maximum of 32 distinct translation tables are supported on each IQE PIC. However, this maximum is limited by the number of classifiers configured along with translation tables because on the IQE PIC the hardware tables are not always merged. For example, if a translation table and a classifier are both configured on the same logical interface (such as **unit 0**), there is only one hardware table and only one table added to the 32 translation table limit. However, if the translation table is configured on **unit 0** and the classifier on **unit 1** on the same physical interface, then two hardware tables are used and these two tables count toward the 32 maximum.

If you try to configure mutually exclusive translation tables on the same interface unit, you will get a warning message when you display or commit the configuration:

```
ge-0/1/1 {
    unit 0 {
        translation-table {
            ##
            ## Warning: to-dscp-from-dscp and
to-inet-precedence-from-inet-precedence not allowed on same unit
            ##
            to-inet-precedence-from-inet-precedence inet-trans-table;
            to-dscp-from-dscp dscp-trans-table;
        }
```

```
    }
}
```

You can issue the following operational mode commands to verify your configuration:

- **show class-of-service translation-table**
- **show class-of-service interface** ***interface-name***

To verify that the correct values are configures, use the **show class-of-service translation-table** command. The **show class-of-service translation-table** command displays the code points of all translation tables configured. All values are displayed, not just those configured:

```
user@host> show class-of-service translation-table
Translation Table: dscp-trans-table, Translation table type: dscp-to-dscp, Index:
 6761
  From Code point    To Code Point
  000000                      000111
  000001                      000111
  000010                      000111
  000011                      000111
  000100                      000111
  000101                      000111
  000110                      000111
  000111                      111000
  001000                      000111
  001001                      000111
  001010                      000111
  001011                      000111
  001100                      000111
  001101                      000111
  001110                      000111
  001111                      000111
  010000                      000111
  010001                      000111
  010010                      000111
  010011                      000111
  010100                      000111
  010101                      000111
  010110                      000111
  010111                      000111
  011000                      000111
  011001                      000111
  011010                      000111
  011011                      000111
  011100                      000111
  011101                      000111
  011110                      000111
  011111                      000111
  100000                      000111
  100001                      000111
  100010                      000111
  100011                      000111
  100100                      000111
  100101                      000111
  100110                      000111
  100111                      111000
  101000                      000111
  101001                      000111
```



```
101010                    000111
101011                    000111
101100                    000111
101101                    000111
101110                    000111
101111                    000111
110000                    000111
110001                    000111
110010                    000111
110011                    000111
110100                    000111
110101                    000111
110110                    000111
110111                    000111
111000                    000111
111001                    000111
111010                    000111
111011                    000111
111100                    000111
111101                    000111
111110                    000001
111111                    000000
```

To verify that the configured translation table is applied to the correct interface, use the **show class-of-service interface** ***interface-name*** command. The **show class-of-service interface** ***interface-name*** command displays the translation tables applied to the IQE interface:

```
user@host> show class-of-service interface ge-0/1/1
Physical interface: ge-0/1/1, Index: 156  From Code point    To Code Point
  Queues supported: 4, Queues in use: 4
    Scheduler map: <default>, Index: 2
    Chassis scheduler map: <default-chassis>, Index: 4

  Logical interface: so-2/3/0.0, Index: 68
    Object                  Name                   Type                    Index

    Rewrite                 exp-default            exp (mpls-any)             29

    Classifier              dscp-default           dscp                        7

    Classifier              exp-default            exp                        10

    Translation Table       exp-trans-table        EXP_TO_EXP              61925
```

ToS translation on the IQE PIC is a form of behavior aggregate (BA) classification. The IQE PIC does not support multifield classification of packets at the PIC level. For more information about multifield classification, see "Multifield Classifier Overview" on page 71.

Configuring Excess Bandwidth Sharing on IQE PICs

The IQE PIC gives users more control over excess bandwidth sharing. You can set a shaping rate and a guaranteed rate on a queue or logical interface and control the excess bandwidth (if any) that can be used after all bandwidth guarantees have been satisfied.

This section discusses the following topics related to excess bandwidth sharing on the IQE PIC:

- IQE PIC Excess Bandwidth Sharing Overview on page 296
- IQE PIC Excess Bandwidth Sharing Configuration on page 297

IQE PIC Excess Bandwidth Sharing Overview

On some types of PICs, including the IQ and IQ2, and Enhanced Queuing DPCs, you can configure either a committed information rate (CIR) using the **guaranteed-rate** statement or a peak information rate (PIR) using the **shaping-rate** statement. You can configure both a PIR and CIR, and in most cases the CIR is less than the value of PIR. For bursty traffic, the CIR represents the average rate of traffic per unit time and the PIR represents the maximum amount of traffic that can be transmitted in a given interval. In other words, the PIR (**shaping-rate**) establishes the maximum bandwidth available. The CIR (**guaranteed-rate**) establishes the minimum bandwidth available if all sources are active at the same time. Theoretically, the PIR or CIR can be established at the queue, logical interface, or physical interface level. In this section, the PIRs or CIRs apply at the queue or logical interface (or both) levels.

NOTE: You can configure a shaping rate at the physical interface, logical interface, or queue level. You can configure a guaranteed rate or excess rate only at the logical interface and queue level.

Once all of the bandwidth guarantees (the sum of the CIRs at that level) are met, there could still be some excess bandwidth available for use. In existing PICs, you have no control over how this excess bandwidth is used. For example, consider the situation shown in Table 58 on page 296 regarding a 10-Mbps physical interface. This example assumes that all queues are of the same priority. Also, if you do not specify a priority for the excess bandwidth, the excess priority is the same as the normal priority.

Table 58: Default Handling of Excess Traffic

Queue	Transmit Rate (CIR)	Shaping Rate (PIR)	Traffic Rate	Guaranteed Rate (Total = 6 Mbps)	Maximum Rate	Excess Bandwidth (Part of 4 Mbps Excess)	Expected Transmit Rate (Guarantee + Excess)
Q0	10%	80%	10 Mbps	1 Mbps	8 Mbps	0.73 Mbps	1.73 Mbps
Q1	20%	50%	10 Mbps	2 Mbps	5 Mbps	1.45 Mbps	3.45 Mbps
Q2	5%	5%	10 Mbps	0.5 Mbps	0.5 Mbps	0 Mbps	0.5 Mbps
Q3	25%	NA (“100%”)	10 Mbps	2.5 Mbps	10 Mbps	1.82 Mbps	4.32 Mbps

A 10-Mbps interface (the Traffic Rate column) has four queues, and the guaranteed rates are shown as percentages (Transmit Rate column) and in bits per second (Guaranteed Rate column). The table also shows the shaping rate (PIR) as a percentage (Shaping

Rate column) and the actual maximum possible transmitted rate (Traffic Rate column) on the oversubscribed interface. Note the guaranteed rates (CIRs) add up to 60 percent of the physical port speed or 6 Mbps. This means that there are 4 Mbps of "excess" bandwidth that can be used by the queues. This excess bandwidth is used as shown in the last two columns. One column (the Excess Bandwidth column) shows the bandwidth partitioned to each queue as a part of the 4-Mbps excess. The excess 4 Mbps bandwidth is shared in the ratio of the transmit rate (CIR) percentages of 10, 20, 5, and 25, adjusted for granularity. The last column shows the transmit rate the users can expect: the sum of the guaranteed rate plus the proportion of the excess bandwidth assigned to the queue.

Note that on PICs other than the IQE PICs the user has no control over the partitioning of the excess bandwidth. Excess bandwidth partitioning is automatic, simply assuming that the distribution and priorities of the excess bandwidth should be the same as the distribution and priorities of the other traffic. However, this might not always be the case and the user might want more control over excess bandwidth usage.

For more information on how excess bandwidth sharing is handled on the Enhanced Queuing DPC, see "Configuring Excess Bandwidth Sharing" on page 364.

IQE PIC Excess Bandwidth Sharing Configuration

On PICs other than IQE PICs, you can limit a queue's transmission rate by including the **transmit-rate** statement with the **exact** option at the **[edit class-of-service schedulers *scheduler-name*]** hierarchy level. However, on the IQE PIC, you can set a shaping rate independent of the transmit rate by including the **shaping-rate** statement at the **[edit class-of-service schedulers *scheduler-name*]** hierarchy level. Also, other PICs share excess bandwidth (bandwidth left over once the guaranteed transmit rate is met) in an automatic, nonconfigurable fashion. You cannot configure the priority of the queues for the excess traffic on other PICs either.

To share excess bandwidth on IQE PICs, include the **excess-rate** statement along with the **guaranteed-rate** statement (to define the CIR) and the **shaping-rate** statement (to define the PIR):

```
[edit class-of-service traffic-control-profile profile-name]
[edit class-of-service schedulers scheduler-name]
excess-rate percent percentage;
guaranteed-rate (percent percentage | rate);
shaping-rate (percent percentage | rate);
```

To apply these limits to a logical interface, configure the statements at the **[edit class-of-service traffic-control-profile *profile-name*]** hierarchy level. To apply these limits to a specific queue, configure the statements at the **[edit class-of-service schedulers *scheduler-name*]** hierarchy level. You must also complete the configuration by applying the scheduler map or traffic control profile correctly.

You configure the excess rate as a percentage from 1 through 100. By default, excess bandwidth is automatically distributed as on other PIC types.

You can also configure a high or low priority for excess bandwidth by including the **excess-priority** statement with the **high** or **low** option at the **[edit class-of-service schedulers *scheduler-name*]** hierarchy level. This statement establishes the priority at the queue level, which then applies also at the logical and physical interface levels.

```
[edit class-of-service schedulers scheduler-name]
excess-priority (high | low);
```

NOTE: You cannot configure an excess rate for a logical interface if there is no guaranteed rate configured on any logical interface belonging to the physical interface.

The following example configures the excess rate in a traffic control profile:

```
[edit class-of-service traffic-control-profiles]
for-unit-0-percent {
    shaping-rate 10k;
    guaranteed-rate 1k;
    excess-rate percent 30;
}
for-unit-1-proportion {
    shaping-rate 20k;
    guaranteed-rate 10k;
    excess-rate percent 35;
}
```

The following example configures the excess rate in a scheduler.

```
[edit class-of-service schedulers]
scheduler-for-excess-low {
    transmit-rate 1m;
    shaping-rate 5m;
    excess-rate percent 30;
    excess-priority low;
}
scheduler-for-excess-high {
    transmit-rate percent 20;
    shaping-rate percent 30;
    excess-rate percent 25;
    excess-priority high;
}
```

NOTE: All of these parameters apply to egress traffic only and only for per-unit schedulers. That is, there is no hierarchical or shared scheduler support.

You can issue the following operational mode commands to verify your configuration:

- **show class-of-service scheduler-map**
- **show class-of-service traffic-control-profile**

Calculation of Expected Traffic on IQE PIC Queues

This topic discusses the following topics related to calculating the expected traffic flow on IQE PIC queues:

Excess Bandwidth Calculations Terminology

The following terms are used in this discussion of IQE PIC queue calculations:

- CIR mode—A physical interface is in CIR mode when one of more of its “children” (logical interfaces in this case) have a guaranteed rate configured, but some logical interfaces have a shaping rate configured.
- Default mode—A physical interface is in default mode if none of its “children” (logical interfaces in this case) have a guaranteed rate or shaping rate configured.
- Excess mode—A physical interface is in excess mode when one of more of its “children” (logical interfaces in this case) have an excess rate configured.
- PIR mode—A physical interface is in PIR mode if none of its “children” (logical interfaces in this case) have a guaranteed rate configured, but some logical interfaces have a shaping rate configured.

Excess Bandwidth Basics

This basic example illustrates the interaction of the guaranteed rate, the shaping rate, and the excess rate applied to four queues. The same concepts extend to logical interfaces (units) and cases in which the user does not configure an explicit value for these parameters (in that case, the system uses implicit parameters).

In this section, the term “not applicable” (NA) means that the feature is not explicitly configured. All traffic rates are in megabits per second (Mbps).

The hardware parameters derived from the configured rates are relatively straightforward except for the excess weight. The excess rate is translated into an absolute value called the excess weight. The scheduler for an interface picks a logical unit first, and then a queue within the logical unit for transmission. Logical interfaces and queues that are within their guaranteed rates are picked first, followed by those in the excess region. If the transmission rate for a logical interface or queue is more than the shaping rate, the scheduler skips the logical interface or queue. Scheduling in the guaranteed region uses straight round-robin, whereas scheduling in the excess region uses weighed round-robin (WRR) based on the excess weights. The excess weights are in the range from 1 to 127, but they are transparent to the user and subject to change with implementation. The weights used in this example are for illustration only.

This example uses a logical interface with a transmit rate (CIR) of 10 Mbps and a shaping rate (PIR) of 10 Mbps. The user has also configured percentage values of transmit rate (CIR), shaping rate (PIR), and excess rate as shown in Table 59 on page 300.

Table 59: Basic Example of Excess Bandwidth

Queue	Transmit Rate (CIR)	Shaping Rate (PIR)	Excess Rate	Traffic Sent To Queue
Q0	5%	5%	10%	10 Mbps
Q1	30%	80%	50%	10 Mbps
Q2	10%	15%	30%	10 Mbps
Q3	15%	35%	30%	10 Mbps

The values used by the hardware based on these parameters are shown in Table 60 on page 300.

Table 60: Hardware Use of Basic Example Parameters

Queue	Transmit Rate (CIR)	Shaping Rate (PIR)	Excess Weight	Expected Traffic Rate
Q0	0.5 Mbps	0.5 Mbps	10	0.5 Mbps
Q1	3 Mbps	8 Mbps	50	5.19 Mbps
Q2	1 Mbps	1.5 Mbps	30	1.5 Mbps
Q3	1.5 Mbps	3.5 Mbps	30	2.81 Mbps
Totals:	6 Mbps	13.5 Mbps	120	10 Mbps (maximum output)

There are a number of important points regarding excess bandwidth calculations:

- The guaranteed rates should add up to less than the logical interface guaranteed rate (10 Mbps).
- Shaping rates (PIRs) can be oversubscribed.
- Excess rates can be oversubscribed. This rate is only a ratio at which the sharing occurs.
- Each queue receives the minimum of the guaranteed bandwidth because each queue is transmitting at its full burst if it can.
- The excess (remaining) bandwidth is shared among the queues in the ratio of their excess rates. In this case, the excess bandwidth is the logical interface bandwidth minus the sum of the queue transmit rates, or 10 Mbps – 6 Mbps = 4 Mbps.
- However, transmission rates are capped at the shaping rate (PIR) of the queue. For example, Queue 0 gets 0.5 Mbps.
- Queue 0 also gets a guaranteed transmit rate (CIR) of 0.5 Mbps and is eligible for excess bandwidth calculated as 4 Mbps (10 Mbps – 6 Mbps) multiplied by 10/127.

However, because the shaping rate (PIR) for Queue 0 is 0.5 Mbps, the expected traffic rate is capped at 0.5 Mbps.

- Queue 1 gets its guaranteed transmit rate (CIR) of 3 Mbps. Because Queue 0 has already been dealt with, Queue 1 is eligible for sharing the excess bandwidth along with Queue 2 and Queue 3. So Queue 1 is entitled to an excess bandwidth of 4 Mbps multiplied by 50 / (30 + 30 + 50), or 1.81 Mbps.
- In the same way, Queue 2 is eligible for its guaranteed transmit rate (CIR) of 1 Mbps and an excess bandwidth of 4 Mbps multiplied by 30 / (30 + 30 + 50), or 1.09 Mbps. However, because Queue 2 has a shaping rate (PIR) of 1.5 Mbps, the bandwidth of Queue 2 is capped at 1.5 Mbps. The additional 0.59 Mbps can be shared by Queue 1 and Queue 3.
- Queue 3 is eligible for an excess of 4 Mbps multiplied by 30 / (30 + 30 + 50), or 1.09 Mbps. This total of 2.59 Mbps is still below the shaping rate (PIR) for Queue 3 (3.5 Mbps).
- The remaining bandwidth of 0.59 Mbps (which Queue 2 could not use) is shared between Queue 1 and Queue 3 in the ratio 50/30. So Queue 3 can get 0.59 multiplied by 30 / (50 + 30), or 0.22 Mbps. This gives a total of 2.81 Mbps.
- Therefore, Queue 1 gets 3 Mbps + 1.82 Mbps + (0.59 Mbps * 50 / (50 + 30)), or approximately 5.19 Mbps.

Logical Interface Modes on IQE PICs

On IQE PICs, scheduling occurs level-by-level. That is, based on the parameters configured on the logical interface, the scheduler first picks a logical interface to transmit from. Then, based on the configuration of the underlying queues, the IQE PIC selects one of the queues to transmit from. Therefore, it is important to understand how different logical interface parameters are configured or derived (not explicitly configured), and also how the same values are established at the queue level.

In the following examples, assume that the bandwidth available at the physical interface level is 400 Mbps and there are four logical interfaces (units) configured. A per-unit scheduler is configured, so the logical interfaces operate in different modes depending on the parameters configured.

If no class-of-service parameters are configured on any of the logical interfaces, the interface is in default mode. In default mode, the guaranteed rate (CIR) available at the physical interface (400 Mbps) is divided equally among the four logical interfaces. Each of the four gets a guaranteed rate (CIR) of 100 Mbps. Because none of the four logical interfaces have a shaping rate (PIR) configured, each logical interface can transmit up to the maximum of the entire 400 Mbps. Because there is no excess rate configured on any of the logical interfaces, each of the four gets an equal, minimum excess weight of 1. The configured and hardware-derived bandwidths for this default mode example are shown in Table 61 on page 302.

Table 61: Default Mode Example for IQE PICs

	Configured			Hardware		
Logical Interface	Guaranteed rate (CIR)	Shaping Rate (PIR)	Excess Rate	Guaranteed Rate	Shaping Rate	Excess Weight
Unit 0	NA	NA	NA	100 Mbps	400 Mbps	1
Unit 1	NA	NA	NA	100 Mbps	400 Mbps	1
Unit 2	NA	NA	NA	100 Mbps	400 Mbps	1
Unit 3	NA	NA	NA	100 Mbps	400 Mbps	1

If a subset of the logical interfaces (units) have a shaping rate (PIR) configured, but none of them have a guaranteed rate (CIR) or excess rate, then the physical interface is in PIR mode. Furthermore, if the sum of the shaping rates on the logical interfaces is less than or equal to the physical interface bandwidth, the physical interface is in undersubscribed PIR mode. If the sum of the shaping rates on the logical interfaces is more than the physical interface bandwidth, the physical interface is in oversubscribed PIR mode. These modes are the same as on other PICs, where only a shaping rate and guaranteed rate can be configured.

In undersubscribed PIR mode, the logical interfaces with a configured shaping rate receive preferential treatment over those without a configured shaping rate. For logical interfaces with a shaping rate configured, the guaranteed rate is set to the shaping rate. For the logical interfaces without a shaping rate, the remaining logical interface bandwidth is distributed equally among them. Excess weights for the logical interfaces with a shaping rate are set to an implementation-dependent value proportional to the shaping rate. Excess weights for the logical interfaces without a shaping rate are set to the minimum weight (1). However, although the excess weights for the configured logical interfaces are never used because the logical interfaces cannot transmit above their guaranteed rates, the excess weights are still determined for consistency with oversubscribed mode. Also, logical interfaces without a configured shaping rate can transmit up to a maximum of the physical bandwidth of the other queues that are not transmitting. Therefore, the shaping rate (PIR) is set to the physical interface bandwidth on these interfaces.

The configured and hardware-derived bandwidths for the undersubscribed PIR mode example are shown in Table 62 on page 302. Note that the sum of the shaping rates configured on the logical interfaces (500 Mbps) is more than the physical interface bandwidth (400 Mbps).

Table 62: Undersubscribed PIR Mode Example for IQE PICs

	Configured			Hardware		
Logical Interface	Guaranteed rate (CIR)	Shaping Rate (PIR)	Excess Rate	Guaranteed Rate	Shaping Rate	Excess Weight
Unit 0	NA	100 Mbps	NA	100 Mbps	100 Mbps	127

Table 62: Undersubscribed PIR Mode Example for IQE PICs *(continued)*

	Configured			Hardware		
Logical Interface	Guaranteed rate (CIR)	Shaping Rate (PIR)	Excess Rate	Guaranteed Rate	Shaping Rate	Excess Weight
Unit 1	NA	200 Mbps	NA	200 Mbps	200 Mbps	63
Unit 2	NA	NA	NA	50 Mbps	400 Mbps	1
Unit 3	NA	NA	NA	50 Mbps	400 Mbps	1

In the oversubscribed PIR mode, where the sum of the configured shaping rates on the logical interfaces exceeds the physical interface bandwidth, we cannot set the guaranteed rate to the shaping rate because this might result in the sum of the guaranteed rates exceeding the physical interface bandwidth, which is not possible. In this mode, we want the logical interfaces with shaping rates configured to share the traffic proportionally when these logical interfaces are transmitting at full capacity. This could not happen if the guaranteed rate was set to the shaping rate. Instead, in hardware, we set the guaranteed rates to a “scaled down” shaping rate, so that the sum of the guaranteed rates of the logical interfaces do not exceed the physical interface bandwidth. Because there is no remaining bandwidth once this is done, the other logical interfaces receive a guaranteed rate of 0. Excess weights are set proportionally to the shaping rates and for logical interfaces without a shaping rate, the excess weight is set to a minimum value (1). Finally, the shaping rate is set to the shaping rate configured on the logical interface or to the physical interface bandwidth otherwise.

NOTE: When the sum of shaping rate at a logical interface is greater than the interface's bandwidth and a rate limit is applied to one of the logical interface queues, the bandwidth limit for the queue is based on a scaled down logical interface shaping rate value rather than the configured logical interface shaping rate.

The configured and hardware-derived bandwidths for the oversubscribed PIR mode example are shown in Table 63 on page 303. Note that the sum of the shaping rates configured on the logical interfaces (300 Mbps) is less than the physical interface bandwidth (400 Mbps).

Table 63: Oversubscribed PIR Mode Example for IQE PICs

	Configured			Hardware		
Logical Interface	Guaranteed rate (CIR)	Shaping Rate (PIR)	Excess Rate	Guaranteed Rate	Shaping Rate	Excess Weight
Unit 0	NA	100 Mbps	NA	80 Mbps	100 Mbps	50
Unit 1	NA	150 Mbps	NA	120 Mbps	150 Mbps	76

Table 63: Oversubscribed PIR Mode Example for IQE PICs *(continued)*

Logical Interface	Configured			Hardware		
	Guaranteed rate (CIR)	Shaping Rate (PIR)	Excess Rate	Guaranteed Rate	Shaping Rate	Excess Weight
Unit 2	NA	250 Mbps	NA	200 Mbps	250 Mbps	127
Unit 3	NA	NA	NA	0 Mbps	400 Mbps	1

If none of the logical interfaces have an excess rate configured, but at least one of the logical interfaces has a guaranteed rate (CIR) configured, then the physical interface is in CIR mode. In this case, the guaranteed rates are set in hardware to the configured guaranteed rate on the logical interface. For logical interfaces that do not have a guaranteed rate configured, the guaranteed rate is set to 0. The hardware shaping rate is set to the value configured on the logical interface or to the full physical interface bandwidth otherwise. The excess weight is calculated proportional to the configured guaranteed rates. Logical interfaces without a configured guaranteed rate receive a minimum excess weight of 1.

The configured and hardware-derived bandwidths for the CIR mode example are shown in Table 64 on page 304. In CIR mode, the shaping rates are ignored in the excess weight calculations. So although logical unit 1 has an explicitly configured PIR and logical unit 3 does not, they both receive the minimum excess weight of 1.

Table 64: CIR Mode Example for IQE PICs

Logical Interface	Configured			Hardware		
	Guaranteed rate (CIR)	Shaping Rate (PIR)	Excess Rate	Guaranteed Rate	Shaping Rate	Excess Weight
Unit 0	50 Mbps	100 Mbps	NA	50 Mbps	100 Mbps	127
Unit 1	NA	150 Mbps	NA	0 Mbps	150 Mbps	1
Unit 2	100 Mbps	NA	NA	100 Mbps	400 Mbps	63
Unit 3	NA	NA	NA	0 Mbps	400 Mbps	1

If one of the logical interfaces has an excess rate configured, then the physical interface is in excess rate mode. Strictly speaking, this mode only matters for the calculation of excess weights on the logical interface. The hardware guaranteed and shaping rates are determined as described previously. In excess rate mode, the excess weights are set to a value based on the configured excess rate. Logical interfaces which do not have excess rates configured receive a minimum excess weight of 1.

NOTE: Because the excess rate only makes sense above the guaranteed rate, you cannot configure an excess rate in PIR mode (PIR mode has only shaping rates configured). You must configure at least one guaranteed rate (CIR) on a logical interface to configure an excess rate.

The excess rate is configured as a percentage in the range from 1 through 100. The configured value is used to determine the excess weight in the range from 1 through 127.

The configured and hardware-derived bandwidths for the excess rate mode example are shown in Table 65 on page 305. When an excess rate is configured on one or more logical interfaces, the shaping rate and the guaranteed rate are both ignored in the excess weight calculations. So logical unit 2 gets a minimum excess weight of 1, even though it has a guaranteed rate configured.

Table 65: Excess Rate Mode Example for IQE PICs

	Configured			Hardware		
Logical Interface	Guaranteed rate (CIR)	Shaping Rate (PIR)	Excess Rate	Guaranteed Rate	Shaping Rate	Excess Weight
Unit 0	50 Mbps	100 Mbps	20%	50 Mbps	100 Mbps	50
Unit 1	NA	150 Mbps	50%	0 Mbps	150 Mbps	127
Unit 2	100 Mbps	NA	NA	100 Mbps	400 Mbps	1
Unit 3	NA	NA	50%	0 Mbps	400 Mbps	127

Default Rates for Queues on IQE PICs

The IQE PIC operates at the queue level as well as at the logical unit level. This section discusses how the IQE PIC derives hardware values from the user configuration parameters. First, the default behavior without explicit configuration is investigated, along with the rules used to derive hardware parameters from the scheduler map configuration of the transmit rate, shaping rate, and excess rate. For more information about configuring schedulers and scheduler maps, see “Schedulers Overview” on page 146.

When you do not configure any CoS parameters, a default scheduler map is used to establish four queues: best-effort, expedited-forwarding, assured-forwarding, and network-control. Each queue has the default transmit rate, shaping rate, and excess rate shown in Table 66 on page 305.

Table 66: Default Queue Rates on the IQE PIC

Queue	Transmit Rate	Shaping Rate	Excess Rate
best-effort (Q0)	95%	100%	95%
expedited-forwarding (Q1)	0%	100%	0%

Table 66: Default Queue Rates on the IQE PIC *(continued)*

Queue	Transmit Rate	Shaping Rate	Excess Rate
assured-forwarding (Q2)	0%	100%	0%
network-control (Q3)	5%	100%	5%

When you configure a scheduler map to change the defaults, the IQE PIC hardware derives the values for each of the three major parameters: transmit rate, shaping rate, and excess rate.

The transmit rate is determined as follows:

- If a transmit rate is configured, then:
  - If the transmit rate is configured as an absolute bandwidth value, the configured value is used by the hardware.
  - If the transmit rate is configured as a percentage, then the percentage is used to calculate an absolute value used by the hardware, based on the guaranteed rate (CIR) configured at the logical interface or physical interface level. The CIR itself can be a default, configured, or derived value.
  - If the transmit rate is configured as a remainder, then the remaining value of the logical interface (unit) guaranteed rate (CIR) is divided equally among the queues configured as remainder.
- If a transmit rate is not configured, then the default transmit rate is derived based on remainder (for backward compatibility).
- If an excess rate is configured on any of the queues in a scheduler map, then the transmit rate on the queue is set to 0.

The shaping rate is determined as follows:

- If a shaping rate is configured:
  - If the shaping rate is configured as an absolute bandwidth value, the configured value is used by the hardware.
  - If the shaping rate is configured as a percentage, then the percentage is used to calculate an absolute value used by the hardware, based on the guaranteed rate (CIR) configured at the logical interface or physical interface level. Although it seems odd to base a shaping rate (PIR) on the CIR instead of a PIR, this is done so the shaping rate can be derived on the same basis as the transmit rate.
- If a shaping rate is not configured, then the default shaping rate is set to the shaping rate configured at the logical interface or physical interface level.

The excess rate is determined as follows:

- If an excess rate is configured on a queue, the value is used to derive an excess weight used by the IQE PIC hardware. The excess weight determines the proportional share of the excess bandwidth for which each queue can contend. The excess rate can be:
  - Percentage in the range from 1 through 100. This value is scaled to a hardware excess weight. Excess rates can add up to more than 100% for all queues under a logical or physical interface.
- If an excess rate is not configured on a queue, then the default excess rate is one of the following:
  - If a transmit rate is configured on any of the queues, then the excess weight is proportional to the transmit rates. Queues that do not have a transmit rate configured receive a minimum weight of 1.
  - If a transmit rate is not configured on any of the queues, but some queues have a shaping rate, then the excess weight is proportional to the shaping rates. Queues that do not have a shaping rate configured receive a minimum weight of 1.
  - If no parameters are configured on a queue, then the queue receives a minimum weight of 1.

Sample Calculations of Excess Bandwidth Sharing on IQE PICs

The following four examples show calculations for the PIR mode. In PIR mode, the transmit rate and shaping rate calculations are based on the shaping rate of the logical interface. All calculations assume that one logical interface (unit) is configured with a shaping rate (PIR) of 10 Mbps and a scheduler map with four queues.

The first example has only a shaping rate (PIR) configured on the queues, as shown in Table 67 on page 307.

Table 67: PIR Mode, with No Excess Configuration

Queue	Transmit Rate (CIR)	Shaping Rate (PIR)	Excess Rate	Traffic Sent To Queue
Q0	NA	80%	NA	10 Mbps
Q1	NA	50%	NA	1 Mbps
Q2	NA	40%	NA	0 Mbps
Q3	NA	30%	NA	5 Mbps

The way that the IQE PIC hardware interprets these parameters is shown in Table 68 on page 308.

Table 68: PIR Mode, with No Excess Hardware Behavior

Queue	Transmit Rate	Shaping Rate	Excess Weight	Expected Output Rate
Q0	2.5 Mbps	8.0 Mbps	50	6 Mbps
Q1	2.5 Mbps	5.0 Mbps	31	1 Mbps
Q2	2.5 Mbps	4.0 Mbps	25	0 Mbps
Q3	2.5 Mbps	3.0 Mbps	19	3 Mbps

In this first example, all four queues are initially serviced round-robin. Because there are no transmit rates configured on any of the queues, they receive a default "remainder" transmit rate of 2.5 Mbps per queue. But because there are shaping rates configured, the excess weights are calculated based on the shaping rates. For the traffic sent to each queue, Queue 0 and Queue 3 get their transmit rates of 2.5 Mbps and Queue 1 gets 1 Mbps. The remaining 4 Mbps is excess bandwidth and is divided between Queue 0 and Queue 3 in the ratio of the shaping rates (80/30). So Queue 3 expects an excess bandwidth of 4 Mbps * (30% / (80% + 30%)) = 1.09 Mbps. However, because the shaping rate on Queue 3 is 3 Mbps, Queue 3 can transmit only 3 Mbps and Queue 0 receives the remaining excess bandwidth and can transmit at 6 Mbps.

Note that if there were equal transmit rates explicitly configured, such as 2.5 Mbps for each queue, the excess bandwidth would be split based on the transmit rate (equal in this case), as long as the result in below the shaping rate for the queue.

The second example has a shaping rate (PIR) and transmit rate (CIR) configured on the queues, as shown in Table 69 on page 308.

Table 69: PIR Mode with Transmit Rate Configuration

Queue	Transmit Rate (CIR)	Shaping Rate (PIR)	Excess Rate	Traffic Sent To Queue
Q0	50%	80%	NA	10 Mbps
Q1	40%	50%	NA	5 Mbps
Q2	10%	20%	NA	5 Mbps
Q3	NA	5%	NA	1 Mbps

The way that the IQE PIC hardware interprets these parameters is shown in Table 70 on page 308.

Table 70: PIR Mode with Transmit Rate Hardware Behavior

Queue	Transmit Rate	Shaping Rate	Excess Weight	Expected Output Rate
Q0	5.0 Mbps	8.0 Mbps	63	5 Mbps

Table 70: PIR Mode with Transmit Rate Hardware Behavior *(continued)*

Queue	Transmit Rate	Shaping Rate	Excess Weight	Expected Output Rate
Q1	4.0 Mbps	5.0 Mbps	50	4 Mbps
Q2	1.0 Mbps	2.0 Mbps	12	1 Mbps
Q3	0.0 Mbps	0.5 Mbps	1	0.0 Mbps

In this second example, because the transmit rates are less than the shaping rates, each queue receives its transmit rate.

The third example also has a shaping rate (PIR) and transmit rate (CIR) configured on the queues, as shown in Table 71 on page 309.

Table 71: Second PIR Mode with Transmit Rate Configuration Example

Queue	Transmit Rate (CIR)	Shaping Rate (PIR)	Excess Rate	Traffic Sent To Queue
Q0	50%	80%	NA	10 Mbps
Q1	40%	50%	NA	5 Mbps
Q2	5%	20%	NA	0 Mbps
Q3	NA	5%	NA	1 Mbps

The way that the IQE PIC hardware interprets these parameters is shown in Table 72 on page 309.

Table 72: Second PIR Mode with Transmit Rate Hardware Behavior Example

Queue	Transmit Rate	Shaping Rate	Excess Weight	Expected Output Rate
Q0	5.0 Mbps	8.0 Mbps	66	5.27 Mbps
Q1	4.0 Mbps	5.0 Mbps	53	4.23 Mbps
Q2	0.5 Mbps	2.0 Mbps	13	0.0 Mbps
Q3	0.5 Mbps	0.5 Mbps	1	0.5 Mbps

In this third example, all four queues are initially serviced round-robin. However, Queue 2 has no traffic sent to its queue. So Queue 0, Queue 1, and Queue 3 all get their respective transmit rates, a total of 9.5 Mbps. The remaining 0.5 Mbps is used by Queue 3, because the transmit rate is the same as the shaping rate. Once this traffic is sent, Queue 0 and Queue 1 share the excess bandwidth in the ratio of their transmit rates, which total 9 Mbps. In this case, Queue 0 = 5 Mbps + (0.5 Mbps * 5/9) = 5.27 Mbps. Queue 1 = 4 Mbps + (0.5 Mbps * 4/9) = 4.23 Mbps.

The fourth example has a shaping rate (PIR), transmit rate (CIR), and excess rate configured on the queues, as shown in Table 73 on page 310.

Table 73: PIR Mode with Transmit Rate and Excess Rate Configuration

Queue	Transmit Rate (CIR)	Shaping Rate (PIR)	Excess Rate	Traffic Sent To Queue
Q0	30%	80%	50%	10 Mbps
Q1	25%	50%	10%	5 Mbps
Q2	10%	20%	30%	0 Mbps
Q3	5%	5%	NA	1 Mbps

The way that the IQE PIC hardware interprets these parameters is shown in Table 74 on page 310.

Table 74: PIR Mode with Transmit Rate and Excess Rate Hardware Behavior

Queue	Transmit Rate	Shaping Rate	Excess Weight	Expected Output Rate
Q0	3.0 Mbps	8.0 Mbps	70	6.33 Mbps
Q1	2.5 Mbps	5.0 Mbps	14	3.17 Mbps
Q2	1.0 Mbps	2.0 Mbps	42	0.0 Mbps
Q3	0.5 Mbps	0.5 Mbps	1	0.5 Mbps

In this fourth example, all four queues are initially serviced round-robin. Queue 3 gets 0.5 Mbps of guaranteed bandwidth but cannot transmit more because the shaping rate is the same. Queue 2 has no traffic to worry about at all. Queue 0 and Queue 1 get the respective transmit rates of 3.0 Mbps and 2.5 Mbps. The excess bandwidth of 4 Mbps is divided between Queue 0 and Queue 1 in the ratio on their excess rates. So Queue 1 gets 2.5 Mbps (the guaranteed rate) + 4 Mbps (the excess) + (10% / (50% + 10%)) = 3.17 Mbps. Queue 0 gets the rest, for a total of 6.33 Mbps.

You can configure only an excess rate on the queues, as shown in Table 75 on page 310.

Table 75: Excess Rate Configuration

Queue	Transmit Rate (CIR)	Shaping Rate (PIR)	Excess Rate	Traffic Sent To Queue
Q0	NA	NA	50%	10 Mbps
Q1	NA	NA	40%	10 Mbps
Q2	NA	NA	30%	10 Mbps
Q3	NA	NA	20%	10 Mbps

The way that the IQE PIC hardware interprets these excess rate parameters is shown in Table 76 on page 311.

Table 76: Excess Rate Hardware Behavior

Queue	Transmit Rate	Shaping Rate	Excess Weight	Expected Output Rate
Q0	0 Mbps	10.0 Mbps	45	3.57 Mbps
Q1	0 Mbps	10.0 Mbps	40	2.86 Mbps
Q2	0 Mbps	10.0 Mbps	30	2.14 Mbps
Q3	0 Mbps	10.0 Mbps	20	1.43 Mbps

In this excess rate example, there are no transmit or shaping rates configured on any of the queues, only excess rates, so bandwidth division happens only on the basis of the excess rates. Note that all the transmit (guaranteed) rates are set to 0. Usually, when there are no excess rates configured, the queue transmit rate is calculated by default. But when there is an excess rate configured on any of the queues, the transmit rate is set to 0. The excess bandwidth (all bandwidths in this case) is shared in the ratio of the excess weights. So Queue 0 receives 10 Mbps * (50 / (50 + 40+ 30+ 20)) = 3.57 Mbps.

It is possible to configure rate limits that result in error conditions. For example, consider the configuration shown in Table 77 on page 311.

Table 77: PIR Mode Generating Error Condition

Queue	Transmit Rate (CIR)	Shaping Rate (PIR)	Excess Rate	Traffic Sent To Queue
Q0	NA	80%	NA	10 Mbps
Q1	NA	50%	NA	5 Mbps
Q2	NA	20%	NA	5 Mbps
Q3	NA	5%	NA	1 Mbps

The way that the IQE PIC hardware interprets these parameters is shown in Table 78 on page 311.

Table 78: PIR Mode Generating Error Condition Behavior

Queue	Transmit Rate	Shaping Rate	Excess Weight	Expected Output Rate
Q0	2.5 Mbps	8.0 Mbps	818	4.03 Mbps
Q1	2.5 Mbps	5.0 Mbps	511	3.47 Mbps
Q2	2.5 Mbps	2.0 Mbps	255	2 Mbps

Table 78: PIR Mode Generating Error Condition Behavior *(continued)*

Queue	Transmit Rate	Shaping Rate	Excess Weight	Expected Output Rate
Q3	2.5 Mbps	0.5 Mbps	51	0.1 Mbps

In the error example, note that the shaping rates calculated on Queue 2 and Queue 3 are less than the transmit rates on those queues (2.0 Mbps and 0.5 Mbps are each less than 2.5 Mbps). This is an error condition and results in a syslog error message.

The following set of five examples involve the IQE PIC operating in CIR mode. In CIR mode, the transmit rate and shaping rate calculations are based on the transmit rate of the logical interface. All calculations assume that the logical interface has a shaping rate (PIR) of 20 Mbps and a transmit rate (CIR) of 10 Mbps. The scheduler map has four queues.

The first example has only a shaping rate (PIR) with no excess rate configured on the queues, as shown in Table 79 on page 312.

Table 79: CIR Mode with No Excess Rate Configuration

Queue	Transmit Rate (CIR)	Shaping Rate (PIR)	Excess Rate	Traffic Sent To Queue
Q0	NA	80%	NA	10 Mbps
Q1	NA	70%	NA	10 Mbps
Q2	NA	40%	NA	10 Mbps
Q3	NA	30%	NA	10 Mbps

NOTE: The transmit rate (CIR) of 10 Mbps is configured on the logical interface (unit) not the queues in the scheduler map. This is why the queue transmit rates are labeled NA.

The way that the IQE PIC hardware interprets these parameters is shown in Table 80 on page 312.

Table 80: CIR Mode with No Excess Rate Hardware Behavior

Queue	Transmit Rate	Shaping Rate	Excess Weight	Expected Output Rate
Q0	2.5 Mbps	8.0 Mbps	50	6.76 Mbps
Q1	2.5 Mbps	7.0 Mbps	31	6.23 Mbps
Q2	2.5 Mbps	4.0 Mbps	25	4.0 Mbps
Q3	2.5 Mbps	3.0 Mbps	19	3.0 Mbps

In this first example, all four queues split the 10-Mbps transmit rate equally and each get a transmit rate of 2.5 Mbps. However, the shaping rate on the interface is 20 Mbps. The 10-Mbps excess bandwidth is divided among the queues in the ratio of their shaping rates. But Queue 2 and Queue 3 are shaped at 3.0 and 4.0 Mbps, respectively, so they cannot use more bandwidth and get those rates. This accounts for 2 Mbps (the 7 Mbps shaped bandwidth minus the 5 Mbps guaranteed bandwidth for Queue 2 and Queue 3) of the 10-Mbps excess, leaving 8 Mbps for Queue 0 and Queue 1. So Queue 0 and Queue 1 share the 8-Mbps excess bandwidth in the ratio of their shaping rates, which total 15 Mbps. In this case, Queue 0 = 8.0 Mbps * 8/15 = 4.26 Mbps, for a total of 2.5 Mbps + 4.26 Mbps = 6.76 Mbps. Queue 1 = 8.0 Mbps * 7/15 = 3.73 Mbps, for a total of 2.5 Mbps + 3.73 Mbps = 6.23 Mbps.

The second example has only a few shaping rates (PIR) with no excess rate configured on the queues, as shown in Table 81 on page 313.

Table 81: CIR Mode with Some Shaping Rates and No Excess Rate Configuration

Queue	Transmit Rate (CIR)	Shaping Rate (PIR)	Excess Rate	Traffic Sent To Queue
Q0	NA	80%	NA	10 Mbps
Q1	NA	50%	NA	5 Mbps
Q2	NA	NA	NA	10 Mbps
Q3	NA	NA	NA	1 Mbps

NOTE: If a configuration results in the calculated transmit rate of the queue exceeding the shaping rate of the queue, an error message is generated. For example, setting the shaping rate on Queue 2 and Queue 3 in the above example to 20 percent and 5 percent, respectively, generates an error message because the calculated transmit rate for these queues (2.5 Mbps) is more than their calculated shaping rates (2.0 Mbps and 0.5 Mbps).

The way that the IQE PIC hardware interprets these parameters is shown in Table 82 on page 313.

Table 82: CIR Mode with Some Shaping Rates and No Excess Rate Hardware Behavior

Queue	Transmit Rate	Shaping Rate	Excess Weight	Expected Output Rate
Q0	2.5 Mbps	8.0 Mbps	78	8.0 Mbps
Q1	2.5 Mbps	5.0 Mbps	48	5.0 Mbps
Q2	2.5 Mbps	20 Mbps	1	6.0 Mbps
Q3	2.5 Mbps	20 Mbps	1	1.0 Mbps

In this second example, all four queues split the 10-Mbps transmit rate equally and each get a transmit rate of 2.5 Mbps. Because of their configured queue shaping rates, Queue 0 and Queue 1 receive preference over Queue 2 and Queue 3 for the excess bandwidth. Queue 0 (8.0 Mbps) and Queue 1 (5.0 Mbps) account for 13 Mbps of the 20 Mbps shaping rate on the logical interface. The remaining 7 Mbps is divided equally between Queue 2 and Queue 3. However, because Queue 3 only has 1 Mbps to send, Queue 2 uses the remaining 6 Mbps.

The third example has shaping rates (PIR) and transmit rates with no excess rate configured on the queues, as shown in Table 83 on page 314.

Table 83: CIR Mode with Shaping Rates and Transmit Rates and No Excess Rate Configuration

Queue	Transmit Rate (CIR)	Shaping Rate (PIR)	Excess Rate	Traffic Sent To Queue
Q0	50%	80%	NA	10 Mbps
Q1	40%	50%	NA	5 Mbps
Q2	10%	20%	NA	5 Mbps
Q3	NA	10%	NA	1 Mbps

The way that the IQE PIC hardware interprets these parameters is shown in Table 84 on page 314.

Table 84: CIR Mode with Shaping Rates and Transmit Rates and No Excess Rate Hardware Behavior

Queue	Transmit Rate	Shaping Rate	Excess Weight	Expected Output Rate
Q0	5.0 Mbps	8.0 Mbps	63	8.0 Mbps
Q1	4.0 Mbps	5.0 Mbps	50	5.0 Mbps
Q2	1.0 Mbps	2.0 Mbps	12	2.0 Mbps
Q3	0.0 Mbps	0.5 Mbps	1	0.5 Mbps

In this third example, the first three queues get their configured transmit rates and are serviced in round-robin fashion. This adds up to 10 Mbps, leaving a 10-Mpbs excess from the logical interface shaping rate of 20 Mbps. The excess is shared in the ratio of the transmit rates, or 5:4:1:0. Therefore, Queue 0 receives 5 Mbps + (5 * 10/10) = 10 Mbps. This value is greater than the 8 Mbps shaping rate on Queue 0, so Queue 0 is limited to 8 Mbps. Queue 1 receives 4 Mbps + (4 * 10/10) = 8 Mbps. This value is greater than the 5 Mbps shaping rate on Queue 1, so Queue 1 is limited to 5 Mbps. Queue 2 receives 1 Mbps + (1 * 10/10) = 2 Mbps. This value is equal to the 2 Mbps shaping rate on Queue 2, so Queue 2 receives 2 Mbps. This still leaves 5 Mbps excess bandwidth, which can be used by Queue 3. Note that in this example bandwidth usage never reaches the shaping rate configured on the logical interface (20 Mbps).

The fourth example has shaping rates (PIR) and transmit rates with no excess rate configured on the queues. However, in this case the sum of the shaping rate percentages configured on the queues multiplied by the transmit rate configured on the logical interface is greater than the shaping rate configured on the logical interface. The configuration is shown in Table 85 on page 315.

Table 85: CIR Mode with Shaping Rates Greater Than Logical Interface Shaping Rate Configuration

Queue	Transmit Rate (CIR)	Shaping Rate (PIR)	Excess Rate	Traffic Sent To Queue
Q0	50%	80%	NA	10 Mbps
Q1	40%	70%	NA	10 Mbps
Q2	10%	50%	NA	10 Mbps
Q3	NA	50%	NA	10 Mbps

The way that the IQE PIC hardware interprets these parameters is shown in Table 86 on page 315.

Table 86: CIR Mode with Shaping Rates Greater Than Logical Interface Shaping Rate Hardware Behavior

Queue	Transmit Rate	Shaping Rate	Excess Weight	Expected Output Rate
Q0	5.0 Mbps	8.0 Mbps	63	8.0 Mbps
Q1	4.0 Mbps	7.0 Mbps	50	7.0 Mbps
Q2	1.0 Mbps	5.0 Mbps	12	5.0 Mbps
Q3	0.0 Mbps	5.0 Mbps	1	0.0 Mbps

In this fourth example, the first three queues get their configured transmit rates and are serviced in round-robin fashion. This adds up to 10 Mbps, leaving a 10-Mpbs excess from the logical interface shaping rate of 20 Mbps. The excess is shared in the ratio of the transmit rates, or 5:4:1:0. Therefore, Queue 0 receives 5 Mbps + (5 * 10/10) = 10 Mbps. This value is greater than the 8 Mbps shaping rate on Queue 0, so Queue 0 is limited to 8 Mbps. Queue 1 receives 4 Mbps + (4 * 10/10) = 8 Mbps. This value is greater than the 7 Mbps shaping rate on Queue 1, so Queue 1 is limited to 7 Mbps. Queue 2 receives 1 Mbps + (1 * 10/10) = 2 Mbps. This value is less than the 5 Mbps shaping rate on Queue 2, so Queue 2 receives 2 Mbps. This still leaves 3 Mbps excess bandwidth, which can be used by Queue 2 (below its shaping rate) and Queue 3 (also below its shaping rate) in the ratio 1:0 (because of the transmit rate configuration). But 1:0 means Queue 3 cannot use this bandwidth, and Queue 2 utilizes 2 Mbps + (3 Mbps * 1/1) = 5 Mbps. This is equal to the shaping rate of 5 Mbps, so Queue 2 receives 5 Mbps.

The fifth example has excess rates and transmit rates, but no shaping rates (PIR) configured on the queues. The configuration is shown in Table 87 on page 316.

Table 87: CIR Mode with Excess Rate Configuration

Queue	Transmit Rate (CIR)	Shaping Rate (PIR)	Excess Rate	Traffic Sent To Queue
Q0	30%	NA	50%	10 Mbps
Q1	25%	NA	10%	10 Mbps
Q2	NA	NA	30%	10 Mbps
Q3	10%	NA	NA	10 Mbps

The way that the IQE PIC hardware interprets these parameters is shown in Table 88 on page 316.

Table 88: CIR Mode with Excess Rate Hardware Behavior

Queue	Transmit Rate	Shaping Rate	Excess Weight	Expected Output Rate
Q0	3.0 Mbps	20 Mbps	70	10.5 Mbps
Q1	2.5 Mbps	20 Mbps	14	4.0 Mbps
Q2	0.0 Mbps	20 Mbps	42	4.5 Mbps
Q3	1.0 Mbps	20 Mbps	1	1.0 Mbps

In this fifth example, Queue 2 does not have a transmit rate configured. If there were no excess rates configured, then Queue 2 would get a transmit rate equal to the remainder of the bandwidth (3.5 Mbps in this case). However, because there is an excess rate configured on some of the queues on this logical interface, the transmit rate for Queue 2 is set to 0 Mbps. The others queues get their transmit rates and there leaves 13.5 Mbps of excess bandwidth. This bandwidth is divided among Queue 0, Queue 1, and Queue 3 in the ratio of their excess rates. So Queue 0, for example, gets 3.0 Mbps + 13.5 Mbps * (50 / (50 + 10 + 30)) = 10.5 Mbps.

Four other examples calculating expected traffic distribution are of interest. The first case has three variations, so there are six more examples in all.

- Oversubscribed PIR mode at the logical interface with transmit rates, shaping rates, and excess rates configured at the queues (this example has three variations).
- CIR mode at the logical interface (a non-intuitive case is used).
- Excess priority configured.
- Default excess priority used.

The first three examples all concern oversubscribed PIR mode at the logical interface with transmit rates, shaping rates, and excess rates configured at the queues. They all use a configuration with a physical interface having a shaping rate of 40 Mbps. The physical interface has two logical units configured, logical unit 1 and logical unit 2, with a shaping rate of 30 Mbps and 20 Mbps, respectively. Because the sum of the logical interface shaping rates is more than the shaping rate on the physical interface, the physical interface is in oversubscribed PIR mode. The CIRs (transmit rates) are set to the scaled values of 24 Mbps and 16 Mbps, respectively.

Assume that logical unit 1 has 40 Mbps of traffic to be sent. The traffic is capped at 30 Mbps because of the shaping rate of 30 Mbps. Because the CIR is scaled down to 24 Mbps, the remaining 6 Mbps (30 Mbps – 24 Mbps) qualifies as excess bandwidth.

The following three examples consider different parameters configured in a scheduler map and the expected traffic distributions that result.

The first example uses oversubscribed PIR mode with only transmit rates configured on the queues. The configuration is shown in Table 89 on page 317.

Table 89: Oversubscribed PIR Mode with Transmit Rate Configuration

Queue	Transmit Rate (CIR)	Shaping Rate (PIR)	Excess Rate	Traffic Sent To Queue
Q0	40%	NA	NA	15 Mbps
Q1	30%	NA	NA	10 Mbps
Q2	25%	NA	NA	10 Mbps
Q3	5%	NA	NA	5 Mbps

The way that the IQE PIC hardware interprets these parameters is shown in Table 90 on page 317.

Table 90: Oversubscribed PIR Mode with Transmit Rate Hardware Behavior

Queue	Transmit Rate	Shaping Rate	Excess Weight	Expected Output Rate
Q0	9.6 Mbps	30 Mbps	50	12 Mbps
Q1	7.2 Mbps	30 Mbps	38	9 Mbps
Q2	6.0 Mbps	30 Mbps	31	7.5 Mbps
Q3	1.2 Mbps	30 Mbps	6	1.5 Mbps

The first example has hardware queue transmit rates based on the parent (logical interface unit 1) transmit rate (CIR) value of 24 Mbps. Because there are no excess rates configured, the excess weights are determined by the transmit rates. Therefore, both the logical interface CIR and excess bandwidth are divided in the ratio of the transmit rates. This is essentially the same as the undersubscribed PIR mode and the traffic distribution should be the same. The only difference is that the result is achieved as a combination of guaranteed rate (CIR) and excess rate sharing.

The second example also uses oversubscribed PIR mode, but this time with only excess rate configured on the queues. In other words, the same ratios are established with excess rate percentages instead of transmit rate percentages. In this case, when excess rates are configured, queues without a specific transmit rate are set to 0 Mbps. So the entire bandwidth qualifies as excess at the queue level and the bandwidth distribution is based on the configured excess rates. The expected output rate results are exactly the same as in the first example, except the calculation is based on different parameters.

The third example also uses oversubscribed PIR mode, but with both transmit rates and excess rates configured on the queues. The configuration is shown in Table 91 on page 318.

Table 91: Oversubscribed PIR Mode with Transmit Rate and Excess Rate Configuration

Queue	Transmit Rate (CIR)	Shaping Rate (PIR)	Excess Rate	Traffic Sent To Queue
Q0	40%	NA	50%	15 Mbps
Q1	30%	NA	50%	12 Mbps
Q2	25%	NA	NA	8 Mbps
Q3	5%	NA	NA	5 Mbps

The way that the IQE PIC hardware interprets these parameters is shown in Table 92 on page 318.

Table 92: Oversubscribed PIR Mode with Transmit Rate and Excess Rate Hardware Behavior

Queue	Transmit Rate	Shaping Rate	Excess Weight	Expected Output Rate
Q0	9.6 Mbps	30 Mbps	63	12.6 Mbps
Q1	7.2 Mbps	30 Mbps	63	10.2 Mbps
Q2	6.0 Mbps	30 Mbps	1	6.0 Mbps
Q3	1.2 Mbps	30 Mbps	1	1.2 Mbps

The third example has the configured queue transmit rate (CIR) divided according to the ratio of the transmit rates based on the logical interface unit 1 CIR of 25 Mbps. The rest of the excess bandwidth divided according the ratio of the excess rates. The excess 6-Mbps bandwidth is divided equally between Queue 0 and Queue 1 because the excess rates are both configured at 50%. This type of configuration is not recommended, however, because the CIR on the logical interface is a system-derived value based on the PIRs of the other logical units and the traffic distribution at the queue level is based on this value and, therefore, not under direct user control. We recommend that you either configure excess rates without transmit rates at the queue level when in PIR mode, or also define a CIR at the logical interface if you want to configure a combination of transmit rates and excess rates at the queue level. That is, you should use configurations of the CIR mode with excess rates types.

The fourth example uses CIR mode at the logical interface. For this example, assume that a physical interface is configured with a 40-Mbps shaping rate and logical interfaces unit 1 and unit 2. Logical interface unit 1 has a PIR of 30 Mbps and logical interface unit 2 has a PIR of 20 Mbps and a CIR of 10 Mbps. The configuration at the queue level of logical interface unit 1 is shown in Table 93 on page 319.

Table 93: CIR Mode with Transmit Rate and Excess Rate Configuration

Queue	Transmit Rate (CIR)	Shaping Rate (PIR)	Excess Rate	Traffic Sent To Queue
Q0	40%	NA	50%	15 Mbps
Q1	30%	NA	50%	12 Mbps
Q2	25%	NA	NA	8 Mbps
Q3	5%	NA	NA	5 Mbps

The way that the IQE PIC hardware interprets these parameters is shown in Table 94 on page 319.

Table 94: CIR Mode with Transmit Rate and Excess Rate Hardware Behavior

Queue	Transmit Rate	Shaping Rate	Excess Weight	Expected Output Rate
Q0	0 Mbps	30 Mbps	63	15 Mbps
Q1	0 Mbps	30 Mbps	63	12 Mbps
Q2	0 Mbps	30 Mbps	1	1.5 Mbps
Q3	0 Mbps	30 Mbps	1	1.5 Mbps

The fourth example might be expected to divide the 40 Mbps of traffic between the two logical units in the ratio of the configured transmit rates. But note that because the logical interfaces are in CIR mode, and logical interface unit 1 does not have a CIR configured, the hardware CIR is set to 0 Mbps at the queue level. Bandwidth distribution happens based only on the excess weights. So Queue 0 and Queue 1 get to transmit up to 15 Mbps and 12 Mbps, respectively, while the remaining 3 Mbps is divided equally by Queue 2 and Queue 3.

NOTE: We recommend configuring a CIR value explicitly for the logical interface if you are configuring transmit rates and excess rates for the queues.

The fifth example associates an excess priority with the queues. Priorities are associated with every queue and propagated to the parent node (logical or physical interface). That is, when the scheduler picks a logical interface, the scheduler considers the logical interface priority as the priority of the highest priority queue under that logical interface. On the IQE PIC, you can configure an excess priority for every queue. The excess priority can differ from the priority used for guaranteed traffic and applies only to traffic in the excess region. The IQE PIC has three "regular" priorities and two excess priorities (high and low, which is the default). The excess priorities are lower than the regular priorities. For more information about configuring excess bandwidth sharing and priorities, see "IQE PIC Excess Bandwidth Sharing Configuration" on page 297.

Consider a logical interface configured with a shaping rate of 10 Mbps and a guaranteed rate of 10 Mbps. At the queue level, parameters are configured as shown in Table 95 on page 320.

Table 95: Excess Priority Configuration

Queue	Transmit Rate (CIR)	Shaping Rate (PIR)	Excess Rate	Traffic Sent To Queue
Q0	40%	NA	50%	10 Mbps
Q1	30%	NA	50%	10 Mbps
Q2	25%	NA	NA	0 Mbps
Q3	5%	NA	NA	1 Mbps

In this fifth example, Queue 0 is configured with an excess priority of **high** and all other queues have the default excess priority (**low**). Because there is no traffic on Queue 2, there is an excess bandwidth of 2.5 Mbps. Because Queue 0 has a higher excess priority, Queue 0 gets the entire excess bandwidth. So the expected output rates on the queues are 4 Mbps+ 2.5 Mbps= 6.5 Mbps for Queue 0, 3 Mbps for Queue 1, 0 Mbps for Queue 2, and 0.5 Mbps for Queue 3. Note that this behavior is different than regular priorities. With regular priorities, the transmission is still governed by transmit rates and the priority controls only the order in which the packets are picked up by the scheduler. So without excess configuration, if Queue 0 had a regular priority of **high** and there was 10 Mbps of traffic on all four queues, the traffic distribution would be 4 Mbps for Queue 0, 3 Mbps for Queue 1, 2.5 Mbps for Queue 2, and 0.5 Mbps for Queue 3 instead of giving all 10 Mbps to Queue 0. Excess priority traffic distributions are governed first by the excess priority and then by the excess rates. Also note that in this example, although the queues are in the excess region because they are transmitting above their configured transmit rates, the logical interface is still within its guaranteed rate. So at the logical interface level, the priority of the queues get promoted to a regular priority and this priority is used by the scheduler at the logical interface level.

The sixth and final example considers the effects of the default excess priority. When the excess priority for a queue is not configured explicitly, the excess priority is based on the regular priority. A regular priority of **high** maps to an excess priority of **high**. All other regular priorities map to an excess priority of **low**. When there is no regular priority configured, the regular and excess priorities are both set to **low**.

Configuring Layer 2 Policing on IQE PICs

The IQE PIC can police traffic at Layer 2 in a hierarchical manner. *Policing* is the practice of making sure that different streams of incoming traffic conform to certain parameters and limits. If the incoming traffic exceeds the established boundaries, that traffic can be marked or even ignored, depending on configuration. Hierarchical policing maintains two rates: an aggregate rate and a high-priority rate. The traffic is marked differently depending on service class (currently, the classes are expedited forwarding and nonexpedited forwarding). The expedited traffic has an additional rate configured, the guaranteed rate (CIR), which is only marked above that limit. If there is no expedited traffic present, then the non-expedited traffic is able to use the aggregate bandwidth rate before being marked with a packet loss priority. When expedited traffic is present, it is marked above the guaranteed rate, but also uses bandwidth from the nonexpedited range.

For example, consider an aggregate rate of 10 Mbps and a high-priority rate of 2 Mbps of a Fast Ethernet interface. The guaranteed rate is also set at 2 Mbps for expedited forwarding traffic. If there is no expedited traffic present, then nonexpedited traffic can use up to 10 Mbps before being marked. When expedited forwarding traffic is present, the expedited traffic is guaranteed 2 Mbps (of the 10 Mbps) without being marked, but is marked above the 2 Mbps limit. In this case, the nonexpedited forwarding traffic can use the remaining 8 Mbps before being marked.

This section discusses the following IQE PIC Layer 2 policing topics:

- Layer 2 Policer Limitations on page 321
- Configuring Layer 2 Policers on IQE PICs on page 322

Layer 2 Policer Limitations

Layer 2 policers configured on IQE PICs have the following limitations:

- Only one kind of policer is supported on a physical or logical interface. For example, a hierarchical or two- or three-color policer in the same direction on the same logical interface is not supported.
- Applying policers to both physical port and logical interface (policer chaining) is not supported.
- If there is no behavior aggregate classification, there is a limit of 64 policers per interface. (Usually, there will be a single policer per DLCI in frame relay and other logical interface types.)
- The policer should be independent of behavior aggregate classification. (Without a behavior aggregate, all traffic is treated as either expedited or non-expedited forwarding, depending on configuration.)
- With a behavior aggregate, traffic not matching any classification bits (such as DSCP or EXP) is policed as nonexpedited forwarding traffic.
- Only two levels of traffic policing are supported: **aggregate** and **premium**.

Configuring Layer 2 Policers on IQE PICs

To configure Layer 2 policing on the IQE PIC, for each forwarding class include the **class** statement with the **policing-priority** option at the **[edit class-of-service forwarding-classes]** hierarchy level. One forwarding class has the **premium** option and the others are configured as **normal**.

```
[edit class-of-service forwarding-classes]
{
  class fc1 queue-num 0 priority high policing-priority premium;
  class fc2 queue-num 1 priority low policing-priority normal;
  class fc3 queue-num 2 priority low policing-priority normal;
  class fc4 queue-num 3 priority low policing-priority normal;
}
```

You must also configure the **aggregate** and **premium** statements in the firewall filter performing the policing.

```
[edit firewall]
hierarchical-policer hier_example1 {
  aggregate {
    if-exceeding {
      bandwidth-limit 70m;
      burst-size-limit 1800;
    }
    then {
      discard;
    }
  }
  premium {
    if-exceeding {
      bandwidth-limit 70m;
      burst-size-limit 3600;
    }
    then {
      discard;
    }
  }
}
```

You must also apply the policer to the logical or physical interface on the IQE PIC:

```
[edit interfaces]
so-6/0/0 {
  unit 0 {
    layer2-policer {
      input-hierarchical-policer hier_example1; # Apply policer to logical unit.
    }
    family inet {
      address 10.0.22.1/30;
    }
    family iso;
    family mpls;
  }
}
```



```
so-5/0/0 {
  layer2-policer {
    input-hierarchical-policer hier_example1; # Apply policer to physical interface.
  }
  unit 0 {
    family inet {
      address 10.0.22.1/30;
    }
    family iso;
    family mpls;
  }
}
```

For SONET/SDH physical interfaces, the hierarchical policer configuration statements will only be visible for IQE PICs.

Configuring Low-Latency Static Policers on IQE PICs

You can rate-limit the strict-high and high queues on the IQE PIC. Without this limiting, traffic that requires low latency (delay) such as voice can block the transmission of medium-priority and low-priority packets. Unless limited, high and strict-high traffic is always sent before lower priority traffic, causing the lower priority queues to “starve” and cause timeouts and unnecessarily resent packets.

On the IQE PIC you can rate-limit queues before the packets are queued for output. All packets exceeding the configured rate limit are dropped, so care is required when establishing this limit. This model is also supported on IQ2 PICs and is the only way to perform egress policing on IQE PICs. This feature introduces no new configuration statements.

Although intended for low-latency traffic classes such as voice, the configuration allows any queue to be rate-limited. However, the configuration requires the rate-limited queue to have either a high or strict-high priority.

NOTE: You can configure a low-latency static policer for only one rate-limited queue per scheduler map. You can configure up to 1024 low-latency static policers.

This example limits the transmit rate of a strict-high expedited-forwarding queue to 1 Mbps. The scheduler and scheduler map are defined, and then applied to the traffic at the **[edit interfaces]** and **[edit class-of-service]** hierarchy levels:

```
[edit class-of-service]
schedulers {
  scheduler-1 {
    transmit-rate 1m rate-limit;
    priority strict-high;
  }
}
scheduler-maps {
  scheduler-map-1 {
    forwarding-class expedited-forwarding scheduler scheduler-1;
```

```
    }
}

[edit interfaces]
s0-2/0/0 {
    per-unit-scheduler;
    encapsulation frame-relay;
    unit 0 {
        dlci 1;
    }
}

[edit class-of-service]
interfaces {
    so-2/0/0 {
        unit 0 {
            scheduler-map scheduler-map-1;
            shaping-rate 2m;
        }
    }
}
```

You can issue the following operational mode commands to verify your configuration (the first shows the rate limit in effect):

- **show class-of-service scheduler-map** ***scheduler-map-name***
- **show class-of-service interface** ***interface-name***

CHAPTER 19

Configuring CoS on Ethernet IQ2 and Enhanced IQ2 PICs

This topic discusses the following:

CoS on Enhanced IQ2 PICs Overview

Some PICs, such as the Gigabit Ethernet Intelligent Queuing 2 (IQ2) and Ethernet Enhanced IQ2 (IQ2E) PICs, have eight egress queues enabled by default on platforms that support eight queues.

The IQ2E PICs preserve all of the features of the IQ2 PICs, such as the default support for eight egress queues on platforms that support eight queues.

The IQ2E PICs add features such as the ability to perform hierarchical scheduling. You can mix IQ2 and IQ2E PICs on the same router.

The IQ2E PICs offer:

- Three levels of hierarchical CoS
- More granularity than a high priority queue

- 16,000 queues
- 2,000 schedulers with 8 queues
- 4,000 schedulers with 4 queues

The IQ2E PICs also offer automatic scheduler allocation across ports, so there is no need to reset the PIC when this changes. Random early detection (RED) keeps statistics on a per-drop-profile basis, improving the ability to perform network capacity planning.

When you include the **per-unit-scheduler** statement at the **[edit interfaces *interface-name*]** hierarchy level, each logical interface (unit) gets a dedicated scheduler (one scheduler is reserved for overflow). You can include the **per-session-scheduler** statement at the **[edit interfaces *interface-name* unit *logical-unit-number*]** hierarchy level to shape Layer 2 Tunneling Protocol (L2TP) sessions. The behavior of these two-port scheduler modes is the same as in IQ2 PICs. However, IQ2E PICs use hierarchical schedulers and not shared schedulers; IQ2E PICs do not support the **shared-scheduler** statement at the **[edit interfaces *interface-name*]** hierarchy level.

For more information about configuring hierarchical schedulers, including examples, see “Configuring Hierarchical Schedulers for CoS” on page 207.

You can shape traffic at the physical interface (port), logical interface (unit), or interface set (set of units) levels. Shaping is not supported at the queue level. However, you can include the **transmit-rate** statement with the **rate-limit** option at the **[edit class-of-service schedulers *scheduler-name*]** hierarchy level to police the traffic passing through a queue (but only in the egress direction). See “Configuring Rate Limits on IQ2 and Enhanced IQ2 PICs” on page 327.

At the physical interface (port) level, you can configure only a shaping rate (PIR). At the logical interface (unit) and interface set levels, you can configure both a shaping rate and a guaranteed rate (CIR). Note that the guaranteed rates at any level must be consistent with the parent level’s capacity. In other words, the sum of the guaranteed rates on the logical interface (units) should be less than the guaranteed rate on the interface set, and the sum of the guaranteed rates on the logical interface (units) and interface sets should be less than the guaranteed rate on the physical interface (port).

The weighed RED (WRED) decision on the IQ2E PICs is done at the queue level. Once the accept or drop decision is made and the packet is queued, it is never dropped. Four drop profiles are associated with each queue: low, low-medium, medium-high, and high. WRED statistics are available for each loss priority (this feature is not supported on the IQ2 PICs). Also in contrast to the IQ2 PICs, the IQ2E PICs support WRED scaling profiles, allowing a single drop profile to be reused with a wide range of values. This practice increases the effective number of WRED drop profiles.

The IQ2E PICs provide four levels of strict priorities: strict-high, high, medium-high (medium-low) and low. In contrast to the IQ2 PICs, which support only one strict-high queue, the IQ2E PICs do not restrict the number of queues with a given priority. There is priority propagation among three levels: the logical interface, the logical interface set, and the physical port. These features are the same as those supported by Enhanced Queueing Dense Port Concentrators (DPCs) for Juniper Network MX Series Ethernet

Services Routers. For more information about configuring these features, see "Enhanced Queuing DPC Hardware Properties" on page 355.

The IQ2E PIC's queues are serviced with modified deficit round-robin (MDRR), as with the Enhanced Queueing DPCs. Excess bandwidth (bandwidth available after all guaranteed rates have been satisfied) can be shared equally or in proportion to the guaranteed rates. For more information about excess bandwidth sharing, see "Configuring Excess Bandwidth Sharing" on page 364.

Setting the Number of Egress Queues on IQ2 and Enhanced IQ2 PICs

Gigabit Ethernet IQ2 4-port and 8-port Type 2 PICs are oversubscribed, which means the amount of traffic coming to the PIC can be more than the maximum bandwidth from the PIC to the Flexible PIC Concentrator (FPC).

By default, PICs on M320, MX Series, and T Series routers support a maximum of four egress queues per interface. Some PICs, such as the IQ2 and IQ2E PICs, have eight egress queues enabled by default on platforms that support eight queues. You configure the number of egress queues as four or eight by including the **max-queues-per-interface** statement at the **[edit chassis fpc *slot-number* pic *pic-slot-number*]** hierarchy level:

```
[edit chassis fpc slot-number pic pic-slot-number]
max-queues-per-interface (4 | 8);
```

The numerical value can be **4** or **8**.

For more information about configuring egress queues, see "Enabling Eight Queues on Interfaces" on page 123.

Configuring Rate Limits on IQ2 and Enhanced IQ2 PICs

You can rate-limit strict-high and high queues on IQ2 and IQ2E PICs. Without this limiting, traffic in higher priority queues can block the transmission of lower priority packets. Unless limited, higher priority traffic is always sent before lower priority traffic, causing the lower priority queues to "starve," which in turn leads to timeouts and unnecessary resending of packets.

On the IQ2 and IQ2E PICs you can rate-limit queues before the packets are queued for output. All packets exceeding the configured rate limit are dropped, so care is required when establishing this limit. For more information about configuring CoS on IQ2E PICs, see "CoS on Enhanced IQ2 PICs Overview" on page 325.

To rate-limit queues, include the **transmit-rate** statement with the **rate-limit** option at the **[edit class-of-service schedulers *scheduler-name*]** hierarchy level:

```
[edit class-of-service schedulers scheduler-name]
transmit-rate rate rate-limit;
```

This example limits the transmit rate of a strict-high expedited-forwarding queue to 1 megabit per second (Mbps). The scheduler and scheduler map are defined and then applied to the traffic at the **[edit interfaces]** and **[edit class-of-service]** hierarchy levels:

```
[edit class-of-service]
schedulers {
```

```
    scheduler-1 {
        transmit-rate 1m rate-limit; # This establishes the limit
        priority strict-high;
    }
}
scheduler-maps {
    scheduler-map-1 {
        forwarding-class expedited-forwarding scheduler scheduler-1;
    }
}

[edit interfaces]
s0-2/1/0 {
    per-unit-scheduler;
    encapsulation frame-relay;
    unit 0 {
        dlci 1;
    }
}

[edit class-of-service]
interfaces {
    so-2/1/0 {
        unit 0 {
            scheduler-map scheduler-map-1;
            shaping-rate 2m;
        }
    }
}
```

You can issue the following operational mode commands to verify your configuration (the first shows the rate limit in effect):

- **show class-of-service scheduler-map *scheduler-map-name***
- **show class-of-service interface *interface-name***

Configuring Shaping on 10-Gigabit Ethernet IQ2 PICs

The 10-Gigabit Ethernet IQ2 PIC (which has **xe-** interfaces) is unlike other Gigabit Ethernet IQ2 PICs in that it does not have oversubscription. The bandwidth from the PIC to the FPC is sufficient to transmit the full line rate. However, the 10-Gigabit Ethernet IQ2 PIC has the same hardware architecture as other Gigabit Ethernet IQ2 PICs and supports all the same class-of-service (CoS) features. For more information, see the PIC guide for your routing platform.

To handle oversubscribed traffic, you can configure input shaping and scheduling based on Layer 2, MPLS, and Layer 3 packet fields. Gigabit Ethernet IQ2 PICs also support simple filters, accounting, and policing. This chapter discusses input and output shaping and scheduling. For information about simple filters, see “Overview of Simple Filters” on page 79 and the *Junos OS Policy Framework Configuration Guide*. For information about accounting and policing, see the *Junos OS Network Interfaces Configuration Guide*.

NOTE: The CoS functionality supported on Gigabit Ethernet IQ2 PICs is not available across aggregated Ethernet links. However, if you configure a CoS scheduler map on the link bundle, the configuration is honored by the individual links within that bundle.

Therefore, CoS behaves as configured on a per-link level, but not across the aggregated links. For example, if you configure a shaping transmit rate of 100 Mbps on an aggregated Ethernet bundle with three ports (by applying a scheduler for which the configuration includes the **transmit-rate** statement with the **exact** option at the **[edit class-of-service schedulers *scheduler-name*]** hierarchy level), each port is provisioned with a 33.33 Mbps shaping transmit rate.

You can configure shaping for aggregated Ethernet interfaces that use interfaces originating from Gigabit Ethernet IQ2 PICs. However, you cannot enable shaping on aggregated Ethernet interfaces when the aggregate bundle combines ports from IQ and IQ2 PICs.

By default, transmission 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. The default operation can be changed by configuring the software.

NOTE: For Gigabit Ethernet IQ2 interfaces, the logical interface egress statistics displayed in the **show interfaces** command output might not accurately reflect the traffic on the wire when output shaping is applied. Traffic management output shaping might drop packets after they are tallied by the **Output bytes** and **Output packets** logical interface counters. However, correct values display for both of these **Transit statistics** when per-unit scheduling is enabled for the Gigabit Ethernet IQ2 physical interface, or when a single logical interface is actively using a shared scheduler.

To configure input and output shaping and scheduling, include the following statements at the **[edit class-of-service]** and **[edit interfaces]** hierarchy levels of the configuration:

```
[edit class-of-service]
traffic-control-profiles profile-name {
  delay-buffer-rate (percent percentage | rate);
  excess-rate percent percentage;
  guaranteed-rate (percent percentage | rate);
  scheduler-map map-name;
  shaping-rate (percent percentage | rate);
}
interfaces {
  interface-name {
    input-scheduler-map map-name;
    input-shaping-rate rate;
```



```
      scheduler-map map-name; # Output scheduler map
      shaping-rate rate; # Output shaping rate
      }
      unit logical-unit-number {
        input-scheduler-map map-name;
        input-shaping-rate (percent percentage | rate);
        scheduler-map map-name;
        shaping-rate (percent percentage | rate);
        input-traffic-control-profile profile-name shared-instance instance-name;
        output-traffic-control-profile profile-name shared-instance instance-name;
      }
    }
  }

  [edit interfaces interface-name]
  per-unit-scheduler;
  shared-scheduler;
```

NOTE: As indicated by the preceding configuration, the **scheduler-map** and **shaping-rate** statements can be included at the **[edit class-of-service interfaces *interface-name* unit *logical-unit-number*]** hierarchy level. However, we do not recommend this configuration. Include the **output-traffic-control-profile** statement instead.

Shaping Granularity Values for Enhanced Queuing Hardware

Due to the limits placed on shaping thresholds used in the hierarchy, there is a granularity associated with the Enhanced IQ2 (IQ2E) PIC and the Enhanced Queuing (EQ) DPC. For these hardware models, 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 the port level (Level 1). Table 96 on page 330 shows the accuracy of the logical interface shaper at various rates for Ethernet ports operating at 1 Gbps.

Table 96: 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
65.535 to 131.072 Mbps	512 Kbps

Table 96: Shaper Accuracy of 1-Gbps Ethernet at the Logical Interface Level *(continued)*

Range of Logical Interface Shaper	Step Granularity
131.072 to 262.144 Mbps	1024 Kbps
262.144 to 1 Gbps	4096 Kbps

Table 97 on page 331 shows the accuracy of the logical interface shaper at various rates for Ethernet ports operating at 10 Gbps.

Table 97: 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

Table 98 on page 331 shows the accuracy of the interface set shaper at various rates for Ethernet ports operating at 1 Gbps.

Table 98: 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 99 on page 332 shows the accuracy of the interface set shaper at various rates for Ethernet ports operating at 10 Gbps.

Table 99: 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

Table 100 on page 332 shows the accuracy of the physical port shaper at various rates for Ethernet ports operating at 1 Gbps.

Table 100: 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 101 on page 332 shows the accuracy of the physical port shaper at various rates for Ethernet ports operating at 10 Gbps.

Table 101: 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

Differences Between Gigabit Ethernet IQ and Gigabit Ethernet IQ2 PICs

Because Gigabit Ethernet IQ PICs and Gigabit Ethernet IQ2 PICs use different architectures, they differ in the following ways:

- Gigabit Ethernet IQ2 PICs support a transmission rate within a queue, but do not support an exact rate within a queue. You can apply a weight to a queue, but you cannot put an upper limit on the queue transmission rate that is less than the logical interface can support. Consequently, including the **exact** option with the **transmit-rate (*rate* | percent *percent*)** statement at the **[edit class-of-service schedulers *scheduler-name*]** hierarchy level is not supported for Gigabit Ethernet IQ2 interfaces.
- Gigabit Ethernet IQ2 PICs support only one queue in the scheduler map with **medium-high**, **high**, or **strict-high** priority. If more than one queue is configured with **high** or **strict-high** priority, the one that appears first in the configuration is implemented as **strict-high** priority. This queue receives unlimited transmission bandwidth. The remaining queues are implemented as **low** priority, which means they might be starved.
- To ensure that protocol control traffic (such as OSPF, BGP, and RIP) are not dropped at the oversubscribed ingress direction, the software puts control protocol packets into a separate control scheduler. There is one control scheduler per port. These control schedulers are implemented as strict-high priority, so they transmit traffic until they are empty.
- On Gigabit Ethernet IQ2 PICs, you can configure a single traffic-control profile to contain both a PIR (the **shaping-rate** statement) and a CIR (the **guaranteed-rate** statement). On Gigabit Ethernet IQ PICs, these statements are mutually exclusive.
- Gigabit Ethernet IQ2 PICs support only two fill levels in the RED drop profile. The recommended definition of the RED drop profile is as follows:

  ```
  class-of-service {
    drop-profiles {
      drop-iq2-example1 {
        fill-level 20 drop-probability 0;
        fill-level 100 drop-probability 80;
      }
    }
  }
  ```

  This configuration defines a drop profile with a linear drop probability curve when the fill level is between 20 and 100 percent, and a maximum drop probability of 80 percent.

You can configure more than two fill levels in a drop profile, but the software only uses the points (**min_fill_level**, **0**) and (**max_fill_level**, **max_probability**) and ignores other fill levels. The drop probability at the minimum fill level is set to 0 percent even if you configure a non-zero drop probability value at the minimum fill level. The following example shows a sample configuration and the software implementation:

Configuration

```
class-of-service {
  drop-profiles {
    drop-iq2-example2 {
      fill-level 30 drop-probability 10;
      fill-level 40 drop-probability 20;
```

```
            fill-level 100 drop-probability 80;
        }
    }
}
```

Implementation

```
class-of-service {
    drop-profiles {
        drop-iq2-example2-implementation {
            fill-level 30 drop-probability 0;
            fill-level 100 drop-probability 80;
        }
    }
}
```

If you configure more than two fill levels, a system log message warns you that the software supports only two fill levels and displays the drop profile that is implemented.

Though the **interpolate** statement is supported in the definition of a RED drop profile, we do not recommend using it. The following example shows a sample configuration and the software implementation:

Configuration

```
class-of-service {
    drop-profiles {
        drop-iq2-example3 {
            interpolate {
                fill-level [ 30 50 80 ];
                drop-probability [ 10 20 40 ];
            }
        }
    }
}
```

When you use the **interpolate** statement and the maximum fill level is not 100 percent, the software adds the point (100, 100). Therefore, the drop-iq2-example3 drop profile is implemented as:

Implementation

```
class-of-service {
    drop-profiles {
        drop-iq2-example3-implementation {
            fill-level 2 drop-probability 0;
            fill-level 100 drop-probability 100;
        }
    }
}
```

The implemented minimum fill level is not 30 percent as configured, but 2 percent because of the 64-point interpolation.

Configuring Traffic Control Profiles for Shared Scheduling and Shaping

Shared scheduling and shaping allows you to allocate separate pools of shared resources to subsets of logical interfaces belonging to the same physical port. You configure this by first creating a traffic-control profile, which specifies a shaping rate and references a scheduler map. You must then share this set of shaping and scheduling resources by applying an instance of the traffic-control profile to a subset of logical interfaces. You can apply a separate instance of the same (or a different) traffic-control profile to another subset of logical interfaces, thereby allocating separate pools of shared resources.

To configure a traffic-control profile, perform the following steps:

1. Include the **shaping-rate** statement at the **[edit class-of-service traffic-control-profiles *profile-name*]** hierarchy level:

   ```
   [edit class-of-service traffic-control-profiles profile-name]
   shaping-rate (percent percentage | rate);
   ```

   You can configure the shaping rate as a percentage from 1 through 100 or as an absolute rate from 1000 through 160,000,000,000 bits per second (bps). The shaping rate corresponds to a peak information rate (PIR). For more information, see “Oversubscribing Interface Bandwidth” on page 184.

2. Include the **scheduler-map** statement at the **[edit class-of-service traffic-control-profiles *profile-name*]** hierarchy level:

   ```
   [edit class-of-service traffic-control-profiles profile-name]
   scheduler-map map-name;
   ```

   For information about configuring schedulers and scheduler maps, see “Configuring Schedulers” on page 148 and “Configuring Scheduler Maps” on page 167. Gigabit Ethernet IQ2 interfaces support up to eight forwarding classes and queues.

3. Include the **delay-buffer-rate** statement at the **[edit class-of-service traffic-control-profiles *profile-name*]** hierarchy level:

   ```
   [edit class-of-service traffic-control-profiles profile-name]
   delay-buffer-rate (percent percentage | rate);
   ```

   You can configure the delay-buffer rate as a percentage from 1 through 100 or as an absolute rate from 1000 through 160,000,000,000 bits per second. The delay-buffer rate controls latency. For more information, see “Oversubscribing Interface Bandwidth” on page 184 and “Providing a Guaranteed Minimum Rate” on page 191.

4. Include the **guaranteed-rate** statement at the **[edit class-of-service traffic-control-profiles *profile-name*]** hierarchy level:

   ```
   [edit class-of-service traffic-control-profiles profile-name]
   guaranteed-rate (percent percentage | rate);
   ```

   You can configure the guaranteed rate as a percentage from 1 through 100 or as an absolute rate from 1000 through 160,000,000,000 bps. The guaranteed rate corresponds to a committed information rate (CIR). For more information, see “Providing a Guaranteed Minimum Rate” on page 191.

   You must now share an instance of the traffic-control profile.

To share an instance of the traffic-control profile, perform the following steps:

1. Include the **shared-scheduler** statement at the **[edit interfaces *interface-name*]** hierarchy level:

   ```
   [edit interfaces interface-name]
   shared-scheduler;
   ```

This statement enables logical interfaces belonging to the same physical port to share one set of shaping and scheduling resources.

NOTE: On each physical interface, the **shared-scheduler** and **per-unit-scheduler** statements are mutually exclusive. Even so, you can configure one logical interface for each shared instance. This effectively provides the functionality of per-unit scheduling.

2. To apply the traffic-control profile to an input interface, include the **input-traffic-control-profile** and **shared-instance** statements 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*]
   input-traffic-control-profile *profile-name* **shared-instance** *instance-name*;

   These statements are explained in Step 3.

3. To apply the traffic-control profile to an output interface, include the **output-traffic-control-profile** and **shared-instance** statements 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*]
   output-traffic-control-profile *profile-name* **shared-instance** *instance-name*;

   The profile name references the traffic-control profile you configured in Step 1 through Step 4 of the "Configuring Traffic Control Profiles for Shared Scheduling and Shaping" section. The shared-instance name does not reference a configuration. It can be any text string you wish to apply to multiple logical interfaces that you want to share the set of resources configured in the traffic-control profile. Each logical interface shares a set of scheduling and shaping resources with other logical interfaces that are on the same physical port and that have the same shared-instance name applied.

   This concept is demonstrated in "Example: Configuring Shared Resources on Ethernet IQ2 Interfaces" on page 343.

NOTE: You cannot include the **output-traffic-control-profile** statement in the configuration if any of the following statements are included in the logical interface configuration: **scheduler-map**, **shaping-rate**, **adaptive-shaper**, or **virtual-channel-group** (the last two are valid on J Series routers only).

Differences Between Gigabit Ethernet IQ and Gigabit Ethernet IQ2 PICs

Because Gigabit Ethernet IQ PICs and Gigabit Ethernet IQ2 PICs use different architectures, they differ in the following ways:

- Gigabit Ethernet IQ2 PICs support a transmission rate within a queue, but do not support an exact rate within a queue. You can apply a weight to a queue, but you cannot put an upper limit on the queue transmission rate that is less than the logical interface can support. Consequently, including the **exact** option with the **transmit-rate (*rate* | percent *percent*)** statement at the **[edit class-of-service schedulers *scheduler-name*]** hierarchy level is not supported for Gigabit Ethernet IQ2 interfaces.
- Gigabit Ethernet IQ2 PICs support only one queue in the scheduler map with **medium-high**, **high**, or **strict-high** priority. If more than one queue is configured with **high** or **strict-high** priority, the one that appears first in the configuration is implemented as **strict-high** priority. This queue receives unlimited transmission bandwidth. The remaining queues are implemented as **low** priority, which means they might be starved.
- To ensure that protocol control traffic (such as OSPF, BGP, and RIP) are not dropped at the oversubscribed ingress direction, the software puts control protocol packets into a separate control scheduler. There is one control scheduler per port. These control schedulers are implemented as strict-high priority, so they transmit traffic until they are empty.
- On Gigabit Ethernet IQ2 PICs, you can configure a single traffic-control profile to contain both a PIR (the **shaping-rate** statement) and a CIR (the **guaranteed-rate** statement). On Gigabit Ethernet IQ PICs, these statements are mutually exclusive.
- Gigabit Ethernet IQ2 PICs support only two fill levels in the RED drop profile. The recommended definition of the RED drop profile is as follows:

  ```
  class-of-service {
    drop-profiles {
      drop-iq2-example1 {
        fill-level 20 drop-probability 0;
        fill-level 100 drop-probability 80;
      }
    }
  }
  ```

  This configuration defines a drop profile with a linear drop probability curve when the fill level is between 20 and 100 percent, and a maximum drop probability of 80 percent.

You can configure more than two fill levels in a drop profile, but the software only uses the points (**min_fill_level**, **0**) and (**max_fill_level**, **max_probability**) and ignores other fill levels. The drop probability at the minimum fill level is set to 0 percent even if you configure a non-zero drop probability value at the minimum fill level. The following example shows a sample configuration and the software implementation:

Configuration

```
class-of-service {
  drop-profiles {
    drop-iq2-example2 {
      fill-level 30 drop-probability 10;
      fill-level 40 drop-probability 20;
```

```
            fill-level 100 drop-probability 80;
        }
    }
}
```

Implementation

```
class-of-service {
    drop-profiles {
        drop-iq2-example2-implementation {
            fill-level 30 drop-probability 0;
            fill-level 100 drop-probability 80;
        }
    }
}
```

If you configure more than two fill levels, a system log message warns you that the software supports only two fill levels and displays the drop profile that is implemented.

Though the **interpolate** statement is supported in the definition of a RED drop profile, we do not recommend using it. The following example shows a sample configuration and the software implementation:

Configuration

```
class-of-service {
    drop-profiles {
        drop-iq2-example3 {
            interpolate {
                fill-level [ 30 50 80 ];
                drop-probability [ 10 20 40 ];
            }
        }
    }
}
```

When you use the **interpolate** statement and the maximum fill level is not 100 percent, the software adds the point (100, 100). Therefore, the drop-iq2-example3 drop profile is implemented as:

Implementation

```
class-of-service {
    drop-profiles {
        drop-iq2-example3-implementation {
            fill-level 2 drop-probability 0;
            fill-level 100 drop-probability 100;
        }
    }
}
```

The implemented minimum fill level is not 30 percent as configured, but 2 percent because of the 64-point interpolation.

Configuring a Separate Input Scheduler for Each Interface

As an alternative to shared input traffic-control profiles, you can configure each interface to use its own input scheduler. For each physical interface, you can apply an input scheduler map to the physical interface or its logical interfaces, but not both.

For information about configuring schedulers and scheduler maps, see "Configuring Schedulers" on page 148 and "Configuring Scheduler Maps" on page 167. Gigabit Ethernet IQ2 interfaces support up to eight forwarding classes and queues.

To configure a separate input scheduler on the physical interface, include the **input-scheduler-map** statement at the **[edit class-of-service interfaces *interface-name*]** hierarchy level:

```
[edit class-of-service interfaces interface-name]
input-scheduler-map map-name;
```

To configure a separate input scheduler on a logical interface, perform the following steps:

1. Include the **input-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]
   input-scheduler-map map-name;
   ```

2. For the corresponding physical interface, you must also include the **per-unit-scheduler** statement at the **[edit interfaces *interface-name*]** hierarchy level:

   ```
   [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.

On Gigabit Ethernet IQ2 PIC interfaces, configuration of the **per-unit-scheduler** statement requires that you configure VLAN tagging also. When you include the **per-unit-scheduler** 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.

Configuring Per-Unit Scheduling for GRE Tunnels Using IQ2 and IQ2E PICs

M7i, M10i, M120, M320, T Series, and TX Matrix routers with Intelligent Queuing 2 (IQ2) PICs and Intelligent Queuing 2 Enhanced (IQ2E) PICs support per unit scheduling for Generic Routing Encapsulation (GRE) tunnels, adding all the functionality of tunnel PICs to GRE tunnels. The class of service (CoS) for the GRE tunnel traffic will be applied as the traffic is looped through the IQ2 and IQ2E PIC.

Shaping is performed on full packets that pass through the GRE tunnel.

IQ2 and IQ2E PICs support all interfaces that are supported on tunnel PICs, as follows:

- **gr-*fpc/pic/port***
- **vt-*fpc/pic/port***
- **lt-*fpc/pic/port***
- **ip-*fpc/pic/port***
- **pe-*fpc/pic/port***

- **pd-*fpc/pic/port***
- **mt-*fpc/pic/port***

The port variable is always zero.

The provided tunnel functionality is the same as that of regular tunnel PICs.

You can specify that IQ2 and IQ2E PICs work exclusively in tunnel mode or as a regular PIC. The default setting uses IQ2 and IQ2E PICs as a regular PIC. To configure exclusive tunnel mode, use the **tunnel-only** statement at the **[edit chassis fpc *slot-number* pic *slot-number* tunnel-services]** hierarchy level.

You can use the **show interfaces queue gr-*fpc/pic/port*** command to display statistics for the specified tunnel.

IQ2E PIC schedulers can be dynamically allocated across ports.

When IQ2 and IQ2E PICs work exclusively as a tunnel PIC, they support the same number of tunnel logical interfaces as regular tunnel PICs; for example each PIC can support 4K **gr-** logical interfaces.

NOTE: This feature supports only **traffic-control-profile** on **gr-** logical interfaces. It does not support Class of Service (CoS) on **gr-** logical interfaces.

Also, a scheduler is allocated for a gr- logical interface only when there is a traffic-control profile configured for it.

The **tunnel-only** statement is used to specify that the IQ2 or IQ2E PIC needs to work in tunnel mode, as follows:

```
[edit]
chassis {
  fpc slot-number {
    pic slot-number {
      tunnel-services {
        tunnel-only;
      }
    }
  }
}
```

The PIC will be automatically bounced when the tunnel services configuration is changed.

The **chassis traffic-manager** mode must have the ingress traffic manager enabled in order for the tunnel-services to work correctly.

On **gr-** interfaces, you can configure the **output-traffic-control-profile** statement at the **[edit class-of-service interfaces gr-*fpc/pic/port* unit *logical-unit-number*]** hierarchy level:

```
[edit]
class-of-service {
  traffic-control-profiles {
    tcp {
```



```
            shaping-rate rate;
        }
    }
    interfaces {
        gr-fpc/pic/port {
            unit logical-unit-number {
                output-traffic-control-profile tcp
            }
        }
    }
}
```

The **gr-** logical interfaces without an explicit CoS configuration are not assigned a dedicated scheduler. These use a reserved scheduler meant for all unshaped tunnel traffic; that is, all traffic on **gr-** logical interfaces that do not have CoS configured and all traffic from other types of tunnels.

For more information on **chassis tunnel-services** configuration, see the *Junos OS System Basics Configuration Guide*.

To view the configuration and statistics for GRE tunnel logical interfaces, use the **show interfaces queue gr-** command. For more information, see the *Junos OS Interfaces Command Reference*.

Configuring Hierarchical Input Shapers

You can apply input shaping rates to both the physical interface and its logical interfaces. The rate specified at the physical level is distributed among the logical interfaces based on their input shaping-rate ratio.

To configure an input shaper on the physical interface, include the **input-shaping-rate** statement at the **[edit class-of-service interfaces *interface-name*]** hierarchy level:

```
[edit class-of-service interfaces interface-name]
input-shaping-rate rate;
```

To configure an input shaper on the logical interface, include the **input-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]
input-shaping-rate (percent percentage | rate);
```

For each logical interface, you can specify a percentage of the physical rate or an actual rate. The software converts actual rates into percentages of the physical rate.

Example: Configuring a CIR and a PIR on Ethernet IQ2 Interfaces

On Gigabit Ethernet IQ2 interfaces, you can configure a CIR (guaranteed rate) and a PIR (shaping rate) on a single logical interface. The configured rates are gathered into a traffic control profile. If you configure a traffic control profile with a CIR (guaranteed rate) only, the PIR (shaping rate) is set to the physical interface (port) rate.

In the following example, logical unit 0 has a CIR equal to 30 Mbps and a PIR equal to 200 Mbps. Logical unit 1 has a PIR equal to 300 Mbps. Logical unit 2 has a CIR equal to 100 Mbps and a PIR that is unspecified. For logical unit 2, the software causes the PIR to be 1 Gbps (equal to the physical interface rate) because the PIR must be equal to or greater than the CIR.

Excess bandwidth is the leftover bandwidth on the port after meeting all the guaranteed rate requirements of the logical interfaces. For each port, excess bandwidth is shared as follows:

- Proportional to the guaranteed rate—This method is used if you configure one or more logical interfaces on a port to have a guaranteed rate.
- Proportional to the shaping rate—This method is used if you configure none of the logical interfaces on a port to have a guaranteed rate.

In this example, bandwidth is shared proportionally to the guaranteed rate because at least one logical interface has a guaranteed rate.

```
class-of-service {
  traffic-control-profiles {
    profile1 {
      shaping-rate 200m;
      guaranteed-rate 30m;
      delay-buffer-rate 150m;
      scheduler-map sched-map;
    }
    profile2 {
      shaping-rate 300m;
      delay-buffer-rate 500k;
      scheduler-map sched-map;
    }
    profile3 {
      guaranteed-rate 100m;
      scheduler-map sched-map;
    }
  }
  interfaces {
    ge-3/0/0 {
      unit 0 {
        output-traffic-control-profile profile1;
      }
      unit 1 {
        output-traffic-control-profile profile2;
      }
      unit 2 {
        output-traffic-control-profile profile3;
      }
    }
  }
}
```


Example: Configuring Shared Resources on Ethernet IQ2 Interfaces

For input traffic on physical interface **ge-1/2/3**, logical interface units **1**, **2**, and **3** are sharing one set of scheduler-shaper resources, defined by traffic-control profile **s1**. Logical interface units **4**, **5**, and **6** are sharing another set of scheduler-shaper resources, defined by traffic-control profile **s1**.

For output traffic on physical interface **ge-1/2/3**, logical interface units **1**, **2**, and **3** are sharing one set of scheduler-shaper resources, defined by traffic-control profile **s2**. Logical interface units **4**, **5**, and **6** are sharing another set scheduler-shaper resources, defined by traffic-control profile **s2**.

For each physical interface, the **shared-instance** statement creates one set of resources to be shared among units **1**, **2**, and **3** and another set of resources to be shared among units **4**, **5**, and **6**. Input and output shaping rates are configured at the physical interface level, which demonstrates the hierarchical shaping capability of the Gigabit Ethernet IQ2 PIC.

```
[edit]
class-of-service {
  traffic-control-profiles {
    s1 {
      scheduler-map map1;
      shaping-rate 100k;
    }
    s2 {
      scheduler-map map1;
      shaping-rate 200k;
    }
  }
  forwarding-classes { # Map one forwarding class to one queue.
    queue 0 fc-be;
    queue 1 fc-be1;
    queue 2 fc-ef;
    queue 3 fc-ef1;
    queue 4 fc-af11;
    queue 5 fc-af12;
    queue 6 fc-nc1;
    queue 7 fc-nc2;
  }
  classifiers { # Map 802.1p bits to forwarding-class and loss-priority.
    ieee-802.1 ieee-8021p-table {
      forwarding-class fc-nc2 {
        loss-priority low code-points [111];
      }
      forwarding-class fc-nc1 {
        loss-priority low code-points [110];
      }
      forwarding-class fc-af12 {
        loss-priority low code-points [101];
      }
      forwarding-class fc-af11 {
        loss-priority low code-points [100];
```

```
        }
        forwarding-class fc-ef1 {
          loss-priority low code-points [011];
        }
        forwarding-class fc-ef {
          loss-priority low code-points [010];
        }
        forwarding-class fc-be1 {
          loss-priority low code-points [001];
        }
        forwarding-class fc-be {
          loss-priority low code-points [000];
        }
      }
    }
    interfaces {
      ge-1/2/3 {
        input-shaping-rate 500m;
        shaping-rate 500m; # Output shaping rate
        unit 0 { # Apply behavior aggregate classifier to an interface.
          classifiers {
            ieee-802.1 ieee-8021p-table;
          }
        }
        unit 1 {
          input-traffic-control-profile s1 shared-instance 1;
          output-traffic-control-profile s2 shared-instance 1;
        }
        unit 2 {
          input-traffic-control-profile s1 shared-instance 1;
          output-traffic-control-profile s2 shared-instance 1;
        }
        unit 3 {
          input-traffic-control-profile s1 shared-instance 1;
          output-traffic-control-profile s2 shared-instance 1;
        }
        unit 4 {
          input-traffic-control-profile s1 shared-instance 2;
          output-traffic-control-profile s2 shared-instance 2;
        }
        unit 5 {
          input-traffic-control-profile s1 shared-instance 2;
          output-traffic-control-profile s2 shared-instance 2;
        }
        unit 6 {
          input-traffic-control-profile s1 shared-instance 2;
          output-traffic-control-profile s2 shared-instance 2;
        }
      }
    }
  }
```

Configuring a Simple Filter

Configure a simple filter that overrides the classification derived from the lookup of the Layer 2 fields.

```
firewall {
```



```
  family inet {
    simple-filter sf-1 {
      term 1 {
        source-address 172.16.0.0/16;
        destination-address 20.16.0.0/16;
        source-port 1024-9071;
      }
      then { # Action with term-1
      forwarding-class fc-be1;
      loss-priority high;
    }
    term 2 {
      source-address 173.16.0.0/16;
      destination-address 21.16.0.0/16;
    }
    then { # Action with term-2
    forwarding-class fc-ef1;
    loss-priority low;
  }
}
interfaces { # Apply the simple filter.
ge-1/2/3 {
  unit 0 {
    family inet {
      simple-filter {
        input sf-1;
      }
    }
  }
}

class-of-service {
  scheduler-maps { # Configure a custom scheduler map.
  map1 {
    forwarding-class fc-be scheduler sch-Q0;
    forwarding-class fc-be1 scheduler sch-Q1;
    forwarding-class fc-ef scheduler sch-Q2;
    forwarding-class fc-ef1 scheduler sch-Q3;
    forwarding-class fc-af11 scheduler sch-Q4;
    forwarding-class fc-af12 scheduler sch-Q5;
    forwarding-class fc-nc1 scheduler sch-Q6;
    forwarding-class fc-nc2 scheduler sch-Q7;
  }
}
schedulers { # Define schedulers.
sch-Q0 {
  transmit-rate percent 25;
  buffer-size percent 25;
  priority low;
  drop-profile-map loss-priority any protocol any drop-profile drop-default;
}
sch-Q1 {
  transmit-rate percent 5;
  buffer-size temporal 2000;
  priority high;
  drop-profile-map loss-priority any protocol any drop-profile drop-ef;
```

```
}
sch-Q2 {
   transmit-rate percent 35;
   buffer-size percent 35;
   priority low;
   drop-profile-map loss-priority any protocol any drop-profile drop-default;
}
sch-Q3 {
   transmit-rate percent 5;
   buffer-size percent 5;
   drop-profile-map loss-priority any protocol any drop-profile drop-default;
}
sch-Q4 {
   transmit-rate percent 5;
   priority high;
   drop-profile-map loss-priority any protocol any drop-profile drop-ef;
}
sch-Q5 {
   transmit-rate percent 10;
   priority high;
   drop-profile-map loss-priority any protocol any drop-profile drop-ef;
}
sch-Q6 {
   transmit-rate remainder;
   priority low;
   drop-profile-map loss-priority any protocol any drop-profile drop-default;
}
sch-Q7 {
   transmit-rate percent 5;
   priority high;
   drop-profile-map loss-priority any protocol any drop-profile drop-default;
}
```

CHAPTER 20

Configuring CoS on 10-port 10-Gigabit Oversubscribed Ethernet PICs

This chapter discusses the following topics:

CoS on 10-port 10-Gigabit Oversubscribed Ethernet PICs Overview

The 10-port 10-Gigabit Oversubscribed Ethernet (OSE) Type 4 PIC supports intelligent handling of oversubscribed traffic in applications, such as data centers and dense-core uplinks. The 10-port 10-Gigabit OSE PIC supports line rate operation for five 10-Gigabit Ethernet ports from each port group or a total WAN bandwidth of 100 Gbps with Packet Forwarding Engine bandwidth of 50 Gbps.

The class-of-service (CoS) configuration for the 10-port 10-Gigabit OSE PICs are supported on Juniper Networks TX Matrix and TX Matrix Plus and Juniper Networks T640 and T1600 Core Routers. The 10-port 10-Gigabit OSE PICs support behavior aggregate (BA) and fixed classification, weighted round-robin scheduling with two queue priorities (low and strict-high), committed and peak information rate shaping on a per-queue basis, and excess information rate configuration for allocation of excess bandwidth.

To configure these features, include the corresponding class-of-service (CoS) statements at the [**edit class-of-service**] hierarchy level. The CoS statements supported on the 10-port 10-Gigabit OSE PICs are shown in Table 102 on page 348.

Table 102: CoS Statements Supported on the 10-port 10-Gigabit OSE PICs

CoS Statements	Supported
buffer-size	No
drop-profile-map	No
excess-priority	No
excess-rate	Yes
priority	Yes
shaping-rate	Yes
transmit-rate	Yes

NOTE: DSCP rewrite is not supported on the 10-port 10-Gigabit Oversubscribed Ethernet PIC.

For information about CoS components that apply generally to all interfaces, see "Hardware Capabilities and Limitations" on page 259. For general information about configuring interfaces, see the *Junos OS Network Interfaces Configuration Guide*.

BA and Fixed Classification on 10-port 10-Gigabit Oversubscribed Ethernet PICs Overview

The 10-port 10-Gigabit OSE PICs support the following behavior aggregate (BA) classifiers:

- DSCP, DSCP IPv6, or IP precedence—IP packet classification (Layer 3 headers)
- MPLS EXP—MPLS packet classification (Layer 2 headers)
- IEEE 802.1p—Packet classification (Layer 2 headers)
- IEEE 802.1ad—Packet classification for IEEE 802.1ad formats (including DEI bit)

NOTE: DSCP rewrite is not supported on the 10-port 10-Gigabit Oversubscribed Ethernet PIC.

Multiple classifiers can be configured to a single logical interface. However, there are some restrictions on which the classifiers can coexist. For example, the DSCP and IP precedence classifiers cannot be configured on the same logical interface. The DSCP and IP precedence classifiers can coexist with the DSCP IPv6 classifier on the same logical interface. An IEEE 802.1 classifier can coexist with other classifiers and is applicable only if a packet does not match any of the configured classifiers. For information about the supported combinations, see “Applying Classifiers to Logical Interfaces” on page 50.

If the classifiers are not defined explicitly, then the default classifiers are applied as follows:

- All MPLS packets are classified using the MPLS (EXP) classifier. If there is no explicit MPLS (EXP) classifier, then the default MPLS (EXP) classifier is applied.
- All IPv4 packets are classified using the IP precedence and DSCP classifiers. If there is no explicit IP precedence and DSCP classifiers, then the default IP precedence classifier is applied.
- All IPv6 packets are classified using DSCP IPv6 classifier. If there is no explicit DSCP IPv6 classifier, then the default DSCP IPv6 classifier is applied.
- If the IEEE 802.1p classifier is configured and a packet does not match any explicitly configured classifier, then the IEEE 802.1p classifier is applied.

The fixed classification matches the traffic on a logical interface level. The following example classifies all traffic on logical unit zero to the queue corresponding to assured forwarding.

```
[edit class-of-service interfaces xe-0/1/2 unit 0]
forwarding-class fc-af11;
```

NOTE: The 10-port 10-Gigabit OSE PICs do not support multifield classification. However, the multifield classification can be done at the Packet Forwarding Engine using the firewall filters, which overrides the classification done at the PIC level. The multifield classification at the Packet Forwarding Engine occurs after the PIC handles the oversubscribed traffic.

Example: Configuring IEEE 802.1p BA Classifier on 10-port 10-Gigabit Oversubscribed Ethernet PICs

To configure an IEEE 802.1p behavior aggregate (BA) classifier on the 10-port 10-Gigabit Oversubscribed Ethernet (OSE) PICs, include the following statements at the **[edit class-of-service]** hierarchy level:

```
[edit class-of-service classifiers]
ieee-802.1 classifier-name {
  forwarding-class fc-nc2 {
    loss-priority low code-points [111];
  }
  forwarding-class fc-nc1 {
    loss-priority low code-points [110];
  }
  forwarding-class fc-af12 {
    loss-priority low code-points [101];
  }
  forwarding-class fc-af11 {
    loss-priority low code-points [100];
```

```
        }
        forwarding-class fc-ef1 {
            loss-priority low code-points [011];
        }
        forwarding-class fc-ef {
            loss-priority low code-points [010];
        }
        forwarding-class fc-be1 {
            loss-priority low code-points [001];
        }
        forwarding-class fc-be {
            loss-priority low code-points [000];
        }
    }
    [edit class-of-service interfaces xe-0/1/2 unit 0]
    classifiers {
        ieee-802.1 classifier-name;
    }
```

NOTE: The 10-port 10-Gigabit OSE PICs do not support queuing at the logical interface level. However, the classifiers can be configured to the individual logical interfaces. The same classifier can be configured to the multiple logical interfaces.

Queuing on 10-port 10-Gigabit Oversubscribed Ethernet PICs Properties

The 10-port 10-Gigabit OSE PICs have the following features to support queuing:

- Committed and peak information rate shaping on a per-queue basis
- Excess information rate configuration for allocation of excess bandwidth
- Ingress queuing based on behavior aggregate (BA) classification
- Egress queuing at the Packet Forwarding Engine and at the PIC level

  The Packet Forwarding Engine egress queues are shared by two physical interfaces in a port group.
- Weighted round-robin (WRR) scheduling with two queue priorities (low and strict-high)
- Two special queues available in ingress, one per physical interface, called *control queues*

  Layer 2 and Layer 3 control protocol packets (OSPF, OSPF3, VRRP, IGMP, RSVP, PIM, BGP, BFD, LDP, ISIS, RIP, RIPV6, LACP, ARP, IPv6 NDP) are mapped to the control queue. In the control queue, these packets are not dropped even if there is oversubscription or congestion on a port group.

  NOTE: The control queue is rate-limited to 2 Mbps per physical interface. The packets in excess of 2 Mbps are dropped and accounted for.

Example: Mapping Forwarding Classes to CoS Queues on 10-port 10-Gigabit Oversubscribed Ethernet PICs

The 10-port 10-Gigabit OSE PICs support eight CoS queues per port in the egress direction. To map forwarding classes to the eight CoS queues in egress, include the following statements at the **[edit class-of-service]** hierarchy level:

```
[edit class-of-service forwarding-classes] {
  class fc-be queue-num 0;
  class fc-be1 queue-num 1;
  class fc-ef queue-num 2;
  class fc-ef1 queue-num 3;
  class fc-af11 queue-num 4;
  class fc-af12 queue-num 5;
  class fc-nc1 queue-num 6;
  class fc-nc2 queue-num 7;
}
```

The 10-port 10-Gigabit OSE PICs support four ingress queues per physical interface. The PICs use restricted-queues configuration to map multiple forwarding classes to the four queues. There are no queues at the logical interface level. In the following example, two forwarding classes are mapped to one queue.

```
[edit class-of-service restricted-queues] {
  forwarding-class fc-be queue-num 0;
  forwarding-class fc-be1 queue-num 0;
  forwarding-class fc-ef queue-num 1;
  forwarding-class fc-ef1 queue-num 1;
  forwarding-class fc-af11 queue-num 2;
  forwarding-class fc-af12 queue-num 2;
  forwarding-class fc-nc1 queue-num 3;
  forwarding-class fc-nc2 queue-num 3;
}
```

Scheduling and Shaping on 10-port 10-Gigabit Oversubscribed Ethernet PICs Overview

The 10-port 10-Gigabit OSE PIC has ten 10-Gigabit Ethernet ports providing 100 Gbps of WAN bandwidth and 50 Gbps of Packet Forwarding Engine bandwidth. On the ingress side of the 10-port 10-Gigabit OSE PIC, two consecutive physical interfaces on the PICs are grouped together into a port group and are serviced by a single scheduler. The port groups are as shown in Table 103 on page 351:

Table 103: Port Groups on 10-port 10-Gigabit OSE PIC

Port Group	Mapped Ports
Group 1	xe-*fpc*/*pic*/0 xe-*fpc*/*pic*/1
Group 2	xe-*fpc*/*pic*/2 xe-*fpc*/*pic*/3

Table 103: Port Groups on 10-port 10-Gigabit OSE PIC *(continued)*

Port Group	Mapped Ports
Group 3	xe-*fpc*/*pic*/4 xe-*fpc*/*pic*/5
Group 4	xe-*fpc*/*pic*/6 xe-*fpc*/*pic*/7
Group 5	xe-*fpc*/*pic*/8 xe-*fpc*/*pic*/9

The two physical interfaces in a port group share 10 Gbps bandwidth towards the Packet Forwarding Engine. A scheduler has eight class-of-service (CoS) queues and two control queues. On the ingress side of the 10-port 10-Gigabit OSE PIC, the eight CoS queues are split four plus four for the two physical interfaces. Thus, the 10-port 10-Gigabit OSE PIC supports four ingress queues and eight egress queues per physical interface.

At the ingress side of the 10-port 10-Gigabit OSE PIC, multiple forwarding classes can be mapped to one queue using the restricted-queue configuration. When creating a scheduler-map for the ingress queues, only one forwarding class should be chosen from the multiple forwarding classes that map to the same queue. Then, the scheduler-map can be specified using the **set class-of-service scheduler-maps** ***map-name*** **forwarding-class** ***class-name*** **scheduler** ***scheduler*** command.

The 10-port 10-Gigabit OSE PICs manage packet buffering internally and no configuration is required.

NOTE: The delay-bandwidth buffering configuration is not supported on the 10-port 10-Gigabit OSE PICs.

Example: Configuring Shaping Overhead on 10-port 10-Gigabit Oversubscribed Ethernet PICs

By default, the 10-port 10-Gigabit OSE PIC uses 20 bytes as the shaping overhead. This includes 8 bytes preamble and 12 bytes interpacket gap (IPG) in shaper operations. To exclude this overhead, it should be configured as –20 bytes. The shaping overhead value can be set between 0 and 31 bytes, as shown in the following example. This range translates to a CLI range of –20 to 11 bytes for the shaping overhead configuration.

```
show chassis
  fpc 6 {
  pic 0 {
    traffic-manager {
    ingress-shaping-overhead –20;
```

```
        egress-shaping-overhead –20;
        }
    }
}
```

NOTE: When the configuration for the overhead bytes on a PIC are changed, the PIC is taken offline and then brought back online. In addition, the configuration in the CLI is on a per-PIC basis, and thus, applies to all the ports on the PIC.

CHAPTER 21

Configuring CoS on Enhanced Queuing DPCs

This topic discusses the following:

Enhanced Queuing DPC Hardware Properties

On a Juniper Networks MX Series Ethernet Services Router with Enhanced Queuing Dense Port Concentrators (DPCs), you can configure schedulers and queues. You can configure 15 VLAN sets per Gigabit Ethernet (1G) port and 255 VLAN sets per 10-Gigabit Ethernet (10G) port. The Enhanced Queuing DPC performs priority propagation from one hierarchy level to another and drop statistics are available on the Enhanced Queuing DPC per color per queue instead or just per queue.

Juniper Networks MX Series Ethernet Services Routers with Enhanced Queuing DPCs have Packet Forwarding Engines that can support up to 515 MB of frame memory, and packets are stored in 512-byte frames. Table 104 on page 355 compares the major properties of the Intelligent Queuing 2 (IQ2) PIC and the Packet Forwarding Engine within the Enhanced Queuing DPC.

Table 104: IQ2 PIC and Enhanced Queuing DPC Compared

Feature	IQ2 PIC	Packet Forwarding Engine Within Enhanced Queuing DPC
Number of usable queues	8,000	16,000
Number of shaped logical interfaces	1,000 with 8 queues each.	2,000 with 8 queues each, or 4,000 with 4 queues each.

Table 104: IQ2 PIC and Enhanced Queuing DPC Compared *(continued)*

Feature	IQ2 PIC	Packet Forwarding Engine Within Enhanced Queuing DPC
Number of hardware priorities	2	4
Priority propagation	No	Yes
Dynamic mapping	No: schedulers/port are fixed.	Yes: schedulers/port are not fixed.
Drop statistics	Per queues	Per queue per color (PLP high, low)

In addition, the Enhanced Queuing DPC features support for hierarchical weighted random early detection (WRED) and enhanced queuing on aggregated Ethernet interfaces with link protection as well.

The Enhanced Queuing DPC 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 Enhanced Queuing DPC supports the following features for scalability:

- 16,000 queues per Packet Forwarding Engine
- 4 Packet Forwarding Engines per DPC
  - 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 Packet Forwarding Engine on a 10-Gigabit Ethernet DPC
- 15 schedulers at the interface set level (Level 2) per 10-port Packet Forwarding Engine on a 1-Gigabit Ethernet DPC
- 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: Including the **transmit-rate** *rate* **exact** statement at the **[edit class-of-service schedulers** *scheduler-name***]** hierarchy level is not supported on Enhanced Queuing DPCs on MX Series routers.

The way that the Enhanced Queuing DPC 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 results in a restart of the DPC. For more information about the **max-queues-per-interface** statement, see the *Junos OS Network Interfaces Configuration Guide*.

The Enhanced Queuing DPC 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 of the schedulers in the group should use the same queue priority mapping. For example, if the queue priorities of one scheduler are high, medium, low, and low, then all members of this group should have the same queue priority.

Mapping of a group at level 3 to level 2 can be done at any time. However, a group at level 3 can only be unmapped from a level 2 scheduler 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 Enhanced Queuing DPC maps level 2 (service VLAN) schedulers in a fixed mode to level 1 (physical interface) schedulers. On 40-port Gigabit Ethernet DPCs, 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 scheduler. A level 1 scheduler uses level schedulers X*16 through X*16+15. So 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.

Enhanced Queuing is supported on aggregated Ethernet (AE) interfaces with two links in link protection mode. However, only one link in the AE bundle can be active at a time. Traffic is shaped independently on the two links, but the member's links do not need to reside in the same Packet Forwarding Engine or the same DPC. Finally, shared schedulers are not supported on the Enhanced Queuing DPC (use hierarchical schedulers to group logical interfaces).

Configuring Rate Limits on Enhanced Queuing DPCs

You can rate-limit the strict-high and high queues on the Enhanced Queuing DPC. Without this limiting, traffic in higher priority queues can block the transmission of lower priority packets. Unless limited, higher priority traffic is always sent before lower priority traffic, causing the lower priority queues to "starve" and cause timeouts and unnecessarily resent packets.

On the Enhanced Queuing DPC you can rate-limit queues before the packets are queued for output. All packets exceeding the configured rate limit are dropped, so care is required when establishing this limit. This model is also supported on IQ2 PICs. For more information about configuring CoS on IQ2 PICs, see "CoS on Enhanced IQ2 PICs Overview" on page 325.

To rate-limit queues, include the **transmit-rate** statement with the **rate-limit** option at the **[edit class-of-service schedulers *scheduler-name*]** hierarchy level:

```
[edit class-of-service schedulers scheduler-name]
transmit-rate rate rate-limit;
```

This example limits the transmit rate of a strict-high expedited-forwarding queue to 1 Mbps. The scheduler and scheduler map are defined, and then applied to the traffic at the **[edit interfaces]** and **[edit class-of-service]** hierarchy levels:

```
[edit class-of-service]
schedulers {
  scheduler-1 {
    transmit-rate 1m rate-limit; # This establishes the limit
    priority strict-high;
  }
}
scheduler-maps {
  scheduler-map-1 {
    forwarding-class expedited-forwarding scheduler scheduler-1;
  }
}

[edit interfaces]
s0-2/2/0 {
  per-unit-scheduler;
  encapsulation frame-relay;
  unit 0 {
    dlci 1;
  }
}

[edit class-of-service]
interfaces {
  so-2/2/0 {
    unit 0 {
      scheduler-map scheduler-map-1;
      shaping-rate 2m;
    }
```

```
  }
}
```

You can issue the following operational mode commands to verify your configuration (the first shows the rate limit in effect):

- **show class-of-service scheduler-map** ***scheduler-map-name***
- **show class-of-service interface** ***interface-name***

Configuring Simple Filters on Enhanced Queuing DPCs

You can configure and apply a simple filter to perform multifield classification on the ingress interfaces of an MX Series router with Enhanced Queuing DPCs. These simple filters can be used to override default CoS classification parameters such as forwarding class or loss priority. Simple filters, in contrast to other firewall filters, only support a subset of the full firewall filter syntax.

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

```
[edit firewall family inet]
simple-filter filter-name {
  term term-name {
    from {
      ... match-conditions...
    }
    then {
      forwarding-class class-name;
      loss-priority priority;
    }
  }
}
```

For more information about configuring simple filters, see "Overview of Simple Filters" on page 79.

The following example configures a simple filter to detect ingress packets from various source addresses (**10.1.1.1/32**, **10.10.10.10/32**, and **10.4.0.0/8**), destination addresses (**10.6.6.6/32**), protocols (**tcp**), and source ports (**400-500**, **http**). The filter then assigns various forwarding classes and loss priorities to the filtered traffic. Finally, the filter is applied to the input side of an Enhanced Queuing DPC interface (**ge-2/3/3**).

```
[edit]
firewall {
  family inet {
    simple-filter sf-for-eq-dpc {
      term 1 {
        from {
          source-address 10.1.1.1/32;
          protocol tcp;
        }
        then loss-priority low;
      }
      term 2 {
```

```
                from {
                    source-address 10.4.0.0/8;
                    source-port http;
                }
                then loss-priority high;
            }
            term 3 {
                from {
                    destination-address 10.6.6.6/32;
                    source-port 400-500;
                }
                then {
                    loss-priority low;
                    forwarding-class best-effort;
                }
            }
            term 4 {
                from {
                    forwarding-class expedited-forwarding;
                    source-address 10.10.10.10/32;
                }
                then loss-priority low;
            }
            term 5 {
                from {
                    source-address 10.10.10.10/32;
                }
                then loss-priority low;
            }
        }
    }
}
interfaces { # Apply the simple filter above to the input side of the interface.
ge-2/3/3 {
    unit 0 {
        family inet {
            simple-filter {
                input sf-for-eq-dpc;
            }
        }
    }
}
```

Configuring WRED on Enhanced Queuing DPCs

Shaping to drop out-of-profile traffic is done on the Enhanced Queuing DPC at all levels but 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 Enhanced Queuing DPC is similar to the IQ2 PIC, but involves only two levels, not 64. 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.

To configure WRED, include the **drop-profiles** statement at the **[edit class-of-service]** hierarchy level:

```
[edit class-of-service]
drop-profiles {
  profile-name {
    fill-level percentage drop-probability percentage;
  }
}
```

The following example is an Enhanced Queuing DPC drop profile for expedited forwarding traffic:

```
[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 that only two fill levels can be specified for the Enhanced Queuing DPC. 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 a packet delay buffers, where one buffer is equal to 512 bytes. For example, at 10 Mbps, the Enhanced Queuing DPC allocates 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 only configured at the queue, physical interface, and PIC level). Hierarchical RED works with WRED as follows:

- If any level accepts the packet (the queue depth is less than the minimum buffer level), then 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 which might otherwise have been dropped. In other words, the logical interface accepts packets if the physical interface is not congested.

Related Documentation

- Shaping Granularity Values for Enhanced Queuing Hardware on page 330
- For more information about configuring RED drop profiles, see RED Drop Profiles Overview on page 231.

Configuring MDRR on Enhanced Queuing DPCs

The guaranteed rate (CIR) at the interface set level is implemented using modified deficit round-robin (MDRR). The Enhanced Queuing DPC 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 under the shaping rate (PIR). The Enhanced Queuing DPC 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 4096 logical interfaces.

The 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. The Junos OS provides three priorities when there is no guaranteed rate configured on any logical interface.

The relationship between Junos OS priorities and the Enhanced Queuing DPC hardware priorities below and above the guaranteed rate (CIR) is shown in Table 105 on page 362.

Table 105: Junos Priorities Mapped to Enhanced Queuing DPC Hardware Priorities

Junos OS Priority	Enhanced Queuing DPC Hardware Priority Below Guaranteed Rate	Enhanced Queuing DPC 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

To configure MDRR, configure a scheduler at the **[edit class-of-service schedulers]** hierarchy level:

```
[edit class-of-service schedulers]
scheduler-name {
  buffer-size (seconds | percent percentage | remainder | temporal microseconds);
  priority priority-level;
  transmit-rate (percent percentage | rate | remainder) <exact | rate-limit>;
}
```

The following example creates two schedulers for MDRR:

```
[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 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 Enhanced Queuing DPC 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-propotional-rate / 255

To configure the way that the Enhanced Queuing DPC should handle excess bandwidth, configure the **excess-bandwith-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.

This 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 use 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.

Configuring Excess Bandwidth Sharing

When using the Enhanced Queuing DPC on an MX Series router, there are circumstances when you should configure excess bandwidth sharing and minimum logical interface shaping. This section details some of the guidelines for configuring excess bandwidth sharing.

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 106 on page 364.

Table 106: 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 weights on the logical interface are 257 and the WFQ accuracy is the same.

Selecting Excess Bandwidth Sharing Proportional Rates

A good excess bandwidth sharing proportional rate to configure is to 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 weighed round-robin (WRR) rate. This 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 107 on page 365.

Table 107: 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 weight of each = 255).

Mapping Calculated Weights 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 108 on page 365.

Table 108: 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%
57–62	6	194–244 (interval of 10)	2.71%
63–63	1	255–255 (interval of 11)	2.05%

From the table, as an example, 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 18–42).

Allocating Weight 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. In order 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, consider a logical interface configuration with five units, as shown in Table 109 on page 366.

Table 109: 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
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:

- Select the excess bandwidth sharing proportional rate to be the maximum CIR among all the logical interfaces: 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 although 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.

Sharing 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 110 on page 367.

NOTE: On the MX960 router, bandwidth sharing across high priority and strict-high priority schedulers configured on logical interfaces might not be as expected. This is a hardware limitation.

Table 110: Sharing 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

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.
2. 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 111 on page 367.

Table 111: First Example of Bandwidth Sharing

Logical Interface (Unit)	Calculation	Bandwidth
Unit 1	10 / (10+64+128+10) x 60 Mbps	2.83 Mbps
Unit 2	64 / (10+64+128+10) x 60 Mbps	18.11 Mbps
Unit 3	128 / (10+64+128+10) x 60 Mbps	36.22 Mbps
Unit 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 112 on page 368.

Table 112: Second Example of Bandwidth Sharing

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

Finally, Table 113 on page 368 shows the resulting allocation of bandwidth among the logical interfaces when the port is configured with a 100-Mbps shaping rate.

Table 113: Final Example of Bandwidth Sharing

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

Configuring Ingress Hierarchical CoS on Enhanced Queuing DPCs

You can configure ingress CoS parameters, including hierarchical schedulers, on MX Series routers with Enhanced Queuing DPCs. In general, the supported configuration statements apply to per-unit schedulers or to hierarchical schedulers.

To configure ingress CoS for per-unit schedulers, include the following statements at the **[edit class-of-service interfaces *interface-name*]** hierarchy level:

```
[edit class-of-service interfaces interface-name]
input-excess-bandwith-share (proportional value | equal);
input-scheduler-map map-name;
input-shaping-rate rate;
input-traffic-control-profile profiler-name shared-instance instance-name;
unit logical-unit-number;
  input-scheduler-map map-name;
  input-shaping-rate (percent percentage | rate);
  input-traffic-control-profile profile-name shared-instance instance-name;
}
```

To configure ingress CoS for hierarchical schedulers, include the **interface-set *interface-set-name*** statement at the **[edit class-of-service interfaces]** hierarchy level:

```
[edit class-of-service interfaces]
interface-set interface-set-name {
  input-excess-bandwith-share (proportional value | equal);
  input-traffic-control-profile profiler-name shared-instance instance-name;
  input-traffic-control-profile-remaining profile-name;
```

```
    interface interface-name {
        input-excess-bandwith-share (proportional value | equal);
        input-traffic-control-profile profiler-name shared-instance instance-name;
        input-traffic-control-profile-remaining profile-name;
        unit logical-unit-number;
            input-traffic-control-profile profiler-name shared-instance instance-name;
        }
    }
}
```

By default, ingress CoS features are disabled on the Enhanced Queuing DPC.

You must configure the **traffic-manager** statement with **ingress-and-egress** mode to enable ingress CoS on the ED DPC:

```
[edit chassis fpc slot-number pic pic-number]
traffic-manager mode ingress-and-egress;
```

Configured CoS features on the ingress are independent of CoS features on the egress except that:

- If you configure a per-unit or hierarchical scheduler at the **[edit class-of-service interfaces]** hierarchy level, the schedulers apply in both the ingress and egress directions.
- You cannot configure the same logical interface on an ingress and an egress interface set. A logical interface can only belong to one interface set.
- The DPC's frame buffer of 512 MB is shared between ingress and egress configurations.

The following behavior aggregate (BA) classification tables are supported on the ingress side of the Enhanced Queuing DPC:

- inet-precedence
- DSCP
- exp (MPLS)
- DSCP for IPv6
- IEEE 802.1p

CHAPTER 22

Configuring CoS on Trio DPC and MPC/MIC Interfaces

This topic discusses the following:

CoS on Trio DPC and MPC/MIC Features Overview

This topic covers aspects of Class of Service (CoS) configuration for the Trio Distributed Port Concentrator (DPC), Modular Port Concentrator (MPC), and Modular Interface Card (MIC), with the emphasis on differences between the Trio interface family and other families of interface types. The CoS characteristics of the Trio queuing model are optimized compared to the CoS characteristics of the standard queuing model. The Trio queuing model also supports four levels of hierarchical scheduling, with scheduling node levels corresponding to the physical interface to the queue itself. For more information on hierarchical schedulers in general, see "Configuring Hierarchical Schedulers for CoS" on page 207.

Key aspects of the Trio queuing model are:

- The model separates the guaranteed bandwidth concept from the weight of a interface node. Although often used interchangeably, guaranteed bandwidth is the bandwidth a node can use when it wants to, independently of what is happening at the other nodes of the scheduling hierarchy. On the other hand, the weight of a node is a quantity that determines how the excess bandwidth is used. The weight is important when the siblings of a node (that is, other nodes at the same level) use less than the sum of the their guaranteed bandwidths. In some applications, such as constant bit rate voice where there is little concern about excess bandwidth, the guaranteed bandwidth dominates the node; whereas in others, such as bursty data, where a well-defined bandwidth is not always possible, the concept of weight dominates the node.
- The model allows multiple levels of priority to be combined with guaranteed bandwidth in a general and useful way. There is a set of priorities for guaranteed levels and a set of priorities for excess levels that are at a lower absolute level. For each guaranteed level, there is only one excess level paired with it. There are thee guaranteed priorities and two excess priorities. You can configure one guaranteed priority and one excess priority. For example, you can configure a queue for guaranteed low (GL) as the guaranteed priority and configure excess high (EH) as the excess priority.
- However, for an excess level, there can be any number of guaranteed priority levels, including none. Nodes maintain their guaranteed priority level (for example, guaranteed high, GH) as long as they do not exceed their guaranteed bandwidth. If the queue bandwidth exceeds the guaranteed rate, then the priority drops to the excess priority (for example, excess high, EH). Because excess level priorities are lower than their guaranteed counterparts, the bandwidth guarantees for each of the other levels can be maintained.

There are a number of other general points about the Trio DPC and MPC/MIC interfaces that should be kept in mind:

- Input queuing is not supported on the Trio DPC and MPC/MIC interfaces.
- On Trio MPCs, you can configure up to 32 DCSP or Internet or EXP rewrite rules, and 32 IEEE rewrite rules. However, if you configure all 32 allowed rewrite rules, the class-of-service process intermittently fails and generates syslog entries.

- The Trio DPC and MPC/MIC interfaces do not support the **q-pic-large-buffer** statement at the **[edit chassis fpc *fpc-number* pic *pic-number*]** hierarchy level. By default, 500 ms worth of buffer is supported when the delay buffer rate is less than 1 Gbps. By default, 100 ms worth of buffer is supported when the delay buffer rate is 1 Gbps or more. The maximum supported value for the delay buffer is 256 MB and the minimum value is 4 KB. However, due to the limited number of drop profiles supported and the large range of supported speeds, there can be differences between the user-configured value and the observed hardware value. The enhanced queuing (EQ) Trio DPC and MPC/MIC interfaces support up to 255 drop profiles, and up to 128 tail-drop priorities for guaranteed low (GL) priorities and 64 each for guaranteed high and medium priorities.
- All tunnel interfaces have 100-ms buffers. The **huge-buffer-temporal** statement is not supported.
- The Trio DPC and MPC/MIC interfaces take all Layer 1 and Layer 2 overhead bytes into account for all levels of the hierarchy, including preamble, interpacket gaps, frame check sequence, and cyclical redundancy check. Queue statistics also take these overheads into account when displaying byte statistics.
- The Trio DPC and MPC/MIC interfaces do not support the **excess-bandwidth-sharing** statement. You can use the **excess-rate** statement in scheduler maps and traffic control profiles instead.

The Trio MPC/MIC interfaces have a certain granularity in the application of configured shaping and delay buffer parameters. In other words, the values used are not necessarily precisely the values configured. Nevertheless, the derived values are as close tot he configured values as allowed. For the Trio MPC, the shaping rate granularity is 250 kbps for coarse-grained queuing on the basic hardware and 24 kbps for fine-grained queuing on the enhanced queuing devices.

For delay buffers, the coarse-grained devices support 100 ms of transit rate by default, which can be changed by configuring an explicit buffer size. For fine-grained queuing on enhanced queuing devices, 500 ms of transmit rate is available by default, which can be changed by configuring an explicit buffer size. When this value is changed, there are 256 points available and the closest point is chosen. High-priority and medium-priority queues use 64 points, and the low-priority queues uses 128.

Another useful feature is the ability to control how much overhead to count with the **traffic-manager** statement and options. By default, overhead of 24 bytes (20 bytes for the header, plus 4 bytes of cyclical redundancy check [CRC]), is added to egress shaping statistics. You can configure the system to adjust the number of bytes to add to a packet to determine shaped session packet length by adding more bytes (up to 124) of overhead. You can also subtract bytes for egress shaping overhead (up to minus 63 bytes).

This example adds 12 more bytes of overhead to the egress shaping statistics:

```
[edit chassis fpc 0 pic 0]
traffic-manager egress-shaping-overhead 12;
```

In contrast to the Intelligent Queuing Enhanced (IQE) and Intelligent Queuing 2 Enhanced (IQ2E) PICs, the Trio DPC and MPC/MIC interfaces set the guaranteed rate to zero in oversubscribed PIR mode for the per-unit scheduler. Also, the configured rate is scaled down to fit the oversubscribed value. For example, if there are two logical interface units with a shaping rate of 1 Gbps each on a 1-Gbps port (which is, therefore, oversubscribed 2 to 1), then the guaranteed rate on each unit is scaled down to 500 Mbps (scaled down by 2).

With hierarchical schedulers in oversubscribed PIR mode, the guaranteed rate for every logical interface unit is set to zero. This means that the queue transmit rates are always oversubscribed.

Because in oversubscribed PIR mode the queue transmit rates are always oversubscribed, the following are true:

- If the queue transmit rate is set as a percentage, then the guaranteed rate of the queue is set to zero; but the excess rate (weight) of the queue is set correctly.
- If the queue transmit rate is set as an absolute value and if the queue has guaranteed high or medium priority, then traffic up to the queue's transmit rate is sent at that priority level. However, for guaranteed low traffic, that traffic is demoted to the excess low region. This means that best-effort traffic well within the queue's transmit rate gets a lower priority than out-of-profile excess high traffic. This differs from the IQE and IQ2E PICs.

Several other aspects of the Trio DPC and MPC/MIC interfaces should be kept in mind when configuring CoS:

- When the Trio DPC and MPC/MIC interface's delay buffers are oversubscribed by configuration (that is, the user has configured more delay-buffer memory than the system can support), then the configured weighted random early detection (WRED) profiles are implicitly scaled down to drop packets more aggressively from the relatively full queues. This creates buffer space for packets in the relatively empty queues and provides a sense of fairness among the delay buffers. There is no configuration needed for this feature.
- When load balancing on the Trio MPC Type 1 3D EQ MIC interfaces, you should configure odd- and even-numbered interfaces in the form ***interface-fpc/odd* | *even/***ports. For example, if one link is **xe-1/0/0**, the other should be **xe-1/1/0**. If you do not configure odd and even load balancing, the system RED-drops packets when sending at line rate. This limitation does not apply to the Trio MPC Type 2 3D Enhanced Queuing MIC interfaces.

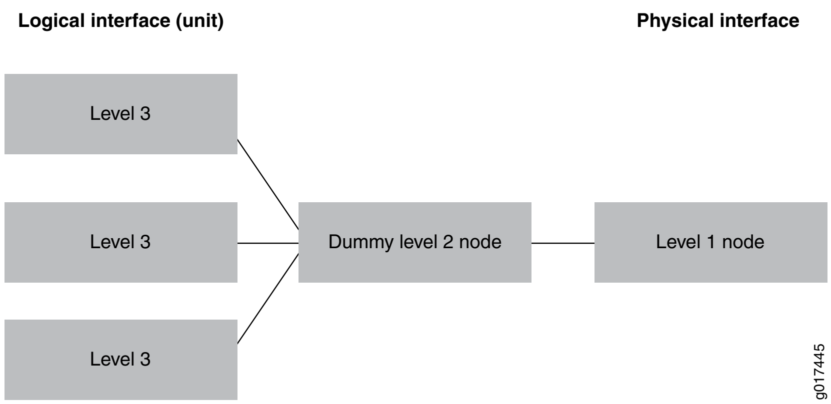
Scheduler Node Scaling on the Trio MPC/MIC Interfaces Overview

The Trio MPC/MIC interface hardware (but not the Trio DPC) supports multiple levels of scheduler nodes. In per-unit-scheduling mode, each logical interface (unit) can have 4 or 8 queues and has a dedicated level 3 scheduler node. Scheduler nodes can be a one of four levels: the queue itself (level 4), the logical interface or unit (level 3), the interface set or virtual LAN (VLAN) collection (level 2), or the physical interface or port (level 1). For more information about hierarchical scheduling levels, see "Configuring Hierarchical Schedulers for CoS" on page 207.

The Trio MPC/MIC interface hardware supports enhanced queuing in both per-unit scheduling and hierarchical scheduler modes. The way that the scheduler hierarchy is built depends on the scheduler mode configured.

In per-unit scheduling mode, each logical interface unit has its own dedicated level 3 node and all logical interface units share a common level 2 node (one per port). This scheduling mode is shown in Figure 22 on page 375.

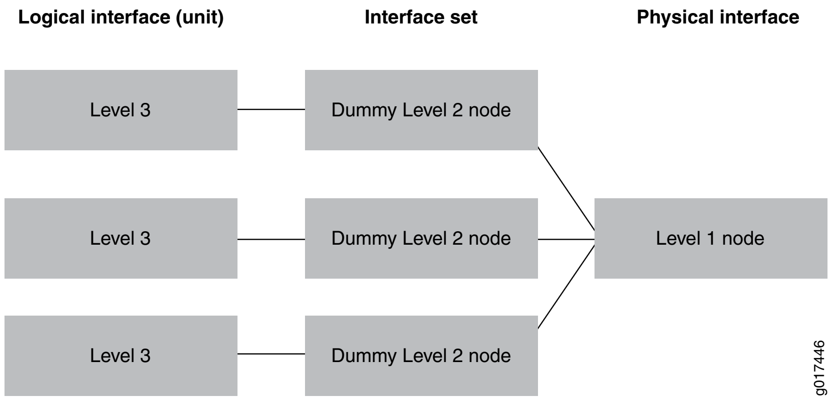
Figure 22: Trio DPC and MPC/MIC interface Per-unit Scheduler Node Scaling



In this case, in per-unit scheduling mode, the level 2 node is a dummy node.

The case with hierarchical scheduling mode, for a similar configuration when there are no interface sets configured and only the logical interfaces have traffic control profiles is shown in Figure 23 on page 376.

Figure 23: Trio DPC and MPC/MIC interface Hierarchical Scheduling Node Scaling



When an interface set has a CoS scheduling policy but none of its child logical interfaces has a CoS scheduling policy, then the interface set is considered to be a leaf node and has one level 2 and one level 3 node.

In per-unit scheduling, the logical interfaces share a common level 2 node (one per port). In hierarchical-scheduling mode, each logical interface has its own level 2 node. So scaling is limited by the number of level 2 nodes. For hierarchical schedulers, in order to better control system resources in hierarchical-scheduling mode, you can limit the number of hierarchical levels in the scheduling hierarchy to two. In this case, all logical interfaces and interface sets with CoS scheduling policy share a single (dummy) level 2 node, so the maximum number of logical interfaces with CoS scheduling policies is increased (the interface sets must be at level 3). To configure scheduler node scaling, include the **hierarchical-schedulers** statement with the **maximum-hierarchy-levels** option at the **[edit interfaces xe-*fpc/pic/port*]** hierarchy level. The only supported value is 2.

```
[edit interfaces]
xe-2/0/0 {
  hierarchical-schedulers {
    maximum-hierarchy-levels 2;
  }
}
```

NOTE: Level 3 interface sets are supported by the **maximum-hierarchy-levels** option, but level 2 interface sets are not supported. If you configure level 2 interface sets with the **maximum-hierarchy-levels** option, you generate Packet Forwarding Engine errors.

Dedicated Queue Scaling for CoS Configurations on Trio MPC/MIC Interfaces Overview

The 30-Gigabit Ethernet Queuing and 60-Gigabit Ethernet Queuing and Enhanced Queuing Ethernet Modular Port Concentrator (MPC) modules provide a set of dedicated queues for subscriber interfaces configured with hierarchical scheduling or per-unit scheduling.

The dedicated queues offered on these modules enable service providers to reduce costs through different scaling configurations. For example, the 60-Gigabit Ethernet Enhanced Queuing MPC module enables service providers to reduce the cost per subscriber by allowing many subscriber interfaces to be created with four or eight queues. Alternatively, the 30-Gigabit Ethernet and 60-Gigabit Ethernet Queuing MPC modules enable service providers to reduce hardware costs, but allow fewer subscriber interfaces to be created with four or eight queues.

This topic describes the overall queue, scheduler node, and logical interface scaling for subscriber interfaces created on these Trio MPC/MIC module combinations.

Queue Scaling for Trio MPC/MIC Interfaces

Table 114 on page 377 lists the number of dedicated queues and number of subscribers supported per Trio MPC module.

Table 114: Dedicated Queues for Trio MPC/MIC Interfaces

MPC	Dedicated Egress Queues	Supported Subscriber Interfaces	Logical Interfaces with 4 Queues	Logical Interfaces with 8 Queues
30-Gigabit Ethernet Queuing MPC	64,000	16,000	16,000 (8000 per PIC)	8000 (4000 per PIC)
60-Gigabit Ethernet Queuing MPC	128,000	32,000	32,000 (8000 per PIC)	16,000 (4000 per PIC)
60-Gigabit Ethernet Enhanced Queuing MPC	512,000	64,000	64,000 (16,000 per PIC)	64,000 (16,000 per PIC)

Each interface-set uses 8 queues from total available egress queues.

Management of Dedicated and Remaining Queues

An SNMP trap is generated to notify you when the number of available dedicated queues on the module drops below 10 percent.

When the maximum number of dedicated queues on the Trio MPC modules is reached, a system log message, **COSD_OUT_OF_DEDICATED_QUEUES**, is generated. The system does not provide subsequent subscriber interfaces with a dedicated set of queues. For per-unit scheduling configurations, there are no configurable queues remaining on the module.

For hierarchical scheduling configurations, remaining queues are available when the number of dedicated queues is reached on the module. Traffic from these logical interfaces are considered unclassified and attached to a common set of queues that are shared by all subsequent logical interfaces. These common queues are the default port queues that are created for every port. You can configure a traffic control profile and attach that to the interface to provide CoS parameters for the remaining queues.

For example, when the 30-Gigabit Ethernet Queuing MPC is configured with 32,000 subscriber interfaces with four queues per subscriber, the module can support 16,000 subscribers with a dedicated set of queues. You can provide CoS shaping and scheduling parameters to the remaining queues for those subscriber interfaces by attaching a special traffic-control profile to the interface.

These subscriber interfaces remain with this traffic control profile, even if dedicated queues become available.

Related Documentation

- For information about managing dedicated queues in a static CoS configuration, see Managing Dedicated and Remaining Queues for Static CoS Configurations on Trio MPC/MIC Interfaces on page 378
- For information about managing dedicated queues in a dynamic subscriber access configuration, see Managing Dedicated and Remaining Queues for Dynamic CoS Configurations on Trio MPC/MIC Interfaces
- Scheduler Node Scaling on the Trio MPC/MIC Overview on page 375
- COSD System Log Messages

Managing Dedicated and Remaining Queues for Static CoS Configurations on Trio MPC/MIC Interfaces

This topic describes how to manage dedicated and remaining queues for static subscriber interfaces configured at the **[edit class-of-service]** hierarchy.

- Configuring the Maximum Number of Queues for Trio MPC/MIC Interfaces on page 378
- Configuring Remaining Common Queues on Trio MPC/MIC Interfaces on page 379

Configuring the Maximum Number of Queues for Trio MPC/MIC Interfaces

30-Gigabit Ethernet Queuing Trio MPC modules and 40-Gigabit Ethernet Queuing and Enhanced Queuing Trio MPC modules support a dedicated number of queues when configured for hierarchical scheduling and per-unit scheduling configurations.

To scale the number of subscriber interfaces per queue, you can modify the number of queues supported on the Trio MIC.

To configure the number of queues:

1. Specify that you want to configure the MIC.

   user@host# **edit chassis fpc *slot-number* pic *pic-number***

2. Configure the number of queues.

   [edit chassis fpc *slot-number* pic *pic-number*]
   user@host# **set max-queues-per-interface (8 | 4)**

Configuring Remaining Common Queues on Trio MPC/MIC Interfaces

30-Gigabit Ethernet Queuing Trio MPC modules and 40-Gigabit Ethernet Queuing and Enhanced Queuing Trio MPC modules support a dedicated set of queues when configured with hierarchical scheduling.

When the number of dedicated queues is reached on the module, there can be queues remaining. Traffic from these logical interfaces are considered unclassified and attached to a common set of queues that are shared by all subsequent logical interfaces.

You can configure traffic shaping and scheduling resources for the remaining queues by attaching a special traffic-control-profile to the interface. This feature enables you to provide the same shaping and scheduling to remaining queues as the dedicated queues.

To configure the remaining queues on a Trio MPC/MIC interface:

1. Configure CoS parameters in a traffic-control profile.

   [edit class-of-service]
   user@host# **edit traffic-control-profiles** ***profile-name***

2. Enable hierarchical scheduling for the interface.

   [edit **interfaces** *interface-name*]
   user@host# **set hierarchical-scheduler**

3. Attach the traffic control profiles for the dedicated and remaining queues to the port on which you enabled hierarchical scheduling.

   To provide the same shaping and scheduling parameters to dedicated and remaining queues, reference the same traffic-control profile.

   a. Attach the traffic-control profile for the dedicated queues on the interface.

      [edit class-of-service **interfaces** *interface-name*]
      user@host# **set output-traffic-control-profile** ***profile-name***

   b. Attach the traffic-control profile for the remaining queues on the interface.

      [edit class-of-service **interfaces** *interface-name*]
      user@host# **set output-traffic-control-profile-remaining** ***profile-name***

Related Documentation

- Dedicated Queue Scaling for CoS Configurations on Trio MPC/MIC Interfaces Overview on page 376
- Verifying the Number of Dedicated Queues Configured on Trio MPC/MIC Interfaces on page 379
- Configuring Hierarchical Schedulers for CoS on page 207
- Configuring Interface Sets on page 208

Verifying the Number of Dedicated Queues Configured on Trio MPC/MIC Interfaces

Purpose Display the number of dedicated queue resources that are configured for the logical interfaces on a port.

Action

```
user@host#show class-of-service interface ge-1/1/0
Physical interface: ge-1/1/0, Index: 166
Queues supported: 4, Queues in use: 4
Total non-default queues created: 4
  Scheduler map: <default>, Index: 2
  Chassis scheduler map: <default-chassis>, Index: 4

  Logical interface: ge-1/1/0.100, Index: 72, Dedicated Queues: no
    Shaping rate: 32000
    Object             Name                   Type                 Index
    Scheduler-map      <remaining>                                 0
    Classifier         ipprec-compatibility   ip                   13

  Logical interface: ge-1/1/0.101, Index: 73, Dedicated Queues: no
    Shaping rate: 32000
    Object             Name                   Type                 Index
    Scheduler-map      <remaining>                                 0
    Classifier         ipprec-compatibility   ip                   13

  Logical interface: ge-1/1/0.102, Index: 74, Dedicated Queues: yes
    Shaping rate: 32000
    Object                  Name               Type                Index
    Traffic-control-profile <control_tc_prof>  Output              45866
```

Related Documentation

- Managing Dedicated and Remaining Queues for Static CoS Configurations on Trio MPC/MIC Interfaces on page 378
- Managing Dedicated and Remaining Queues for Dynamic CoS Configurations on Trio MPC/MIC Interfaces

Excess Bandwidth Distribution on the Trio DPC and MPC/MIC interfaces Overview

Service providers often used tiered services that must provide bandwidth for excess traffic as traffic patterns vary. By default, excess bandwidth between a configured guaranteed rate and shaping rate is shared equally among all queues, which might not be optimal for all subscribers to a service.

You must still apply the profile to the interface

The following statements establish a traffic control profile with a shaping rate of 80 Mbps and an excess rate of 100 percent.

```
[edit class-of-service traffic-control-profiles]
tcp-example-excess {
  shaping-rate 80m;
  excess-rate percent 100;
}
```

You can control the distribution of this excess bandwidth for a user with the **excess-rate** and **shaping-rate** statements. To configure the excess rate and shaping rate for a traffic control profile, include the **excess-rate** and **shaping-rate** statements at the [**edit class-of-service traffic-control-profiles *tcp-name***] hierarchy level and then apply the traffic control profile at the [**edit class-of-service interfaces**] hierarchy level.

You can also control the distribution of this excess bandwidth for a queue at the scheduler level. To configure the excess rate for a queue, include the **excess-rate** and (optionally)

excess-priority statements at the [**edit class-of-service scheduler *scheduler-name***] hierarchy level.

You must still configure a scheduler map

The following statements establish a scheduler with an excess rate of 5 percent and a low priority for excess traffic.

```
[edit class-of-service scheduler]
example-scheduler {
  excess-priority low;
  excess-rate percent 5;
}
```

For queues, you cannot configure the excess rate or excess priority in these two cases:

- When the **transmit-rate exact** statement is configured. In this case, the shaping rate is equal to the transmit rate and the queue does not operate in the excess region.
- When the scheduling priority is configured as **strict-high**. In this case, the queue gets all available bandwidth and never operates in the excess region.

Also, for interface sets and logical interface units, the excess rate configuration will not be allowed in PIR mode. This is because the excess rate distributes excess bandwidth once the scheduling nodes reach their guaranteed rate. However, in PIR mode, none of the scheduling nodes have guaranteed rates configured (so allowing excess rate configuration makes no sense).

For more information on hierarchical scheduling and operational modes, see “Configuring Hierarchical Schedulers for CoS” on page 207.

Per-Priority Shaping on the Trio MPC/MIC Interfaces Overview

Per-priority shaping enables you to configure a separate shaping rate for each of the five priority levels supported by the Trio MPC/MIC interfaces family. The main use of per-priority shaping rates is to ensure that higher priority services such as voice and video do not starve lower priority services such as data.

There are five scheduler priorities supported by the Trio family:

- Guaranteed high (GH)
- Guaranteed medium (GM)
- Guaranteed low (GL)
- Excess high (EH)
- Excess low (EL)

The five scheduler priorities support a shaping rate for each priority supported by the Trio family:

- Shaping rate priority high (GH)
- Shaping rate priority medium (GM)
- Shaping rate priority low (GL)

- Shaping rate excess high (EH)
- Shaping rate excess low (EL)

If each service is represented by a forwarding class queued at a separate priority, then assigning a per-priority shaping rate to higher priority services accomplishes the goal of preventing the starvation of lower priority services.

To configure per-priority shaping rates, include the **shaping-rate-excess-high** ***rate*** **<burst-size** ***burst*****>**, **shaping-rate-excess-low** ***rate*** **<burst-size** ***burst*****>**, **shaping-rate-priority-high** ***rate*** **<burst-size** ***burst*****>**, **shaping-rate-priority-low** ***rate*** **<burst-size** ***burst*****>**, or **shaping-rate-priority-medium** ***rate*** **<burst-size** ***burst*****>** at the **[edit class-of-service traffic-control-profiles** ***tcp-name*****]** hierarchy level and apply the traffic control profile at the **[edit interfaces]** hierarchy level. You can specify the rate in absolute values, or by using **k** (kilo-), **m** (mega-) or **g** (giga-) units.

You can include one or more of the per-priority shaping statements in a traffic control profile:

```
[edit class-of-service]
traffic-control-profiles {
  tcp-ge-port {
    shaping-rate-excess-high rate <burst-size bytes>;
    shaping-rate-excess-low rate <burst-size bytes>;
    shaping-rate-priority-high rate <burst-size bytes>;
    shaping-rate-priority-low rate <burst-size bytes>;
    shaping-rate-priority-medium rate <burst-size bytes>;
  }
}
```

NOTE: We do not recommend configuring the burst size because this can have far-reaching and unintended effects.

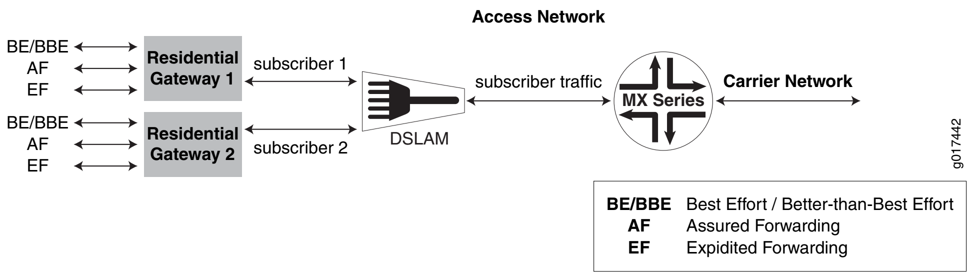
There are several important points about per-priority shaping rates:

- Per-priority shaping rates are only supported on Trio MPC/MIC interface family interfaces (but not the Trio DPC interfaces).
- Per-priority shaping is only available for level 1 and level 2 scheduler nodes. (For more information on hierarchical schedulers, see "Configuring Hierarchical Schedulers for CoS" on page 207.
- Per-priority shaping rates are supported when level 1 or level 2 scheduler nodes have static or dynamic interfaces above them. There is no support for dynamic level 2 interface sets.
- Per-priority shaping rates are supported on aggregated Ethernet (AE) interfaces.
- Per-priority shaping rates are only supported in traffic control profiles.

Per-priority shaping rates can be helpful when the MX Series Ethernet Services Router is in a position between subscriber traffic on an access network and the carrier network, playing the role of a broadband services router. In that case, the MX Series router provides quality-of-service parameters on the subscriber access network so that each subscriber receives a minimum bandwidth (determined by the guaranteed rate) and a maximum bandwidth (determined by the shaping rate). This allows the devices closer to the carrier network to operate more efficiently and more simply and reduces operational network expenses because it allows more centralized network management.

One architecture for using per-priority shaping on the MX Series router is shown in Figure 24 on page 383. In the figure, subscribers use residential gateways with various traffic classes to support voice, video, and data services. The MX Series router sends this traffic from the carrier network to the digital subscriber line access multiplexer (DSLAM) and from the DSLAM on to the residential gateway devices.

Figure 24: Architecture for Trio DPC and MPC/MIC Interface Per-Priority Shaping

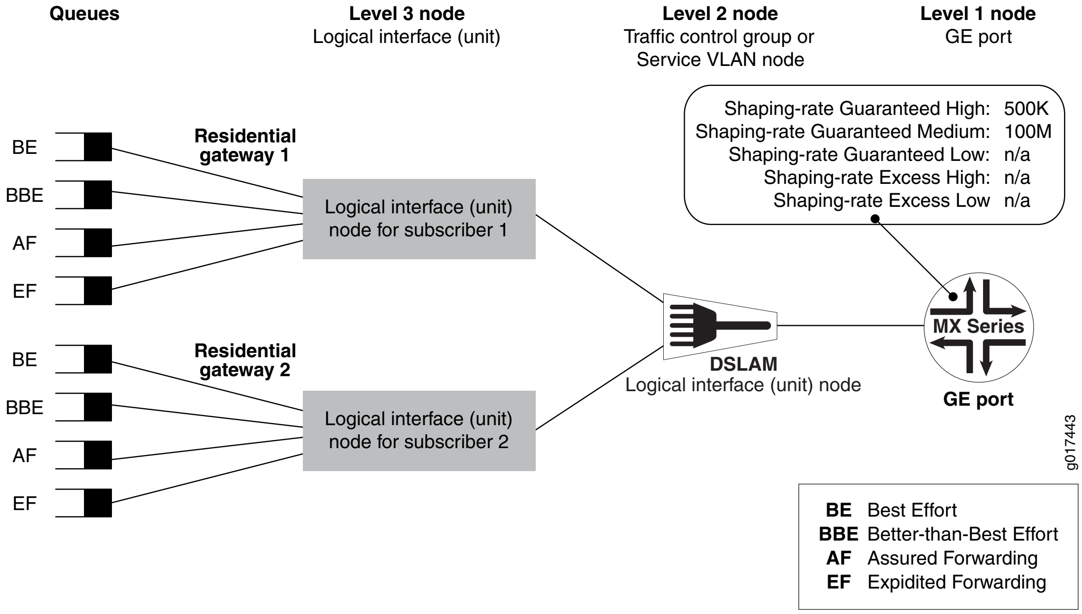


One way that the MX Series router can provide service classes for this physical network topology is shown in Figure 25 on page 384. In the figure, services such as voice and video are placed in separate forwarding classes and the services at different priority levels. For example:

- All expedited-forwarding queues are services at a priority level of guaranteed high.
- All assured-forwarding queues are services at a priority level of guaranteed medium.
- All better-than-best-effort queues are services at a priority level of excess high.
- All best-effort queues are services at a priority level of excess low.

NOTE: This list covers only one possible configuration. Others are possible and reasonable, depending on the service provider's goals. For example, best-effort and better-than-best-effort traffic can have the same priority level, with the better-then-best-effort forwarding class having a higher scheduler weight than the best-effort forwarding class. For more information on forwarding classes, see "Configuring Forwarding Classes" on page 115.

Figure 25: Scheduling Hierarchy for Trio Per-Priority Shaping



Aggregated voice traffic in this topology is shaped by applying a high-priority shaper to the port. Aggregated video traffic is shaped in the same way by applying a medium-priority shaper to the port. As long as the sum of the high- and medium-priority shapers is less than the port speed, some bandwidth is reserved for best-effort and better-then-best-effort traffic. So assured-forwarding and expedited-forwarding voice and video cannot starve best-effort and better-then-best-effort data services. One possible set of values for high-priority (guaranteed high) and medium-priority (guaranteed medium) traffic is shown in Figure 25 on page 384.

NOTE: We recommend that you do not shape delay-sensitive traffic such as voice traffic because it adds delay (latency). Service providers often use connection admission control (CAC) techniques to limit aggregated voice traffic. However, establishing a shaping rate for other traffic guards against CAC failures and can be useful in pacing extreme traffic bursts.

You must still apply the traffic control profile to the port

Per-priority shaping statements:

```
[edit class-of-service]
traffic-control-profile {
  tcp-for-ge-port {
    shaping-rate-priority-high 500k;
    shaping-rate-priority-medium 100m;
  }
}
```

Apply (attach) the traffic control profile to the physical interface (port) at the **[edit class-of-services interfaces]** hierarchy level:

```
[edit class-of-service]
```

```
interfaces {
  ge-1/0/0 {
    output-traffic-control-profile tcp-for-ge-port;
  }
}
```

Traffic control profiles with per-priority shaping rates can only be attached to interfaces that support per-priority shaping.

You can apply per-priority shaping to levels other than the level 1 physical interface (port) of the scheduler hierarchy. Per-priority shaping can also be applied at level 2, the interface set level, which would typically represent the digital subscriber link access multiplexer (DSLAM). At this level you could use per-priority shaping to limit to total amount of video traffic reaching a DSLAM, for example.

You apply (attach) the traffic control profile to an interface set at the **[edit class-of-services interfaces]** hierarchy level:

```
[edit class-of-service]
interfaces {
  interface-set svlan-1 {
    output-traffic-control-profile tcp-for-ge-port;
  }
}
```

NOTE: Although you can configure both input and output traffic control profiles, only output traffic control profiles are supported for per-priority shaping.

You can also apply per-priority shaping in a dynamic interface configuration, not in a static configuration as shown so far. To apply per-priority shaping to a dynamic configuration, configure the statements at the **[edit dynamic-profiles]** hierarchy level:

You must still apply the traffic control profile to the port

Per-priority shaping for dynamic profiles statements:

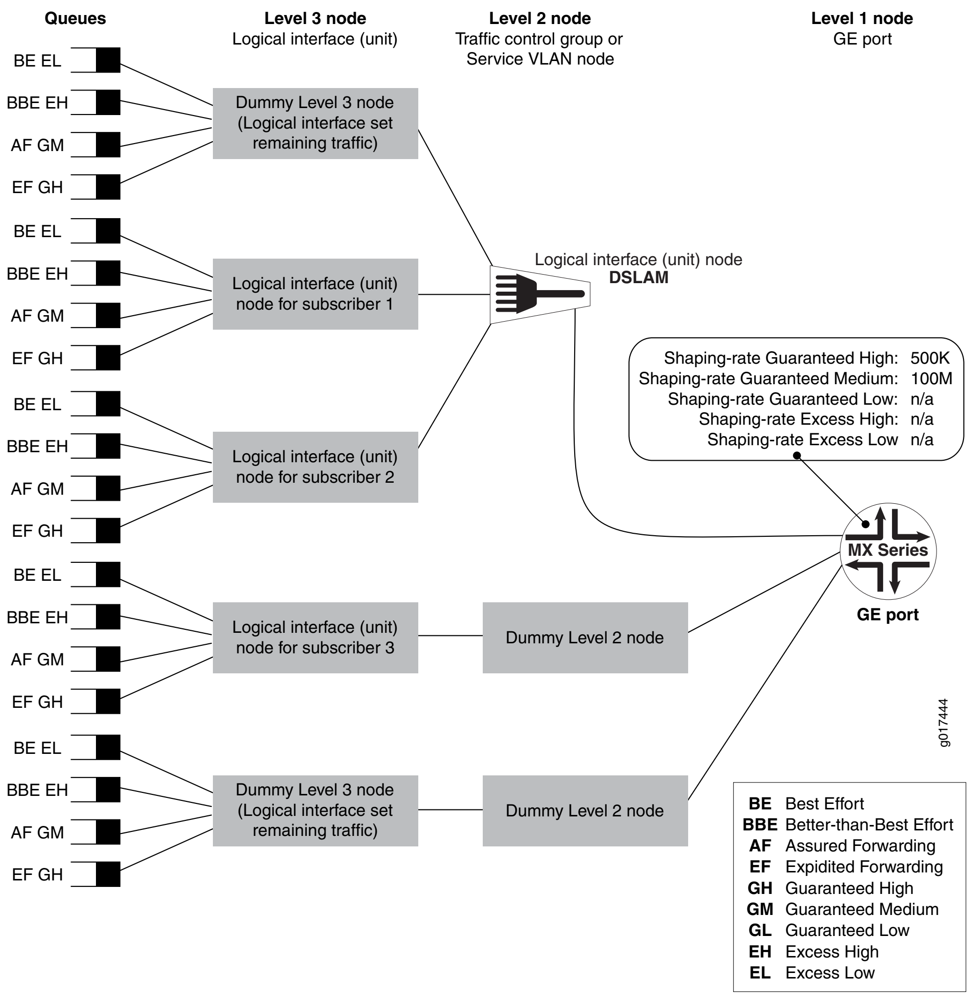
```
[edit dynamic-profiles class-of-service]
class-of-service {
  traffic-control-profile {
    tcp-for-ge-port {
      shaping-rate-priority-high 500k;
      shaping-rate-priority-medium 100m;
    }
  }
}
```

You can configure per-priority shaping for the traffic remaining with the **output-traffic-control-profile-remaining** statement on a physical port (a level 2 node) but not for an interface set (a level 3 node).

Example: Configuring Per-Priority Shaping on Trio MPC/MIC Interfaces

In practice, per-priority shaping is used with other traffic control profiles to control traffic as a whole. Consider the traffic control profile applied to the physical interface (port), as shown in Figure 26 on page 386.

Figure 26: Example Trio MPC/MIC Interface Scheduling Hierarchy



This example is more complex than those used before. In addition to a pair of subscribers in an interface set (DSLAM), the figure now adds the following:

- A dummy level 3 scheduler node (**interface-set-remaining-traffic**) that provides scheduling for interface set members that do not have explicit class-of-service parameters configured.
- A subscriber (Subscriber 3) that is not a member of an interface set. A dummy level 2 node connects Subscriber 3's level 3 node to level 1, making it appear to be at level 2.

- A dummy level 3 scheduler node (**port-remaining-traffic**) in order to provide queues for traffic that does not have explicit class-of-service parameters configured.
- A dummy level 2 scheduler node to connect level 1 and level 3 scheduler nodes. This dummy level 2 scheduler node is internal only.

This example uses a gigabit Ethernet interface with five logical interface units, each one representing one of the level 3 nodes in Figure 26 on page 386.

From the top of the figure to the bottom, the level 3 nodes are:

- Unit 3 is scheduled as a “dummy” level 3 node because unit 3 is a member of an interface set (**ifset-1**) but there is no explicit CoS configuration.
- Unit 1 is scheduled as a logical interface node for subscriber 1 because unit 1 is a member of an interface set (**ifset-1**) and has an explicit CoS configuration under the **[edit class-of-service interfaces]** hierarchy.
- Unit 2 is scheduled as a logical interface node for subscriber 2 because unit 2 is a member of an interface set (**ifset-1**) and has an explicit CoS configuration under the **[edit class-of-service interfaces]** hierarchy.
- Unit 4 is scheduled as a logical interface node for subscriber 3 because unit 4 is not a member of an interface set but has an explicit CoS configuration under the **[edit class-of-service interfaces]** hierarchy level.
- Unit 5 is scheduled by another “dummy” level 3 node, this one for remaining traffic at the port level, because unit 5 is not a member of an interface set and has no explicit CoS configuration.

In this example, per-priority shaping is applied at the physical port level. The example uses three priorities, but other parameters are possible. The example does not use shaping rates, transmit rates, excess priorities, or other options for reasons of simplicity. The example uses five forwarding classes and leaves out a network control forwarding class that would typically be included in real configurations.

The example configuration is presented in several parts:

- Interfaces configuration
- Class-of-service forwarding classes and traffic control profiles configuration
- Class-of-service interfaces configuration
- Class-of-service schedulers and scheduler map configuration

Interfaces configuration:

```
[edit]
interfaces {
  # A three member interface-set.
  interface-set ifset-1 {
    interface ge-1/1/0 {
      unit 1;
      unit 2;
      unit 3;
```

```
        }
      }
      # A ge port configured for "hierarchical-scheduling" and
      # vlans. 5 vlans are configured for the 5 level-3 scheduler
      # nodes
      #
      ge-1/1/0 {
        hierarchical-scheduler;
        vlan-tagging;
        unit 1 {
          vlan-id 1;
        }
        unit 2 {
          vlan-id 2;
        }
        unit 3 {
          vlan-id 3;
        }
        unit 4 {
          vlan-id 4;
        }
        unit 5 {
          vlan-id 5;
        }
      }
    }
```

Class-of-service forwarding classes and traffic control profiles configuration:

```
[edit class-of-service]
forwarding-classes {
  queue 0 BE priority low;
  queue 1 BBE priority low;
  queue 2 AF priority low;
  queue 3 EF priority high;
}
traffic-control-profiles {
  tcp-if-portd {
    shaping-rate-priority-high 500k;
    shaping-rate-priority-medium 100m;
  }
  tcp-if-port-rem {
    scheduler-map smap-1;
  }
  tcp-ifset-rem {
    scheduler-map smap-1;
  }
  tcp-if-unit {
    scheduler-map smap-1;
    shaping-rate 10m;
  }
}
```

Class-of-service interfaces configuration:

```
[edit class-of-service]
interfaces {
```



```
    interface-set ifset-1 {
      output-traffic-control-profile-remaining tcp-ifset-rem;
    }
    ge-1/1/0 {
      output-traffic-control-profile tcp-if-port;
      output-traffic-control-profile-remaining tcp-if-port-rem;
      unit 1 {
        output-traffic-control-profile tcp-if-unit;
      }
      unit 2 {
        output-traffic-control-profile tcp-if-unit;
      }
      # Unit 3 present in the interface config and interface-set
      # config, but is absent in this CoS config so that we can
      # show traffic that uses the interface-set
      # remaining-traffic path.
      unit 4 {
        output-traffic-control-profile tcp-if-unit;
      }
      # Unit 5 is present in the interface config, but is absent
      # in this CoS config so that we can show traffic that
      # uses the if-port remaining-traffic path.
    }
  }
```

Class-of-service schedulers and scheduler map configuration:

```
  [edit class-of-service]
  scheduler-maps {
    smap-1 {
      forwarding-class BE scheduler sched-be;
      forwarding-class BBE scheduler sched-bbe;
      forwarding-class AF scheduler sched-af;
    forwarding-class EF scheduler sched-ef;
  }
  schedulers {
    sched-be {
      priority low;
    }
    sched-bbe {
      priority low;
    }
    sched-af {
      priority medium-high;
    }
    sched-ef {
      priority high;
    }
  }
```

You can configure both a shaping rate and a per-priority shaping rate. In this case, the legacy **shaping-rate** statement specifies the maximum rate for all traffic scheduled through the scheduler. Therefore, the per-priority shaping rates must be less than or equal to the overall shaping rate. So if there is a **shaping-rate 400m** statement configured in a traffic control profile, you cannot configure a higher value for a per-priority shaping rate (such as **shaping-rate-priority-high 500m**). However, the sum of the per-priority shaping rates can exceed the overall shaping rate: for **shaping-rate 400m** you can configure both **shaping-rate-priority-high 300m** and **shaping-rate-priority-low 200m** statements.

Generally, you cannot configure a shaping rate that is smaller than the guaranteed rate (which is why it is guaranteed). However, no such restriction is placed on per-priority shaping rates unless all shaping rates are for priority high or low or medium traffic.

This configuration is allowed (per-priority rates smaller than guaranteed rate):

```
[edit class-of-service]
traffic-control-profile {
  tcp-for-ge-port {
    guaranteed-rate 500m;
    shaping-rate-priority-high 400m;
    shaping-rate-priority-medium 300m;
    shaping-rate-excess-high 100m;
  }
}
```

However, this configuration generates an error (no excess per-priority rate, so the node can never achieve its guaranteed rate):

```
[edit class-of-service]
traffic-control-profile {
  tcp-for-ge-port {
    guaranteed-rate 301m;
    shaping-rate-priority-high 100m;
    shaping-rate-priority-medium 100m;
    shaping-rate-priority-low 100m;
  }
}
```

You verify configuration of per-priority shaping with the **show class-of-service traffic-control-profile** command. This example shows shaping rates established for the high and medium priorities for a traffic control profile named **tcp-ge-port**.

```
user@host# show class-of-service traffic-control-profile
Traffic control profile: tcp-ae, Index: 22093
   Shaping rate: 3000000000
   Scheduler map: <default>

Traffic control profile: tcp-ge-port, Index: 22093
   Shaping rate priority high: 1000000000
   Shaping rate priority medium: 9000000000
   Scheduler map: <default>
```

There are no restrictions on or interactions between per-priority shaping rates and the excess rate. An excess rate (a weight) is specified as a percentage or proportion of excess bandwidth.

Table 115 on page 391 shows where traffic control profiles containing per-priority shaping rates can be attached for both per-unit schedulers and hierarchical schedulers.

Table 115: Applying Traffic Control Profiles

Type of Traffic Control Profile	Per-unit Allowed?	Hierarchical Allowed?
Port level **output-traffic-control-profile** with per-priority shaping	Yes	Yes
Port level **output-traffic-control-profile-remaining** with per-priority shaping	No	Yes
Port level **output-traffic-control-profile** and **output-traffic-control-profile-remaining** with per-priority shaping	No	Yes
Port level **input-traffic-control-profile** with per-priority shaping	No	No
Port level **input-traffic-control-profile-remaining** with per-priority shaping	No	No
Interface set **output-traffic-control-profile** with per-priority shaping	No	Yes
Interface set **output-traffic-control-profile-remaining** with per-priority shaping	No	No
Interface set **input-traffic-control-profile** with per-priority shaping	No	No
Interface set **input-traffic-control-profile-remaining** with per-priority shaping	No	No
Logical interface level **output-traffic-control-profile** with per-priority shaping	No	No
Logical interface level **input-traffic-control-profile** with per-priority shaping	No	No

Bandwidth Management for Downstream Traffic in Edge Networks Overview

In a subscriber access network, traffic with different encapsulations can be passed downstream to other customer premise equipment (CPE) through the MX Series router. Managing the bandwidth of downstream ATM traffic to Ethernet interfaces can be especially difficult because of the different Layer 2 encapsulations.

The *overhead accounting* feature enables you to shape traffic based on either frames or cells and assign a byte adjustment value to account for different encapsulations.

This feature is available on Trio MPC/MIC interfaces on MX Series routers.

Guidelines for Configuring the Shaping Mode

Frame mode is useful for adjusting downstream traffic with different encapsulations. In frame shaping mode, shaping is based on the number of bytes in the frame, without regard to cell encapsulation or padding overhead. Frame is the default shaping mode on the router.

Cell mode is useful for adjusting downstream cell-based traffic. In cell shaping mode, shaping is based on the number of bytes in cells, and accounts for the cell encapsulation and padding overhead.

When you specify cell mode, the resulting traffic stream conforms to the policing rates configured in downstream ATM switches, reducing the number of packet drops in the Ethernet network.

To account for ATM segmentation, the MX Series router adjusts all of the rates by 48/53 to account for ATM AAL5 encapsulation. In addition, the router accounts for cell padding, and internally adjusts each frame by 8 bytes to account for the ATM trailer.

Guidelines for Configuring Byte Adjustments

When the downstream traffic has different byte sizes per encapsulation, it is useful to configure a *byte adjustment* value to adjust the frame sizes. For example, you can configure the frame shaping mode and a byte adjustment value to account for differences in Layer 2 protocols for downstream Ethernet traffic.

We recommend that you specify a byte adjustment value that represents the difference between the CPE protocol overhead and B-RAS protocol overhead.

The system rounds up the byte adjustment value to the nearest multiple of 4. For example, a value of 6 is rounded to 8, and a value of –10 is rounded to –8.

You do not need to configure a byte adjustment value to account for the downstream ATM network. However, you could specify the byte value to account for additional encapsulations or decapsulations in the downstream network.

Relationship with Other CoS Features

Enabling the overhead accounting feature affects the resulting shaping rates, guaranteed rate, and excess rate parameters, if they are configured.

The overhead accounting feature also affects the egress shaping overhead feature that you can configure at the chassis level. We recommend that you use the egress shaping-overhead feature to account for the Layer 2 overhead of the outgoing interface, and use the overhead-accounting feature to account for downstream traffic with different encapsulations and cell-based networks.

When both features are configured together, the total byte adjustment value is equal to the adjusted value of the overhead-accounting feature plus the value of the egress-shaping-overhead feature. For example, if the configured byte adjustment value is 40, and the router internally adjusts the size of each frame by 8, the adjusted overhead accounting value is 48. That value is added to the egress shaping overhead of 30 for a total byte adjustment value of 78.

Related Documentation

- To configure overhead accounting for static Ethernet interfaces, see Configuring Static Shaping Parameters to Account for Overhead in Downstream Traffic Rates on page 393
- To configure overhead accounting for dynamic subscriber access, see Configuring Dynamic Shaping Parameters to Account for Overhead in Downstream Traffic Rates

Configuring Static Shaping Parameters to Account for Overhead in Downstream Traffic Rates

The overhead accounting feature enables you to account for downstream traffic that has different encapsulations or downstream traffic from cell-based equipment, such as ATM switches.

You can configure the overhead accounting feature to shape downstream traffic based on frames or cell shaping mode.

You can also account for the different byte sizes per encapsulation by configuring a byte adjustment value for the shaping mode.

To configure the shaping mode and byte adjustment value for static CoS configurations:

1. Specify the shaping mode.

   Frame shaping mode is enabled by default.

   [edit class-of-service traffic-control-profiles *profile-name*]
   user@host# **set overhead-accounting (frame-mode | cell-mode)**

2. (Optional) Specify a byte adjustment value.

   [edit class-of-service traffic-control-profiles *profile-name*
   user@host# **set overhead-accounting bytes** ***byte-value***]

BEST PRACTICE: We recommend that you specify a byte adjustment value that represents the difference between the customer premise equipment (CPE) protocol overhead and the B-RAS protocol overhead.

The available range is –120 through 124 bytes. The system rounds up the byte adjustment value to the nearest multiple of 4. For example, a value of 6 is rounded to 8, and a value of –10 is rounded to –8.

Related Documentation

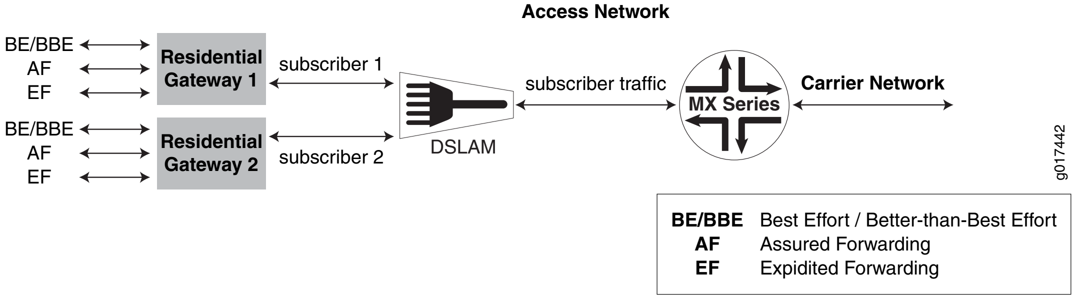
- Bandwidth Management for Downstream Traffic in Edge Networks Overview on page 391

Example: Configuring Static Shaping Parameters to Account for Overhead in Downstream Traffic Rates

This topic describes two scenarios for which you can configure static shaping parameters to account for packet overhead in a downstream network.

Figure 27 on page 394 shows the sample network that the examples reference.

Figure 27: Sample Network Topology for Downstream Traffic



Managing Traffic with Different Encapsulations

In this example, the MX Series router shown in Figure 27 on page 394 sends stacked VLAN frames to the DSLAM, and the DSLAM sends single-tagged VLAN frames to the residential gateway.

To accurately shape traffic at the residential gateway, the MX Series router must account for the different frame sizes. The difference between the stacked VLAN (S-VLAN) frames sent by the router and the single-tagged VLAN frames received at the residential gateway is a 4-byte VLAN tag. The residential gateway receives frames that are 4 bytes less.

To account for the different frame sizes, the network administrator configures the frame shaping mode with –4 byte adjustment:

1. The network administrator configure the traffic shaping parameters and attaches them to the interface.

   Enabling the overhead accounting feature affects the resulting shaping rate, guaranteed rate, and excess rate parameters, if they are configured.

```
[edit]
class-of-service {
  traffic-control-profiles {
    tcp-example-overhead-accounting-frame-mode {
      shaping-rate 10m;
      shaping-rate-priority-high 4m;
      guaranteed-rate 2m;
      excess-rate percent 50;
      overhead-accounting frame-mode bytes -4;
    }
  }
  interfaces {
    ge-1/0/0 {
      output-traffic-control-profile tcp-example-overhead-accounting-frame-mode;
    }
  }
 }
}
```

2. The network administrator verifies the adjusted rates.

```
user@host# show class-of-service traffic-control-profile
```



```
Traffic control profile: tcp-example-overhead-accounting-frame-mode, Index:
 61785
Shaping rate: 10000000
Shaping rate priority high: 4000000
Excess rate 50
Guaranteed rate: 2000000
Overhead accounting mode: Frame Mode
Overhead bytes: –4
```

Managing Downstream Cell-Based Traffic

In this example, the DSLAM and residential gateway shown in Figure 27 on page 394 are connected through an ATM cell-based network. The MX Series router sends Ethernet frames to the DSLAM, and the DSLAM sends ATM cells to the residential gateway.

To accurately shape traffic at the residential gateway, the MX Series router must account for the different physical network characteristics.

To account for the different frame sizes, the network administrator configures the cell shaping mode with –4 byte adjustment:

1. Configure the traffic shaping parameters and attach them to the interface.

   Enabling the overhead accounting feature affects the resulting shaping rate, guaranteed rate, and excess rate parameters, if they are configured.

   ```
   [edit]
   class-of-service {
     traffic-control-profiles {
       tcp-example-overhead-accounting-cell-mode {
         shaping-rate 10m;
         shaping-rate-priority-high 4m;
         guaranteed-rate 2m;
         excess-rate percent 50;
         overhead-accounting cell-mode;
         }
       }
       interfaces {
         ge-1/0/0 {
           output-traffic-control-profile tcp-example-overhead-accounting-cell-mode;
         }
       }
     }
   }
   ```

2. Verify the adjusted rates.

   ```
   user@host#show class-of-service traffic-control-profile

   Traffic control profile: tcp-example-overhead-accounting-cell-mode, Index:
    61785
   Shaping rate: 10000000
   Shaping rate priority high: 4000000
   Excess rate 50
   Guaranteed rate: 2000000
   Overhead accounting mode: Cell Mode
   Overhead bytes: 0
   ```

To account for ATM segmentation, the MX Series router adjusts all of the rates by 48/53 to account for ATM AAL5 encapsulation. In addition, the router accounts for cell padding, and internally adjusts each frame by 8 bytes to account for the ATM trailer.

Related Documentation

- Configuring Static Shaping Parameters to Account for Overhead in Downstream Traffic Rates on page 393

Intelligent Oversubscription on the Trio DPC and MPC/MIC Interfaces Overview

On the Trio DPC and MPC/MIC interfaces, as on other types of interface hardware, arriving packets are assigned to one of two preconfigured traffic classes (network control and best effort) based on their header types and destination media access control (MAC) address. Oversubscription, the situation when the incoming packet rate is much higher than the Packet Forwarding Engine and system can handle, can cause key packets to be dropped and result in a flurry of resends, making the problem worse. However, the Trio DPC and MPC/MIC interfaces handle oversubscription more intelligently and drops lower priority packets when oversubscription occurs. Protocols such as routing protocols are classified as network control. Protocols such as telnet, FTP, and SSH are classified as best effort. No configuration is necessary.

The following frames and packets are assigned to the network control traffic class:

- ARPs: Ethertype **0x0806** for ARP and **0x8035** for dynamic RARP
- IEEE 802.3ad Link Aggregation Control Protocol (LACP): Ethertype **0x8809** and **0x01** or **0x02** (subtype) in first data byte
- IEEE 802.1ah: Ethertype **0x8809** and subtype **0x03**
- IEEE 802.1g: Destination MAC address **0x01–80–C2–00–00–02** with Logical Link Control (LLC) **0xAAAA03** and Ethertype **0x08902**
- PVST: Destination MAC address **0x01–00–0C–CC–CC–CD** with LLC **0xAAAA03** and Ethertype **0x010B**
- xSTP: Destination MAC address **0x01–80–C2–00–00–00** with LLC **0x424203**
- GVRP: Destination MAC address **0x01–80–C2–00–00–21** with LLC **0x424203**
- GMRP: Destination MAC address **0x01–80–C2–00–00–20** with LLC **0x424203**
- IEEE 802.1x: Destination MAC address **0x01–80–C2–00–00–03** with LLC **0x424203**
- Any per-port **my-mac** destination MAC address
- Any configured global Integrated Bridging and Routing (IRB) **my-mac** destination MAC address

In addition, the following Layer 3 control protocols are assigned to the network control traffic class:

- IGMP query and report: Ethertype **0x0800** and carrying an IPv4 protocol or IPv6 next header field set to 2 (IGMP)
- IGMP DVMRP: IGMP field version = **1** and type = **3**
- IPv4 ICMP: Ethertype **0x0800** and IPv4 protocols = **1** (ICMP)
- IPv6 ICMP: Ethertype **0x86DD** and IPv6 next header field = **0x3A** (ICMP)
- IPv4 or IPv6 OSPF: Ethertype **0x0800** and IPv4 protocol field or IPv6 next header field = **89** (OSPF)
- IPv4 or IPv6 VRRP: IPv4 Ethertype **0x0800** or IPv6 Ethertype **0x86DD** and IPv4 protocol field or IPv6 next header field = **112** (IGMP)
- IPv4 or IPv6 RSVP: IPv4 Ethertype **0x0800** or IPv6 Ethertype **0x86DD** and IPv4 protocol field or IPv6 next header field = **46** or **134**
- IPv4 or IPv6 PIM: IPv4 Ethertype **0x0800** or IPv6 Ethertype **0x86DD** and IPv4 protocol field or IPv6 next header field = **103**
- IPv4 or IPv6 IS-IS: IPv4 Ethertype **0x0800** or IPv6 Ethertype **0x86DD** and IPv4 protocol field or IPv6 next header field = **124**
- IPv4 router alert: IPv4 Ethertype **0x0800** and IPv4 option field = **0x94** (router alert)

Also, the following Layer 4 control protocols are assigned to the network control traffic class:

- IPv4 and IPv6 BGP: IPv4 Ethertype **0x0800** or IPv6 Ethertype **0x86DD**, TCP port = **179**, and carrying an IPv4 protocol or IPv6 next header field set to 6 (TCP)
- IPv4 and IPv6 LDP: IPv4 Ethertype **0x0800** or IPv6 Ethertype **0x86DD**, TCP or UDP port = **646**, and carrying an IPv4 protocol or IPv6 next header field set to 6 (TCP) or 17 (UDP)
- IPv4 UDP/L2TP control frames: IPv4 Ethertype **0x0800**, UDP port = **1701**, and carrying an IPv4 protocol field set to 17 (UDP)
- DHCP: Ethertype **0x0800**, IPv4 protocol field set to 17 (UDP), and UDP destination port = **0x43** (DHCP service) or **0x44** (DHCP host)
- IPv4 or IPv6 UDP/BFD: Ethertype **0x0800**, UDP port = **3784**, and IPv4 protocol field or IPv6 next header field set to 17 (UDP)

Finally, any PPP encapsulation (Ethertype **0x8863** (PPPoE Discovery) or **0x8864** (PPP0E Session Control)) is assigned to the network control traffic class (queue 3).

NOTE: These classifications are preconfigured.

PART 4

CoS Configuration for Specific Transports

CHAPTER 23

Configuring Schedulers on Aggregated Ethernet and SONET/SDH Interfaces

This topic discusses the following:

Configuring Schedulers on Aggregated Interfaces

You can apply a class-of-service (CoS) configuration to aggregated Ethernet and aggregated SONET/SDH interfaces. The CoS configuration applies to all member links included in the aggregated interface. You cannot apply different CoS configurations to the individual member links.

You can configure shaping for aggregated Ethernet interfaces that use interfaces originating from Gigabit Ethernet IQ2 PICs. However, you cannot enable shaping on aggregated Ethernet interfaces when there is a mixture of ports from Intelligent Queuing (IQ) and Intelligent Queuing 2 (IQ2) PICs in the same bundle.

You cannot configure a shaping rate and guaranteed rate on an aggregated Ethernet interface with member interfaces on IQ or IQ2 PICs. The commit will fail. These statements are allowed only when the member interfaces are Enhanced Queuing DPC Gigabit Ethernet interfaces.

To view the summation of the queue statistics for the member links of an aggregate interface, issue the **show interfaces queue** command. To view the queue statistics for each member link, issue the **show interfaces queue *aggregated-interface-name*** command.

To configure CoS schedulers on aggregated interfaces, include the following statements at the **[edit class-of-service]** hierarchy level:

```
[edit class-of-service]
interfaces {
  interface-name {
    scheduler-map map-name;
    unit logical-unit-number {
```

```
            scheduler-map map-name;
          }
        }
      }
      scheduler-maps {
        map-name {
          forwarding-class class-name scheduler scheduler-name;
        }
      }
      schedulers {
        scheduler-name {
          buffer-size (percent percentage | remainder | temporal microseconds);
          drop-profile-map loss-priority (any | low | medium-low | medium-high | high)protocol
            (any | non-tcp | tcp) drop-profile profile-name;
          excess-priority (low | high);
          excess-rate percent percentage;
          priority priority-level;
          transmit-rate (rate | percent percentage | remainder) <exact>;
        }
      }
```

Limitations on CoS for Aggregated Interfaces

Both Ethernet and SONET/SDH interfaces can be aggregated. The limitations covered here apply to both.

There are some restrictions when you configure CoS on aggregated Ethernet and SONET/SDH interfaces:

- Chassis scheduling, described in "Applying Scheduler Maps to Packet Forwarding Component Queues" on page 194, is not supported on aggregated interfaces, because a chassis scheduler applies to the entire PIC and not just to one interface.
- An aggregated interface is a pseudo-interface. Therefore, CoS queues are not associated with the aggregated interface. Instead, CoS queues are associated with the member link interfaces of the aggregated interface.
- When you apply CoS parameters to the aggregated interface, they are applied to the CoS queues of the member link interfaces. You can apply CoS classifiers and rewrite rules directly to the member link interfaces, and the software uses the values you configure.
- When you apply scheduler maps to member link interfaces, the software cannot always use the values you configure because the speed of the aggregated interface is the sum of the speeds of its member link interfaces.

When the scheduler map of the aggregate interface has schedulers configured for absolute transmit rate, the scheduler for the member link interfaces is scaled to the speed of each member link interface. Each member link interface has an automatic scheduler map that is not visible in the CLI. This scheduler map is allocated when the member link is added to the aggregate interface and is deleted when the member link is removed from the aggregate interface.

- If you configure the scheduler transmit rate of the aggregate interface as an absolute rate, the software uses the following formula to scale the transmit rate of each member link:

  transmit rate of member link interface =
  (configured transmit rate of aggregate interface /
  total speed of aggregate interface) *
  (total speed of member link interface / total configured percent) * 100

- If you configure the scheduler transmit rate of the aggregate interface as a percentage, the software uses the following formula to scale the transmit rate of each member link:

  transmit rate percent of member link interface =
  (configured transmit rate percent of aggregate interface /
  total configured percent) * 100

The total configured percent is the sum of the configured transmit rate of all schedulers in terms of percentage of the total speed of the aggregate interface.

For more information, see “Examples: Configuring CoS on Aggregated Interfaces” on page 403.

- All the other parameters for the schedulers, including priority, drop profile, and buffer size, are copied without change from the scheduler of the aggregated interface to the member link interfaces.
- The configuration related to the logical interfaces, including classifiers and rewrite rules, is copied from the aggregated logical interface configuration to the member link logical interfaces.
- For the scheduler map applied to an aggregated interface, if you configure a transmission rate in absolute terms, then the traffic of all the member link interfaces might be affected if any of the member link interfaces go up or down.

Examples: Configuring CoS on Aggregated Interfaces

This example illustrates how CoS scheduler parameters are configured and applied to aggregated interfaces.

Applying Scaling Formula to Absolute Rates

Configure queues as follows when the total speed of member link interfaces is 100 Mbps (the available bandwidth is 100 Mbps):

```
[edit class-of-service]
schedulers {
  be {
    transmit-rate 10m;
  }
  af {
    transmit-rate 20m;
  }
  ef {
    transmit-rate 80m;
  }
  nc {
```

```
        transmit-rate 30m;
    }
}
```

The total configured transmit rates of the aggregated interface is **10m** + **20m** + **80m** + **30m** = 140 Mbps, meaning the transmit rate is overconfigured by 40 percent. Therefore, the software scales down the configuration to match the 100 Mbps of available bandwidth, as follows:

```
be = (10/140) * 100 = 7 percent of 100 Mbps = 7 Mbps
af = (20/140) * 100 = 14 percent of 100 Mbps = 14 Mbps
ef = (80/140) * 100 = 57 percent of 100 Mbps = 57 Mbps
nc = (30/140) * 100 = 21 percent of 100 Mbps = 21 Mbps
```

Applying Scaling Formula to Mixture of Percent and Absolute Rates

Configure the following mixture of percent and absolute rates:

```
[edit class-of-service]
schedulers {
    be {
        transmit-rate 20 percent;
    }
    af {
        transmit-rate 40 percent;
    }
    ef {
        transmit-rate 150m;
    }
    nc {
        transmit-rate 10 percent;
    }
}
```

Assuming 300 Mbps of available bandwidth, the configured percentages correlate with the following absolute rates:

```
schedulers {
    be {
        transmit-rate 60m;
    }
    af {
        transmit-rate 120m;
    }
    ef {
        transmit-rate 150m;
    }
    nc {
        transmit-rate 30m;
    }
}
```

The software scales the bandwidth allocation as follows:

```
be = (60/360) * 100 = 17 percent of 300 Mbps = 51 Mbps
af = (120/360) * 100 = 33 percent of 300 Mbps = 99 Mbps
ef = (150/360) * 100 = 42 percent of 300 Mbps = 126 Mbps
nc = (30/360) * 100 = 8 percent of 300 Mbps = 24 Mbps
```

Configuring an Aggregated Ethernet Interface

Configure an aggregated Ethernet interface with the following scheduler map:

```
[edit class-of-service]
scheduler-maps {
  aggregated-sched {
    forwarding-class be scheduler be;
    forwarding-class af scheduler af;
    forwarding-class ef scheduler ef;
    forwarding-class nc scheduler nc;
  }
}
schedulers {
  be {
    transmit-rate percent 10;
    buffer-size percent 25;
  }
  af {
    transmit-rate percent 20;
    buffer-size percent 25;
  }
  ef {
    transmit-rate 80m;
    buffer-size percent 25;
  }
  nc {
    transmit-rate percent 30;
    buffer-size percent 25;
  }
}
```

In this case, the transmission rate for the member link scheduler map is as follows:

- **be**—7 percent
- **af**—14 percent
- **ef**—57 percent
- **nc**—21 percent

If you add a Fast Ethernet interface to the aggregate, the aggregate bandwidth is 200 Mbps, and the transmission rate for the member link scheduler map is as follows:

- **be**—10 percent
- **af**—20 percent
- **ef**—40 percent
- **nc**—30 percent

Example: Configuring Scheduling Modes on Aggregated Interfaces

You can configure class-of-service parameters, such as queuing or shaping parameters on aggregated interfaces, in either link-protect or non-link-protect mode. You can configure these parameters for per-unit schedulers, hierarchical schedulers, or shaping at the physical and logical interface level. You can control the way these parameters are applied by configuring the aggregated interface to operate in **scale** or **replicate** mode.

You can apply these parameters on the following routers:

- MX Series routers with EQ DPCs
- M120 or M320 routers
- T Series routers with IQ2 PICs

You can configure the applied parameters for aggregated interfaces operating in non-link-protected mode. In link-protected mode, only one link in the bundle is active at a time (the other link is a backup link) so schedulers cannot be scaled or replicated. In non-link-protected mode, all the links in the bundle are active and send traffic; however, there is no backup link. If a link fails or is added to the bundle in non-link-protected mode, the links' traffic is redistributed among the active links.

To set the scheduling mode for aggregated interfaces, include the **scale** or **replicate** option of the **member-link-scheduler** statement at the **[edit class-of-service interfaces ea*n*]** hierarchy level, where *n* is the configured number of the interface:

```
[edit class-of-service interfaces ean]
member-link-scheduler (replicate | scale);
```

By default, if you do not include the **member-link-scheduler** statement, scheduler parameters are applied to the member links in the **scale** mode (also called "equal division mode").

The aggregated Ethernet interfaces are otherwise configured as usual. For more information on configuring aggregated Ethernet interfaces, see the *Junos OS Network Interfaces Configuration Guide*.

The following examples set **scale** mode on the **ae0** interface and **replicate** mode on the **ae1** interface.

```
[edit class-of-service]
interfaces ae0 {
  member-link-scheduler scale;
}

[edit class-of-service]
interfaces ae1 {
  member-link-scheduler replicate;
}
```

NOTE: The **member-link-scheduler** statement only appears for aggregated interfaces. You configure this statement for aggregated interfaces in non-link-protected mode. For more information about link protection modes, see the *Network Interfaces Configuration Guide*.

Aggregated interfaces support both hierarchical and per-unit schedulers. For more information about configuring schedulers, see "Configuring Schedulers" on page 148.

When interface parameters are using the **scale** option of the **member-link-scheduler** statement, the following parameters under the **[edit class-of-service traffic-control-profiles *traffic-control-profile-name*]** configuration are scaled on egress when hierarchical schedulers are configured:

- **shaping-rate** (PIR)
- **guaranteed-rate** (CIR)
- **delay–buffer-rate**

When interface parameters are using the **scale** option of the **member-link-scheduler** statement, the following parameters under the **[edit class-of-service schedulers *scheduler-name*]** configuration are scaled on egress when per-unit schedulers are configured:

- **transmit-rate**
- **buffer-size**

NOTE: You cannot apply a hierarchical scheduler at the interface set level for an **ae** interface. (Interface sets cannot be configured under an **ae** interface.)

The following configuration parameters are not supported on **ae** interfaces in non-link-protection mode:

- Input scheduler maps
- Input traffic control profiles
- Input shaping rates

The following configuration conventions are also not supported:

- Scaling of the **input-traffic-control-profile-remaining** statement.
- The **scheduler-map-chassis** statement and the **derived** option for the **ae** interface. Chassis scheduler maps should be applied under the physical interfaces.
- Dynamic and demux interfaces are not supported as part of the **ae** bundle.

Depending on the whether the **scale** or **replicate** option is configured, the **member-link-scheduler** statement operates in either scaled mode (also called “equal division mode”) or replicated mode, respectively.

In scaled mode, a VLAN can have multiple flows that can be sent over multiple member links of the **ae** interface. Likewise, a member link can receive traffic from any VLAN in the **ae** bundle. In scaled mode, the physical interface bandwidth is divided equally among all member links of the **ae** bundle.

In scaled mode, the following scheduler parameter values are divided equally among the member links:

- When the parameters are configured using traffic control profiles, then the parameters scaled are the shaping rate, guaranteed rate, and delay buffer rate.
- When the parameters are configured using scheduler maps, then the parameters scaled are the transmit rate and buffer size.

For example, consider an **ae** bundle between routers R1 and R2 consisting of three links. These are **ge-0/0/1**, **ge-0/0/2** and **ge-0/0/3** (**ae0**) on R1; and **ge-1/0/0**, **ge-1/0/1**, and **ge-1/0/2** (**ae2**) on R2. Two logical interfaces (units) are also configured on the **ae0** bundle on R1: **ae0.0** and **ae0.1**.

On **ae0**, traffic control profiles on R1 are configured as follows:

- **ae0** (the physical interface level) has a PIR of 450 Mbps.
- **ae0.0** (VLAN 100 at the logical interface level) has a PIR of 150 Mbps and a CIR of 90 Mbps.
- **ae0.1** (VLAN 200 at the logical interface level) has a PIR of 90 Mbps and a CIR of 60 Mbps.

In scaled mode, the **ae0** PIR is first divided among the member physical interfaces. Because there are three members, each receives 450 / 3 = 150 Mbps as a derived value. So the scaled PIR for the members interfaces is 150 Mbps each.

However, there are also two logical interfaces (**ae0.0** and **ae0.1**) and VLANs (100 and 200) on **ae0**. Traffic can leave on any of the three physical interfaces (**ge-0/0/1**, **ge-0/0/2**, or **ge-0/0/3**) in the bundle. Therefore, two derived logical interfaces are added to the member links to represent the two VLANs.

There are now six logical interfaces on the physical interfaces of the links making up the **ae** bundle, one set for VLAN 100 and the other for VLAN 200:

- **ge-0/0/1.0** and **ge-0/0/1.1**
- **ge-0/0/2.0** and **ge-0/0/2.1**
- **ge-0/0/3.0** and **ge-0/0/3.1**

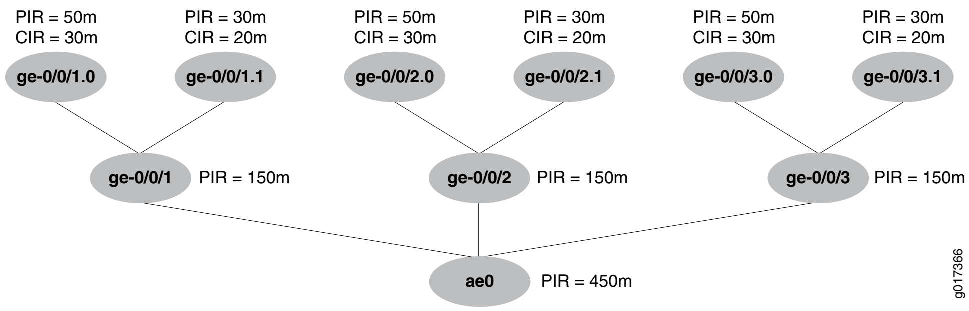
The traffic control profile parameters configured on **ae0.0** are divided across all the underlying logical interfaces (the unit 0s). In the same way, the traffic control profile parameters configured on **ae0.1** are divided across all the underlying logical interfaces (the unit 1s).

Therefore, the derived values of the scaled parameters on the interfaces are:

- For **ge-0/0/1.0** and **ge-0/0/2.0** and **ge-0/0/3.0**, each CIR = 90 / 3 = 30 Mbps, and each PIR = 150 / 3 = 50 Mbps.
- For **ge-0/0/1.1** and **ge-0/0/2.1** and **ge-0/0/3.1**, each CIR = 60 / 3 = 20 Mbps, and each PIR = 90 / 3 = 30 Mbps.

The scaled values are shown in Figure 28 on page 409.

Figure 28: Scaled Mode for Aggregated Ethernet Interfaces



In scaled mode, when a new member link is added to the bundle, or an existing member link is either removed or fails, then the scaling factor (based on the number of active links) is recomputed and the new scheduler or traffic control profile parameters are reassigned. Only the PIR, CIR, and buffer parameters are recomputed: all other parameters are simply copied at each level.

NOTE: In **show class-of-service scheduler-map** commands, values derived in scaled mode instead of explicitly configured are flagged with &**sf***n* suffix, where *n* indicates the value of the scaling factor.

The following sample shows the output for the scheduler map named **smap-all-abs** with and without a scaling factor:

```
user@host> show class-of-service scheduler-map
Scheduler map: smap-all-abs, Index: 65452

Scheduler: q0_sch_abs, Forwarding class: be, Index: 6775
Transmit rate: 40000000 bps, Rate Limit: none, Buffer size: remainder,
Priority: low
    Excess Priority: unspecified
    Drop profiles:
      Loss priority   Protocol    Index    Name
      Low             any             1    <default-drop-profile>
      Medium low      any             1    <default-drop-profile>
      Medium high     any             1    <default-drop-profile>
      High            any             1    <default-drop-profile>

user@host> show class-of-service scheduler-map
Scheduler map: smap-all-abs, Index: 65452

Scheduler: q0_sch_abs&**sf**3, Forwarding class: be, Index: 2128
Transmit rate: 13333333 bps, Rate Limit: none, Buffer size: remainder,
Priority: low
    Excess Priority: unspecified
    Drop profiles:
      Loss priority   Protocol    Index    Name
      Low             any             1    <default-drop-profile>
      Medium low      any             1    <default-drop-profile>
      Medium high     any             1    <default-drop-profile>
      High            any             1    <default-drop-profile>
```


NOTE: There can be multiple scheduler maps created with different scaling factors, depending on when the child interfaces come up. For example, if there are only two active children on a parent interface, a new scheduler map with a scaling factor of 2 is created. The scheduler map name is **smap-all-abs&**sf**2**.

In replicated mode, in contrast to scaled mode, the configured scheduler parameters are simply replicated, not divided, among all member links of the **ae** bundle.

In replicated mode, the following scheduler parameter values are replicated among the member links and logical interfaces:

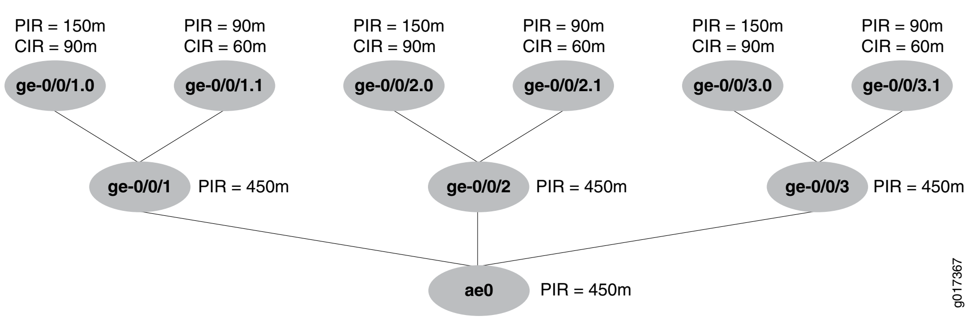
- When the parameters are configured using traffic control profiles, then the parameters replicated are the shaping rate, guaranteed rate, and delay buffer rate.
- When the parameters are configured using scheduler maps, then the parameters replicated are the transmit rate and buffer size.

If the scheduler parameters in the example configuration between routers R1 and R2 are applied with the **member-link-scheduler replicate** statement and option, the following parameters are applied:

- The **ae0** PIR is copied among the member physical interfaces. Each receives 450 Mbps as a PIR.
- For each logical interface unit **.0**, the configured PIR and CIR for **ae0.0** is replicated (copied). Each logical interface unit **.0** receives a PIR of 150 Mbps and a CIR of 90 Mbps.
- For each logical interface unit **.1**, the configured PIR and CIR for **ae0.1** is replicated (copied). Each logical interface unit **.1** receives a PIR of 90 Mbps and a CIR of 60 Mbps.

The replicated values are shown in Figure 29 on page 410.

Figure 29: Replicated Mode for Aggregated Ethernet Interfaces



In replicated mode, when a new member link is added to the bundle, or an existing member link is either removed or fails, the values are either copied or deleted from the required levels.

CHAPTER 24

Configuring CoS on ATM Interfaces

This topic discusses the following:

CoS on ATM Interfaces Overview

The ATM2 intelligent queuing (IQ) interface allows multiple IP queues into each virtual circuit (VC). On Juniper Networks M Series Multiservice Edge Routers (except the M320 router), a VC tunnel can support four class-of-service (CoS) queues. On M320 routers and T Series Core Routers, for all ATM2 IQ PICs except the OC48 PIC, a VC tunnel can support eight CoS queues. Within a VC tunnel, the weighted round-robin (WRR) algorithm schedules the cell transmission of each queue. You can configure the queue admission policies, such as early packet discard (EPD) or weighted random early detection (WRED), to control the queue size during congestion.

For information about CoS components that apply generally to all interfaces, see “CoS Overview” on page 3. For general information about configuring ATM interfaces, see the *Junos OS Network Interfaces Configuration Guide*.

To configure ATM2 IQ VC tunnel CoS components, include the following statements at the **[edit interfaces at-*fpc/pic/port*]** hierarchy level:

```
[edit chassis fpc slot-number pic pic-number]
max-queues-per-interface number;

[edit interfaces at-fpc/pic/port]
atm-options {
   linear-red-profiles profile-name {
      high-plp-max-threshold percent;
      low-plp-max-threshold percent;
      queue-depth cells high-plp-threshold percent low-plp-threshold percent;
   }
   plp-to-clp;
   scheduler-maps map-name {
      forwarding-class class-name {
         epd-threshold cells plp1 cells;
         linear-red-profile profile-name;
         priority (high | low);
         transmit-weight (cells number | percent number);
      }
      vc-cos-mode (alternate | strict);
   }
}
unit logical-unit-number {
   atm-scheduler-map (map-name | default);
   family family {
      address address {
         destination address;
      }
   }
   plp-to-clp;
   shaping {
      (cbr rate | rtvbr peak rate sustained rate burst length  | vbr peak rate sustained rate
         burst length);
   }
   vci vpi-identifier.vci-identifier;
}
```

Configuring Linear RED Profiles on ATM Interfaces

Linear random early detection (RED) profiles define CoS virtual circuit drop profiles. You can configure up to 32 linear RED profiles per port. When a packet arrives, RED checks the queue fill level. If the fill level corresponds to a nonzero drop probability, the RED algorithm determines whether to drop the arriving packet.

To configure linear RED profiles, include the **linear-red-profiles** statement at the **[edit interfaces at-*fpc/pic/port* atm-options]** hierarchy level:

```
[edit interfaces at-fpc/pic/port atm-options]
linear-red-profiles profile-name {
   high-plp-max-threshold percent;
   low-plp-max-threshold percent;
   queue-depth cells high-plp-threshold percent low-plp-threshold percent;
}
```


The **queue-depth**, **high-plp-threshold**, and **low-plp-threshold** statements are mandatory.

You can define the following options for each RED profile:

- **high-plp-max-threshold**—Define the drop profile fill-level for the high packet loss priority (PLP) CoS VC. When the fill level exceeds the defined percentage, all packets with high PLP are dropped.
- **low-plp-max-threshold**—Define the drop profile fill-level for the low PLP CoS VC. When the fill level exceeds the defined percentage, all packets with low PLP are dropped.
- **queue-depth**—Define maximum queue depth in the CoS VC drop profile. Packets are always dropped beyond the defined maximum. The range you can configure is from 1 through 64,000 cells.
- **high-plp-threshold**—Define CoS VC drop profile fill-level percentage when linear RED is applied to cells with high PLP. When the fill level exceeds the defined percentage, packets with high PLP are randomly dropped by RED.
- **low-plp-threshold**—Define CoS VC drop profile fill-level percentage when linear RED is applied to cells with low PLP. When the fill level exceeds the defined percentage, packets with low PLP are randomly dropped by RED.

Configuring ATM Scheduler Support for Ethernet VPLS over ATM Bridged Interfaces

You can configure ATM scheduler maps on Ethernet VPLS over bridged ATM interfaces.

Before you begin, you must have done the following tasks:

- Properly configured the router basics
- Verfied you have support for VPLS and routing instance configuration
- Installed ATM II IQ PICs

When you configure ATM scheduler maps on Etherent VPLS over bridged ATM interfaces, you can assign ATM traffic to various forwarding classes and queues. This feature is only available with the ATM II IQ PIC with Ethernet VPLS-over-ATM encapsulation.

The configuration takes place in four steps: define the scheduler map for ATM options on the interface, set the encapsulation type to Ethernet VPLS over ATM LLC, attach the scheduler map to the logical interface (unit), and include the interface in the VPLS routing instance configuration.

To configure ATM scheduler maps on Etherent VPLS over bridged ATM interfaces:

1. Define the scheduler map for ATM options on the interface:

   [edit interfaces at-*fpc/pic/port* atm-options]
   user@host# set pic-type atm2
   user@host# set vpi *vpi-number*
   user@host# set scheduler-maps *scheduler-map-name* forwarding-class
     forwarding-class-name (forwarding-class option statements)
   (repeat last set command as necessary)

2. Set the encapsulation type to Ethernet VPLS over ATM LLC:

   [edit interfaces at-*fpc/pic/port* unit *unit-number*]
   user@host# set encapsulation ether-vpls-over-atm-llc
   user@host# set vci *vci-number*

3. Attach the scheduler map to the logical interface (unit):

   [edit interfaces at-*fpc/pic/port* unit *unit-number*]
   user@host# set atm-scheduler-map *scheduler-map-name*

4. Include the interface in the VPLS routing instance configuration:

   [edit routing-instanceinterfaces *routing-instance-name*]
   user@host# set interface *atm-interface*
   user@host# set route-distinguisher *value*
   user@host# set vrf-target *target-value*
   user@host# set protocols vpls site-range *value*
   user@host# set protocols vpls site *site-name* site-identifier *number*

When you are done, the configuration statements you added should look like the listings below.

1. The scheduler map for ATM options on the interface:

```
[edit interfaces at-fpc/pic/port atm-options]
pic-type atm2;
vpi vpi-number;
scheduler-maps {
   scheduler-map-name {
      forwarding-class forwarding-class-name {
         (forwarding-class option statements);
      }
   }
}
```

2. The encapsulation type to Ethernet VPLS over ATM LLC:

```
[edit interfaces at-fpc/pic/port unit unit-number]
encapsulation ether-vpls-over-atm-llc;
vci vci-number;
```

3. The scheduler map to the logical interface (unit):

```
[edit interfaces at-fpc/pic/port unit unit-number]
atm-scheduler-map scheduler-map-name;
```

4. The interface in the VPLS routing instance configuration:

```
[edit routing-instanceinterfaces routing-instance-name]
interface atm-interface;
route-distinguisher value;
vrf-target target-value;
protocols {
   vpls {
      site-range value;
      site site-name {
         site-identifer number;
      }
   }
```



```
}
```

Related Documentation

- Example: Configuring ATM Scheduler Support for Ethernet VPLS over ATM Bridged Interfaces on page 415

Example: Configuring ATM Scheduler Support for Ethernet VPLS over ATM Bridged Interfaces

The following example configures an ATM scheduler map named **cos-vpls** and attaches it to the ATM interface **at-1/0/0.0**, configures **ether-vpls-over-atm-llc** encapsulation, attaches the **cos-vpls** scheduler map to the logical interface (unit), and configures the ATM interface **at-1/0/0.0** as part of a VPLS routing instance named **cos-vpls-1**.

```
[edit]
interfaces {
  at-1/0/0 {
    atm-options {
      pic-type atm2;
      vpi 0;
      scheduler-maps {
        cos0 {
          forwarding-class assured-forwarding {
            priority low;
            transmit-weight percent 10;
          }
          forwarding-class best-effort {
            priority low;
            transmit-weight percent 20;
          }
          forwarding-class expedited-forwarding {
            priority low;
            transmit-weight percent 30;
          }
          forwarding-class network-control {
            priority high;
            transmit-weight percent 40;
          }
        }
      }
    }
    unit 0 {
      encapsulation ether-vpls-over-atm-llc;
      vci 0.1000;
      shaping {
        cbr 33k;
      }
      atm-scheduler-map cos0;
    }
  }
}

[edit]
routing-instances {
```



```
    cos-vpls-1 {
        instance-type vpls;
        interface at-1/0/0.0;
        route-distinguisher 10.255.245.51:1;
        vrf-target target:1234:1;
        protocols {
            vpls {
                site-range 10;
                no-tunnel-services;
                site vpls-1-site-1 {
                    site-identifier 1;
                }
            }
        }
    }
}
```

Related Documentation

- Configuring ATM Scheduler Support for Ethernet VPLS over ATM Bridged Interfaces on page 413

Configuring Scheduler Maps on ATM Interfaces

To define a scheduler map, you associate it with a forwarding class. Each class is associated with a specific queue, as follows:

- **best-effort**—Queue 0
- **expedited-forwarding**—Queue 1
- **assured-forwarding**—Queue 2
- **network-control**—Queue 3

NOTE: For M320 and T Series routers only, you can configure more than four forwarding classes and queues. For more information, see “Enabling Eight Queues on ATM Interfaces” on page 418.

When you configure an ATM scheduler map, the Junos OS creates these CoS queues for a VC. The Junos OS prefixes each packet delivered to the VC with the next-hop rewrite data associated with each queue.

To configure an ATM scheduler map, include the **scheduler-maps** statement at the **[edit interfaces at-*fpc/pic/port* atm-options]** hierarchy level:

```
edit interfaces at-fpc/pic/port atm-options]
scheduler-maps map-name {
    forwarding-class class-name {
        epd-threshold cells plp1 cells;
        linear-red-profile profile-name;
        priority (high | low);
        transmit-weight (cells number | percent number);
    }
```

```
    vc-cos-mode (alternate | strict);
}
```

You can define the following options for each forwarding class:

- **epd-threshold**—An EPD threshold provides a queue of cells that can be stored with tail drop. When a beginning-of-packet (BOP) cell is received, the VC's queue depth is checked against the EPD threshold. If the VC's queue depth exceeds the EPD threshold, the BOP cell and all subsequent cells in the packet are discarded.
- **linear-red-profile**—A linear RED profile defines the number of cells using the **queue-depth** statement within the RED profile. (You configure the **queue-depth** statement at the **[edit interfaces at-*fpc/pic/port* atm-options linear-red-profile *profile-name*]** hierarchy level.)

By default, if you include the **scheduler-maps** statement at the **[edit interfaces at-*fpc/pic/port* atm-options]** hierarchy level, the interface uses an EPD threshold that is determined by the Junos OS based on the available bandwidth and other parameters. You can override the default EPD threshold by setting an EPD threshold or a linear RED profile.

If shaping is enabled, the default EPD threshold is proportional to the shaping rate according to the following formula:

```
default epd-threshold = number of buffers * shaping rate / line rate
```

The minimum value is 48 cells. If the formula results in an EPD threshold less than 48 cells, the result is ignored, and the minimum value of 48 cells is used.

- priority—By default, queue 0 is high priority, and the remaining queues are low priority. You can configure high or low queuing priority for each queue.
- transmit-weight—By default, the transmit weight is 95 percent for queue 0, and 5 percent for queue 3. You can configure the transmission weight in number of cells or percentage. Each CoS queue is serviced in WRR mode. When CoS queues have data to send, they send the number of cells equal to their weight before passing control to the next active CoS queue. This allows proportional bandwidth sharing between multiple CoS queues within a rate-shaped VC tunnel. A CoS queue can send from 1 through 32,000 cells or from 5 through 100 percent of queued traffic before passing control to the next active CoS queue within a VC tunnel.

The AAL5 protocol prohibits cells from being interleaved on a VC; therefore, a complete packet is always sent. If a CoS queue sends more cells than its assigned weight because of the packet boundary, the deficit is carried over to the next time the queue is scheduled to transmit. If the queue is empty after the cells are sent, the deficit is waived, and the queue's assigned weight is reset.

NOTE: If you include the **scheduler-maps** statement at the **[edit interfaces at-*fpc/pic/port* atm-options]** hierarchy level, the **epd-threshold** statement at the **[edit interfaces *interface-name* unit *logical-unit-number*]** or **[edit interfaces *interface-name* unit *logical-unit-number* address *address* family *family* multipoint-destination *address*]** hierarchy level has no effect because either the default EPD threshold, the EPD threshold setting in the forwarding class, or the linear RED profile takes effect instead.

Enabling Eight Queues on ATM Interfaces

By default, ATM2 IQ PICs on M320 and T Series routers and Circuit Emulation PICs using ATM on the M120 and M320 are restricted to a maximum of four egress queues per interface. You can enable eight egress queues by including the **max-queues-per-interface** statement at the **[edit chassis fpc *slot-number* pic *pic-number*]** hierarchy level:

```
[edit chassis fpc slot-number pic pic-number]
max-queues-per-interface (4 | 8);
```

The numerical value can be **4** or **8**.

If you include the **max-queues-per-interface** statement, all ports on the PIC use the configured maximum.

When you include the **max-queues-per-interface** statement and commit the configuration, all physical interfaces on the PIC are deleted and re-added. Also, the PIC is taken offline and then brought back online immediately. You do not need to manually take the PIC offline and online. You should change modes between four queues and eight queues only when there is no active traffic going to the PIC.

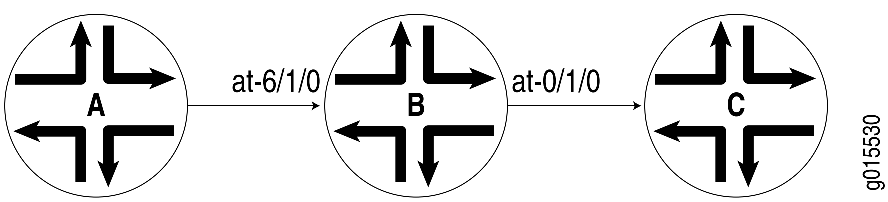
NOTE: When you are considering enabling eight queues on an ATM2 IQ interface, you should note the following:

- ATM2 IQ interfaces using Layer 2 circuit trunk transport mode support only four CoS queues.
- ATM2 IQ interfaces with MLPPP encapsulation support only four CoS queues.
- You can configure only four RED profiles for the eight queues. Thus, queue 0 and queue 4 share a single RED profile, as do queue 1 and queue 5, queue 2 and queue 6, and queue 3 and queue 7. There is no restriction on EPD threshold per queue.
- The default chassis scheduler allocates resources for queue 0 through queue 3, with 25 percent of the bandwidth allocated to each queue. When you configure the chassis to use more than four queues, you must configure and apply a custom chassis scheduler to override the default. To apply a custom chassis scheduler, include the **scheduler-map-chassis** statement at the **[edit class-of-service interfaces at-*fpc/pic/****]** hierarchy level. For more information about configuring and applying a custom chassis scheduler, see “Applying Scheduler Maps to Packet Forwarding Component Queues” on page 194.

Example: Enabling Eight Queues on ATM2 IQ Interfaces

In Figure 30 on page 419, Router A generates IP packets with different IP precedence settings. Router B is an M320 router or a T Series router with two ATM2 IQ interfaces. On Router B, interface **at-6/1/0** receives traffic from Router A, while interface **at-0/1/0** sends traffic to Router C. This example shows the CoS configuration for Router B.

Figure 30: Example Topology for Router with Eight Queues



On Router B:

```
[edit chassis]
fpc 0 {
   pic 1 {
      max-queues-per-interface 8;
   }
}
fpc 6 {
   pic 1 {
      max-queues-per-interface 8;
   }
}
```



```
[edit interfaces]
at-0/1/0 {
   atm-options {
      linear-red-profiles {
         red_1 queue-depth 1k high-plp-threshold 50 low-plp-threshold 80;
         red_2 queue-depth 2k high-plp-threshold 40 low-plp-threshold 70;
         red_3 queue-depth 3k high-plp-threshold 30 low-plp-threshold 60;
         red_4 queue-depth 4k high-plp-threshold 20 low-plp-threshold 50;
      }
      scheduler-maps {
         sch_red {
            vc-cos-mode strict;
            forwarding-class fc_q0 {
               priority high;
               transmit-weight percent 5;
               linear-red-profile red_1;
            }
            forwarding-class fc_q1 {
               priority low;
               transmit-weight percent 10;
               linear-red-profile red_2;
            }
            forwarding-class fc_q2 {
               priority low;
               transmit-weight percent 15;
               linear-red-profile red_3;
            }
            forwarding-class fc_q3 {
               priority low;
               transmit-weight percent 20;
               linear-red-profile red_4;
            }
            forwarding-class fc_q4 {
               priority low;
               transmit-weight percent 5;
               linear-red-profile red_1;
            }
            forwarding-class fc_q5 {
               priority low;
               transmit-weight percent 10;
               linear-red-profile red_2;
            }
            forwarding-class fc_q6 {
               priority low;
               transmit-weight percent 15;
               linear-red-profile red_3;
            }
            forwarding-class fc_q7 {
               priority low;
               transmit-weight percent 20;
               linear-red-profile red_4;
            }
         }
         sch_epd {
            vc-cos-mode alternate;
```



```
            forwarding-class fc_q0 {
                priority high;
                transmit-weight percent 5;
                epd-threshold 1024;
            }
            forwarding-class fc_q1 {
                priority low;
                transmit-weight percent 10;
                epd-threshold 2048;
            }
            forwarding-class fc_q2 {
                priority low;
                transmit-weight percent 15;
                epd-threshold 3072;
            }
            forwarding-class fc_q3 {
                priority low;
                transmit-weight percent 20;
                epd-threshold 4096;
            }
            forwarding-class fc_q4 {
                priority low;
                transmit-weight percent 5;
                epd-threshold 2048;
            }
            forwarding-class fc_q5 {
                priority low;
                transmit-weight percent 10;
                epd-threshold 3072;
            }
            forwarding-class fc_q6 {
                priority low;
                transmit-weight percent 15;
                epd-threshold 4096;
            }
            forwarding-class fc_q7 {
                priority low;
                transmit-weight percent 20;
                epd-threshold 5120;
            }
        }
    }
}
atm-options {
    vpi 0;
}
unit 0 {
    vci 0.100;
    shaping {
        cbr 1920000;
    }
    atm-scheduler-map sch_red;
    family inet {
        address 172.16.0.1/24;
    }
}
```

```
  unit 1 {
    vci 0.101;
    shaping {
      vbr peak 1m sustained 384k burst 256;
    }
    atm-scheduler-map sch_epd;
    family inet {
      address 172.16.1.1/24;
    }
  }
}
at-6/1/0 {
  atm-options {
    vpi 0;
  }
  unit 0 {
    vci 0.100;
    family inet {
      address 10.10.0.1/24;
    }
  }
  unit 1 {
    vci 0.101;
    family inet {
      address 10.10.1.1/24;
    }
  }
}

[edit class-of-service]
classifiers {
  inet-precedence inet_classifier {
    forwarding-class fc_q0 {
      loss-priority low code-points 000;
    }
    forwarding-class fc_q1 {
      loss-priority low code-points 001;
    }
    forwarding-class fc_q2 {
      loss-priority low code-points 010;
    }
    forwarding-class fc_q3 {
      loss-priority low code-points 011;
    }
    forwarding-class fc_q4 {
      loss-priority low code-points 100;
    }
    forwarding-class fc_q5 {
      loss-priority low code-points 101;
    }
    forwarding-class fc_q6 {
      loss-priority low code-points 110;
    }
    forwarding-class fc_q7 {
      loss-priority low code-points 111;
    }
```



```
    }
    forwarding-classes {
        queue 0 fc_q0;
        queue 1 fc_q1;
        queue 2 fc_q2;
        queue 3 fc_q3;
        queue 4 fc_q4;
        queue 5 fc_q5;
        queue 6 fc_q6;
        queue 7 fc_q7;
    }
    interfaces {
        at-6/1/0 {
            unit * {
                classifiers {
                    inet-precedence inet_classifier;
                }
            }
        }
    }
}
[edit routing-options]
static {
    route 10.10.20.2/32 {
        next-hop at-0/1/0.0;
        retain;
        no-readvertise;
    }
    route 10.10.1.2/32 {
        next-hop at-0/1/0.1;
        retain;
        no-readvertise;
    }
}
```

Verifying the Configuration

To see the results of this configuration, you can issue the following operational mode commands:

- **show interfaces at-0/1/0 extensive**
- **show interfaces queue at-0/1/0**
- **show class-of-service forwarding-class**

Configuring VC CoS Mode on ATM Interfaces

VC CoS mode defines the CoS queue scheduling priority. By default, the VC CoS mode is alternate. When it is a queue's turn to transmit, the queue transmits up to its weight in cells as specified by the **transmit-weight** statement at the **[edit interfaces at-*fpc/pic/port* atm-options scheduler-maps *map-name* forwarding-class *class-name*]** hierarchy level. The number of cells transmitted can be slightly over the configured or default transmit weight, because the transmission always ends at a packet boundary.

To configure the VC CoS mode, include the **vc-cos-mode** statement at the **[edit interfaces at-*fpc/pic/port* atm-options scheduler-maps]** hierarchy level:

```
edit interfaces at-fpc/pic/port atm-options scheduler-maps]
vc-cos-mode (alternate | strict);
```

Two modes of CoS scheduling priority are supported:

- **alternate**—Assign **high** priority to one queue. The scheduling of the queues alternates between the **high** priority queue and the remaining queues. Every other scheduled packet is from the **high** priority queue.
- **strict**—Assign strictly **high** priority to one queue. A queue with strictly **high** priority is always scheduled before the remaining queues. The remaining queues are scheduled in round-robin fashion.

Copying the PLP Setting to the CLP Bit on ATM Interfaces

For a provider-edge (PE) router with customer edge (CE)-facing, egress, ATM2 IQ interfaces configured with standard AAL5 encapsulation, you can enable the PLP setting to be copied into the CLP bit.

NOTE: This configuration setting is not applicable to Layer 2 circuit encapsulations because the control word captures and preserves CLP information. For more information about Layer 2 circuit encapsulations, see the *Junos OS Network Interfaces Configuration Guide*.

By default, at egress ATM2 IQ interfaces configured with standard AAL5 encapsulation, the PLP information is not copied to the CLP bit. This means the PLP information is not carried beyond the egress interface onto the CE router.

You can enable the PLP information to be copied into the CLP bit by including the **plp-to-clp** statement:

```
plp-to-clp;
```

You can include this statement at the following hierarchy levels:

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

Applying Scheduler Maps to Logical ATM Interfaces

To apply the ATM scheduler map to a logical interface, include the **atm-scheduler-map** statement:

```
atm-scheduler-map (map-name | default);
```


When you add or change a scheduler map, the associated logical interface is taken offline and then brought back online immediately. For ATM CoS to take effect, you must configure the VCI and VPI identifiers and traffic shaping on each VC by including the following statements:

```
vci vpi-identifier.vci-identifier;
shaping {
  (cbr rate | rtvbr peak rate sustained rate burst length  | vbr peak rate sustained rate burst
    length);
}
```

You can include these statements at the following hierarchy levels:

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

For more information, see the *Junos OS Network Interfaces Configuration Guide*.

You can also apply a scheduler map to the chassis traffic that feeds the ATM interfaces. For more information, see "Applying Scheduler Maps to Packet Forwarding Component Queues" on page 194.

Example: Configuring CoS for ATM2 IQ VC Tunnels

This example configures ATM2 IQ VC tunnel CoS components:

```
[edit interfaces]
at-1/2/0 {
  atm-options {
    vpi 0;
    linear-red-profiles red-profile-1 {
      queue-depth 35000 high-plp-threshold 75 low-plp-threshold 25;
    }
    scheduler-maps map-1 {
      vc-cos-mode strict;
      forwarding-class best-effort {
        priority low;
        transmit-weight percent 25;
        linear-red-profile red-profile-1;
      }
    }
  }
  unit 0 {
    vci 0.128;
    shaping {
      vbr peak 20m sustained 10m burst 20;
    }
    atm-scheduler-map map-1;
    family inet {
      address 192.168.0.100/32 {
        destination 192.168.0.101;
      }
    }
```



```
  }
}
```

Configuring CoS for L2TP Tunnels on ATM Interfaces

The Layer 2 Tunneling Protocol (L2TP) is often used to carry traffic securely between an L2TP Network Server (LNS) to an L2TP Access Concentrator (LAC). CoS is supported for L2TP session traffic to a LAC on platforms configured as an LNS that include egress IQ2 PICs. Supported routers are:

- M7i and M10i routers
- M120 routers

To enable session-aware CoS on an L2TP interface, include the **per-session-scheduler** statement at the **[edit interfaces unit *logical-unit-number*]** hierarchy level.

```
[edit interfaces interface-name unit logical-unit-number]
per-session-scheduler;
```

You also must set the IQ2 PIC mode for session-aware traffic shaping and set the number of bytes to add to or subtract from the packet before ATM cells are created. To configure these options on the ingress side of the tunnel, include the **ingress-shaping-overhead** and **mode** statements at the **[edit chassis fpc *slot-number* pic *pic-number* traffic-manager]** hierarchy level.

```
[edit chassis fpc slot-number pic pic-number]
traffic-manager {
  ingress-shaping-overhead number;
  mode session-shaping;
}
```

Various limitations apply to this feature:

- Only 991 shapers are supported on each IQ2 PIC.
- Sessions in excess of 991 cannot be shaped (but they can be policed).
- There is no support for PPP multilinks.
- The overall traffic rate cannot exceed the L2TP traffic rate, or else random drops result.
- There is no support for logical interface scheduling and shaping at the ingress because all schedulers are now reserved for L2TP.
- There is no support for physical interface rate shaping at the ingress.

You can provide policing support for sessions with more than the 991 shapers on each IQ2 PIC. Each session can have four or eight different classes of traffic (queues). Each class needs its own policer, for example, one for voice and one for data traffic. The policer is configured within a **simple-filter** statement and only **forwarding class** is supported in the **from** clause. Only one policer can be referenced in each simple filter.

The following example shows a policer within a simple filter applied to two assured forwarding classes:


```
[edit firewall]
policer P1 {
   if-exceeding {
      bandwidth-limit 400k;
         burst-size-limit 1500;
      }
   then discard;
}
family inet {
   simple-filter SF-1 {
      term T-1 {
         from {
            forwarding-class [ af11 af21 ];
         }
         then policer P1;
      }
   }
}
```

You can also set the number of bytes to add to or subtract from the packet at the egress of the tunnel. To configure these options on the egress side of the tunnel, include the **egress-shaping-overhead** and **mode** statements at the **[edit chassis fpc *slot-number* pic *pic-number* traffic-manager]** hierarchy level.

```
[edit chassis fpc slot-number pic pic-number]
traffic-manager {
   egress-shaping-overhead number;
   mode session-shaping;
}
```

For MX series routers, when ingress queueing is enabled on EQ DPCs, ingress shaping overhead can be made accurate by using the following values for the **ingress-shaping-overhead** statement:

- For Layer 2, subtract 14 bytes (-14)
- For Layer 3 untagged ports, add 2 bytes
- For Layer 3 dual-tagged ports, add 10 bytes

NOTE: This is a packet forwarding engine setting, so if there is a mix of Layer 2 and Layer 3 traffic, the shaping will be accurate for one of the traffic types, but not the other.

Configuring IEEE 802.1p BA Classifiers for Ethernet VPLS Over ATM

You can apply an IEEE 802.1p behavior aggregate (BA) classifier to VPLS in a bridged Ethernet over ATM environment using ATM (RFC 1483) encapsulation. This extracts the Layer 2 (frame level) IEEE 802.1p information from the cells arriving on the ATM interface. Note that the interface must be configured for the Ethernet VPLS service over ATM links.

This example applies the classifier **atm-ether-vpls-classifer** to an ATM interface using **ether-vpls-over-atm-llc** encapsulation. This is not a complete CoS configuration example.

```
[edit class-of-service interfaces]
at-1/2/3 {
  unit 0 {
    (...) # Other CoS features
    classifiers {
      ieee-802.1 atm-ether-vpls-classifier; # Classifier defined elsewhere
    }
  }
}

[edit]
interface at-1/2/3 {
  atm-options {
    vpi 0;
  }
  unit 0 {
    encapsulation ether-vpls-over-atm-llc; # Required encapsulation type
    vci 0.100;
    family vpls;
  }
}
```

You must configure a routing instance for the VPLS as well:

```
[edit routing-instances]
cos-test-1 {
  instance-type vpls; #This is required
  interface at-1/2/3;
  route-distinguisher 10.10.10.10:1;
  vrf-target target:11111:1;
  protocols {
    vpls {
      site-range 10;
      site cos-test-v1-site1 {
        site-identifier 1;
      }
    }
  }
}
```

The Layer 2 VPN classification on an ATM interface is limited to the Layer 2 granularity, not to each separate VLAN/VPLS instance. In other words, all of the VLAN/VPLS packets arriving on an ATM virtual circuit are classified by a single IEEE 802.1p classifier. The individual flow of each VLAN cannot be identified at this level.

Example: Combine Layer 2 and Layer 3 Classification on the Same ATM Physical Interface

With the ATM II IQ PIC installed on the M320 router with the Enhanced Type 3 FPC or the M120 router, you can combine Layer 2 and Layer 3 classifiers on the same ATM physical interface. However, you must apply the classifiers to different logical interfaces (units). The Layer 3 interface can belong to a Layer 3 VPN or VPLS routing instance and the Layer

2 interface can belong to a VPLS routing instance. If the Layer 3 interface belongs to a VPLS routing instance, only IPv4 DSCP or Internet precedence classification is supported. When the ATM interface is part of a Layer 3 VPN, both IPv4 and IPv6 DSCP or Internet precedence classification is supported.

This example applies a Layer 3 DSCP classifier named **dscp-1** and a Layer 2 IEEE 802.1 classifier named **ieee-1** to ATM interface **at-4/1/1** units 0 and 1. The **inet-precedence** Layer 3 classification is also supported but is not used in this example.

```
[edit]
class-of-service {
    interfaces {
        at-4/1/1 {
            unit 0 {
                classifiers {
                    dscp dscp_1;
                }
                unit 1 {
                    classifiers {
                        ieee-802.1 ieee;
                    }
                }
            }
        }
```

Related Documentation

- BA Classifier Overview on page 39

CHAPTER 25

Configuring CoS for MPLS

This topic discusses the following:

CoS for MPLS Overview

When IP traffic enters a label-switched path (LSP) tunnel, the ingress router marks all packets with a class-of-service (CoS) value, which is used to place the traffic into a transmission priority queue. On the router, each interface has up to eight transmit queues. The CoS value is encoded as part of the Multiprotocol Label Switching (MPLS) header and remains in the packets until the MPLS header is removed when the packets exit from the egress router. The routers within the LSP utilize the CoS value set at the ingress router. The CoS value is encoded by means of the CoS bits (also known as the EXP or experimental bits).

MPLS class of service works in conjunction with the router's general CoS functionality. If you do not configure any CoS features, the default general CoS settings are used. For MPLS class of service, you might want to prioritize how the transmit queues are serviced by configuring weighted round-robin, and to configure congestion avoidance using random early detection (RED).

The next-hop label-switching router (LSR) uses the default classification shown in Table 116 on page 431.

Table 116: LSR Default Classification

Code Point	Forwarding Class	Loss Priority
000	best-effort	low
001	best-effort	high
010	expedited-forwarding	low
011	expedited-forwarding	high
100	assured-forwarding	low

Table 116: LSR Default Classification *(continued)*

Code Point	Forwarding Class	Loss Priority
101	assured-forwarding	high
110	network-control	low
111	network-control	high

Configuring CoS for MPLS Traffic

To configure CoS for MPLS packets in an LSP, include the **class-of-service** statement with the appropriate CoS value:

class-of-service *cos-value*;

If you do not specify a CoS value, the IP precedence bits from the packet's IP header are used as the packet's CoS value.

You can include this statement at the following hierarchy levels:

- [edit protocols mpls]
- [edit protocols mpls interface *interface-name* label-map *label-value*]
- [edit protocols mpls label-switched-path *path-name*]
- [edit protocols mpls label-switched-path *path-name* primary *path-name*]
- [edit protocols mpls label-switched-path *path-name* secondary *path-name*]
- [edit protocols mpls static-path *prefix*]
- [edit protocols rsvp interface *interface-name* link-protection]
- [edit protocols rsvp interface *interface-name* link-protection bypass *destination*]
- [edit logical-systems *logical-system-name* protocols mpls]
- [edit logical-systems *logical-system-name* protocols mpls label-switched-path *path-name*]
- [edit logical-systems *logical-system-name* protocols mpls label-switched-path *path-name* primary *path-name*]
- [edit logical-systems *logical-system-name* protocols mpls label-switched-path *path-name* secondary *path-name*]
- [edit logical-systems *logical-system-name* protocols mpls static-path *prefix*]
- [edit logical-systems *logical-system-name* protocols mpls interface *interface-name* label-map *label-value*]

- **[edit logical-systems *logical-system-name* protocols rsvp interface *interface-name* link-protection]**
- **[edit logical-systems *logical-system-name* protocols rsvp interface *interface-name* link-protection bypass *destination*]**

The **class-of-service** statement at the **[edit protocols mpls label-switched-path]** hierarchy level assigns an initial EXP value for the MPLS shim header of packets in the LSP. This value is initialized at the ingress router only and overrides the rewrite configuration established for that forwarding class. However, the CoS processing (weighted round robin [WRR] and RED) of packets entering the ingress router is not changed by the **class-of-service** statement on an MPLS LSP. Classification is still based on the behavior aggregate (BA) classifier at the **[edit class-of-service]** hierarchy level or the multifield classifier at the **[edit firewall]** hierarchy level.

We recommend configuring all routers along the LSP to have the same input classifier for EXP, and, if a rewrite rule is configured, all routers should have the same rewrite configuration. Otherwise, traffic at the next LSR might be classified into a different forwarding class, resulting in a different EXP value being written to the EXP header.

For more information, see the *Junos OS MPLS Applications Configuration Guide*.

PART 5

CoS Configuration Examples and Statements

CHAPTER 26

CoS Configuration Examples

This topic discusses the following:

Example: Configuring Classifiers, Rewrite Markers, and Schedulers

1. Define a classifier that matches IP traffic arriving on the interface. The affected IP traffic has IP precedence bits with patterns matching those defined by aliases A or B. The loss priority of the matching packets is set to low, and the forwarding class is mapped to best effort (queue 0):

```
[edit]
class-of-service {
   classifiers {
      inet-precedence normal-traffic {
         forwarding-class best-effort {
            loss-priority low code-points [my1 my2];
         }
      }
   }
}
```

   Following are the code-point alias and forwarding-class mappings referenced in the **normal-traffic** classifier:

```
[edit]
class-of-service {
   code-point-aliases {
      inet-precedence {
         my1 000;
         my2 001;
         ...
      }
   }
}

[edit]
class-of-service {
   forwarding-classes {
      queue 0 best-effort;
```

```
            queue 1 expedited-forwarding;
        }
    }
```

2. Use rewrite markers to redefine the bit pattern of outgoing packets. Assign the new bit pattern based on specified forwarding classes, regardless of the loss priority of the packets:

```
[edit]
class-of-service {
    rewrite-rules {
        inet-precedence clear-prec {
            forwarding-class best-effort {
                loss-priority low code-point 000;
                loss-priority high code-point 000;
            }
            forwarding-class expedited-forwarding {
                loss-priority low code-point 100;
                loss-priority high code-point 100;
            }
        }
    }
}
```

3. Configure a scheduler map associating forwarding classes with schedulers and drop-profiles:

```
[edit]
class-of-service {
    scheduler-maps {
        one {
            forwarding-class expedited-forwarding scheduler special;
            forwarding-class best-effort scheduler normal;
        }
    }
}
```

Schedulers establish how to handle the traffic within the output queue for transmission onto the wire. Following is the scheduler referenced in scheduler map **one**:

```
[edit]
class-of-service {
    schedulers {
        special {
            transmit-rate percent 30;
            priority high;
        }
        normal {
            transmit-rate percent 70;
            priority low;
        }
    }
}
```

4. Apply the **normal-traffic** classifier to all SONET/SDH interfaces and all logical interfaces of SONET/SDH interfaces; apply the **clear-prec** rewrite marker to all Gigabit Ethernet

interfaces and all logical interfaces of Gigabit Ethernet interfaces; and apply the **one** scheduler map to all interfaces:

```
[edit]
class-of-service {
   interfaces {
      so-0/0/0 {
         scheduler-map one;
         unit 0 {
            classifiers {
               inet-precedence normal-traffic;
            }
         }
      }
      so-0/0/1 {
         scheduler-map one;
         unit 1 {
            classifiers {
               inet-precedence normal-traffic;
            }
         }
      }
      ge-1/0/0 {
         scheduler-map one;
         unit 0 {
            rewrite-rules {
               inet-precedence clear-prec;
            }
         }
         unit 1 {
            rewrite-rules {
               inet-precedence clear-prec;
            }
         }
      }
      ge-1/0/1 {
         scheduler-map one;
         unit 0 {
            rewrite-rules {
               inet-precedence clear-prec;
            }
         }
         unit 1 {
            rewrite-rules {
               inet-precedence clear-prec;
            }
         }
      }
   }
}
```

Following is the complete configuration:

```
[edit class-of-service]
classifiers {
   inet-precedence normal-traffic {
      forwarding-class best-effort {
```

```
            loss-priority low code-points [my1 my2];
        }
    }
}
code-point-aliases {
    inet-precedence {
        my1 000;
        my2 001;
        cs1 010;
        cs2 011;
        cs3 100;
        cs4 101;
        cs5 110;
        cs6 111;
    }
}
drop-profiles {
    high-priority {
        fill-level 20 drop-probability 100;
    }
    low-priority {
        fill-level 90 drop-probability 95;
    }
    big-queue {
        fill-level 100 drop-probability 100;
    }
}
forwarding-classes {
    queue 0 best-effort;
    queue 1 expedited-forwarding;
}
interfaces {
    so-0/0/0 {
        scheduler-map one;
        unit 0 {
            classifiers {
                inet-precedence normal-traffic;
            }
        }
    }
    so-0/0/1 {
        scheduler-map one;
        unit 1 {
            classifiers {
                inet-precedence normal-traffic;
            }
        }
    }
    ge-1/0/0 {
        scheduler-map one;
        unit 0 {
            rewrite-rules {
                inet-precedence clear-prec;
            }
        }
        unit 1 {
```



```
            rewrite-rules {
                inet-precedence clear-prec;
            }
        }
    }
    ge-1/0/1 {
        scheduler-map one;
        unit 0 {
            rewrite-rules {
                inet-precedence clear-prec;
            }
        }
        unit 1 {
            rewrite-rules {
                inet-precedence clear-prec;
            }
        }
    }
}
rewrite-rules {
    inet-precedence clear-prec {
        forwarding-class best-effort {
            loss-priority low code-point 000;
            loss-priority high code-point 000;
        }
        forwarding-class expedited-forwarding {
            loss-priority low code-point 100;
            loss-priority high code-point 100;
        }
    }
}
scheduler-maps {
    one {
        forwarding-class expedited-forwarding scheduler special;
        forwarding-class best-effort scheduler normal;
    }
}
schedulers {
    special {
        transmit-rate percent 30;
        priority high;
    }
    normal {
        transmit-rate percent 70;
        priority low;
    }
}
```

Example: Configuring a CoS Policy for IPv6 Packets

1. Define a new classifier of type DSCP IPv6.

```
[edit class-of-service]
classifiers {
  dscp-ipv6 core-dscp-map {
    forwarding-class best-effort {
      loss-priority low code-points 000000;
    }
    forwarding-class assured-forwarding {
      loss-priority low code-points 001010;
    }
    forwarding-class network-control {
      loss-priority low code-points 110000;
    }
  }
}
```

2. Define a new rewrite rule of type DSCP IPv6.

```
[edit class-of-service]
rewrite-rules {
  dscp-ipv6 core-dscp-rewrite {
    forwarding-class best-effort {
      loss-priority low code-point 000000;
    }
    forwarding-class assured-forwarding {
      loss-priority low code-point 001010;
    }
    forwarding-class network-control {
      loss-priority low code-point 110000;
    }
  }
}
```

3. Assign the classifier and rewrite rule to a logical interface.

```
[edit class-of-service]
interfaces {
  so-2/0/0 {
    unit 0 {
      classifiers { # Both dscp and dscp-ipv6 classifiers on this interface.
        dscp default;
        dscp-ipv6 core-dscp-map;
      }
      rewrite-rules { # Both dscp and dscp-ipv6 rewrite rules on this interface.
        dscp default;
        dscp-ipv6 core-dscp-rewrite;
      }
    }
  }
}
```


CHAPTER 27

Summary of CoS Configuration Statements

The following sections explain each of the class-of-service (CoS) configuration statements. The statements are organized alphabetically.

action

Syntax

```
action {
   loss-priority high then discard;
}
```

Hierarchy Level [edit firewall **three-color-policer** *policer-name*]

Release Information Statement introduced in Junos OS Release 8.2.

Description This statement discards high loss priority traffic as part of a configuration using tricolor marking on a logical interface on an IQ2 PIC.

Required Privilege Level interface—To view this statement in the configuration.
interface-control—To add this statement to the configuration.

Related Documentation

- Configuring Tricolor Marking Policers on page 98
- **logical-interface-policer on page 516**

address

Syntax

```
address address {
  destination address;
}
```

Hierarchy Level

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

Release Information Statement introduced before Junos OS Release 7.4.

Description For CoS on ATM interfaces, configure the interface address.

Options ***address***—Address of the interface.

The remaining statements are explained separately.

Required Privilege Level

interface—To view this statement in the configuration.
interface-control—To add this statement to the configuration.

Related Documentation

- Example: Configuring CoS for ATM2 IQ VC Tunnels on page 425
- *Junos OS Network Interfaces Configuration Guide*

application-profile

Syntax

```
application-profile profile-name;
application-profile profile-name {
  ftp {
    data {
      dscp (alias | bits);
      forwarding-class class-name;
    }
  }
  sip {
    video {
      dscp (alias | bits);
      forwarding-class class-name;
    }
    voice {
      dscp (alias | bits);
      forwarding-class class-name;
    }
  }
}
```

Hierarchy Level

[edit **services cos**],
[edit services cos rule *rule-name* term *term-name* **then**],
[edit **services** cos rule *rule-name* term *term-name* then **(reflexive | reverse)**]

Release Information Statement introduced in Junos OS Release 8.1.

Description Define or apply a CoS application profile. When you apply a CoS application profile in a CoS rule, terminate the profile name with a semicolon (;).

Options ***profile-name***—Identifier for the application profile.

The remaining statements are explained separately.

Required Privilege Level

interface—To view this statement in the configuration.
interface-control—To add this statement to the configuration.

Related Documentation

- Configuring Application Profiles on page 285

application-sets

Syntax applications-sets [*set-name*];

Hierarchy Level [edit **services** cos rule *rule-name* term *term-name* **from**]

Release Information Statement introduced in Junos OS Release 8.1.

Description Define one or more target application sets.

Options ***set-name***—Name of the target application set.

Required Privilege Level interface—To view this statement in the configuration.
interface-control—To add this statement to the configuration.

Related Documentation
- Configuring Match Conditions in a CoS Rule on page 284

applications

Syntax applications [*application-name*];

Hierarchy Level [edit **services** cos rule *rule-name* term *term-name* **from**]

Release Information Statement introduced in Junos OS Release 8.1.

Description Define one or more applications to which the CoS services apply.

Options ***application-name***—Name of the target application.

Required Privilege Level interface—To view this statement in the configuration.
interface-control—To add this statement to the configuration.

Related Documentation
- Configuring Match Conditions in a CoS Rule on page 284

atm-options

Syntax

```
atm-options {
  linear-red-profiles profile-name {
    high-plp-max-threshold percent;
    low-plp-max-threshold percent;
    queue-depth cells high-plp-threshold percent low-plp-threshold percent;
  }
  plp-to-clp;
  scheduler-maps map-name {
    forwarding-class class-name {
      epd-threshold cells plp1 cells;
      linear-red-profile profile-name;
      priority (high | low);
      transmit-weight (cells number | percent number);
    }
    vc-cos-mode (alternate | strict);
  }
}
```

Hierarchy Level [edit interfaces *interface-name*]

Release Information Statement introduced before Junos OS Release 7.4.

Description Configure ATM-specific physical interface properties.

The statements are explained separately.

Required Privilege Level interface—To view this statement in the configuration.
interface-control—To add this statement to the configuration.

Related Documentation

- Configuring Linear RED Profiles on ATM Interfaces on page 412
- **shaping on page 559**
- **vci on page 589**

atm-scheduler-map

Syntax atm-scheduler-map (*map-name* | default);

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

Release Information Statement introduced before Junos OS Release 7.4.

Description Associate a scheduler map with a virtual circuit on a logical interface.

Options *map-name*—Name of scheduler map that you define at the **[edit interfaces *interface-name* scheduler-maps]** hierarchy level.

default—The default scheduler mapping.

Required Privilege Level interface—To view this statement in the configuration.
interface-control—To add this statement to the configuration.

Related Documentation

- Applying Scheduler Maps to Logical ATM Interfaces on page 424
- **scheduler-maps on page 556**

buffer-size

Syntax buffer-size (percent *percentage* | remainder | temporal *microseconds*);

Hierarchy Level [edit class-of-service **schedulers** *scheduler-name*]

Release Information Statement introduced before Junos OS Release 7.4.

Description Specify buffer size.

Default If you do not include this statement, 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.

Options **percent *percentage***—Buffer size as a percentage of total buffer.

remainder—Remaining buffer available.

temporal *microseconds*—Buffer size as a temporal value. The queuing algorithm starts dropping packets when it queues more than a computed number of bytes. This maximum is computed by multiplying the logical interface speed by the configured temporal value.

Range: The ranges vary by platform as follows:

- For M320 and T Series routers with Type 1 and Type 2 FPCs: 1 through 80,000 microseconds.
- For M320 and T Series routers with Type 3 FPCs: 1 through 50,000 microseconds.
- For M7i, M10i, M5, and M10 routers: 1 through 100,000 microseconds.
- For other M Series routers: 1 through 200,000 microseconds.
- For IQ PICs on M320 and T Series routers: 1 through 50,000 microseconds.
- For IQ PICs on other M Series routers: 1 through 100,000 microseconds.

Required Privilege Level interface—To view this statement in the configuration.
interface-control—To add this statement to the configuration.

Related Documentation

- Configuring the Scheduler Buffer Size on page 148
- Example: Configuring CoS for a PBB Network on MX Series Routers

cbr

Syntax cbr *rate*;

Hierarchy Level [edit interfaces at-*fpc/pic/port* atm-options vpi *vpi-identifier* **shaping**],
[edit interfaces at-*fpc/pic/port* unit *logical-unit-number* address *address* family *family* multipoint-destination *address* **shaping**],
[edit interfaces at-*fpc/pic/port* unit *logical-unit-number* **shaping**],
[edit logical-systems *logical-system-name* interfaces at-*fpc/pic/port* unit *logical-unit-number* address *address* family *family* multipoint-destination *address* **shaping**],
[edit logical-systems *logical-system-name* interfaces at-*fpc/pic/port* **unit** *logical-unit-number* **shaping**]

Release Information Statement introduced before Junos OS Release 7.4.

Description For ATM encapsulation only, define a constant bit rate bandwidth utilization in the traffic-shaping profile.

Default Unspecified bit rate (UBR); that is, bandwidth utilization is unlimited.

Options ***rate***—Peak rate, in bits per second (bps) or cells per second (cps). You can specify a value in bits per second either as a complete decimal number or as a decimal number followed by the abbreviation **k** (1000), **m** (1,000,000), or **g** (1,000,000,000). You can also specify a value in cells per second by entering a decimal number followed by the abbreviation **c**; values expressed in cells per second are converted to bits per second by means of the formula 1 cps = 384 bps.

For ATM1 OC3 interfaces, the maximum available rate is 100 percent of ***line-rate***, or 135,600,000 bps. For ATM1 OC12 interfaces, the maximum available rate is 50 percent of ***line-rate***, or 271,263,396 bps. For ATM2 IQ interfaces, the maximum available rate is 542,526,792 bps.

Required Privilege Level interface—To view this statement in the configuration.
interface-control—To add this statement to the configuration.

Related Documentation

- Applying Scheduler Maps to Logical ATM Interfaces on page 424
- **rtvbr on page 549**
- **shaping on page 559**
- **vbr on page 587**

class

See the following sections:

- **class (CoS-Based Forwarding) on page 451**
- **class (Forwarding Classes) on page 452**

class (CoS-Based Forwarding)

Syntax

```
class class-name {
   classification-override {
      forwarding-class class-name;
   }
}
```

Hierarchy Level [edit class-of-service forwarding-policy]

Release Information Statement introduced before Junos OS Release 7.4.

Description Configure CoS-based forwarding class.

Options *class-name*—Name of the routing policy class.

The remaining statements are explained separately.

Usage Guidelines See "Overriding the Input Classification" on page 132.

Required Privilege Level interface—To view this statement in the configuration.
interface-control—To add this statement to the configuration.

class (Forwarding Classes)

Syntax class *class-name* queue-num *queue-number* **priority** (high | low);

Hierarchy Level [edit class-of-service **forwarding-classes**]

Release Information Statement introduced in Junos OS Release 8.1.

Description On M120 , M320, MX Series routers, and T Series routers only, specify the output transmission queue to which to map all input from an associated forwarding class.

This statement enables you to configure up to 16 forwarding classes with multiple forwarding classes mapped to single queues. If you want to configure up to eight forwarding classes with one-to-one mapping to output queues, use the **queue** statement instead of the **class** statement at the **[edit class-of-service forwarding-classes]** hierarchy level.

Options *class-name*—Name of forwarding class.

queue-number—Output queue number.

Range: 0 through 15. Some T Series router PICs are restricted to 0 through 3.

The remaining statement is explained separately.

Usage Guidelines See "Configuring Forwarding Classes" on page 115.

Required Privilege Level interface—To view this statement in the configuration.
interface-control—To add this statement to the configuration.

Related Documentation

- **queue (Global Queues) on page 542**

class-of-service

Syntax class-of-service { ... }

Hierarchy Level [edit]

Release Information Statement introduced before Junos OS Release 7.4.

Description Configure Junos CoS features.

Default If you do not configure any CoS features, all packets are transmitted from output transmission queue 0.

Usage Guidelines See "CoS Overview" on page 3.

Required Privilege Level interface—To view this statement in the configuration.
interface-control—To add this statement to the configuration.

classification-override

Syntax

```
classification-override {
   forwarding-class class-name;
}
```

Hierarchy Level [edit class-of-service forwarding-policy **class** *class-name*]

Release Information Statement introduced before Junos OS Release 7.4.

Description For IPv4 packets, override the incoming packet classification, assigning all packets sent to a destination prefix to the same output transmission queue.

Usage Guidelines See “Overriding the Input Classification” on page 132.

Required Privilege Level interface—To view this statement in the configuration.
interface-control—To add this statement to the configuration.

Related Documentation

- **policy-statement** in the *Junos OS Routing Protocols Configuration Guide*

classifiers

See the following sections:

- **classifiers (Application) on page 454**
- **classifiers (Application for Routing Instances) on page 455**
- **classifiers (Definition) on page 456**

classifiers (Application)

Syntax

```
classifiers {
  type (classifier-name | default) family (mpls | inet);
}
```

Hierarchy Level [edit class-of-service interfaces *interface-name* unit *logical-unit-number*]

Release Information Statement introduced before Junos OS Release 7.4.

Description Apply a CoS aggregate behavior classifier to a logical interface. You can apply a default classifier or one that is previously defined.

Options ***classifier-name***—Name of the aggregate behavior classifier.

type—Traffic type.

Values: dscp, **dscp-ipv6**, **exp**, **ieee-802.1**, **inet-precedence**

NOTE: You can only specify a family for the **dscp** and **dscp-ipv6** types.

Usage Guidelines See "Applying Classifiers to Logical Interfaces" on page 50.

Required Privilege Level interface—To view this statement in the configuration.
interface-control—To add this statement to the configuration.

classifiers (Application for Routing Instances)

Syntax

```
classifiers {
  exp (classifier-name | default);
  dscp (classifier-name | default);
  dscp-ipv6 (classifier-name | default);
}
```

Hierarchy Level [edit class-of-service **routing-instances** *routing-instance-name*]

Release Information Statement introduced before Junos OS Release 7.4.
dscp and **dscp-ipv6** support introduced in Junos OS Release 9.6.

Description For routing instances with VRF table labels enabled, apply a custom Multiprotocol Label Switching (MPLS) EXP classifier or DSCP classifier to the routing instance. You can apply the default classifier or one that is previously defined.

Options ***classifier-name***—Name of the behavior aggregate MPLS EXP or DSCP classifier.

Usage Guidelines See “Applying MPLS EXP Classifiers to Routing Instances” on page 56 and “Applying Classifiers to Logical Interfaces” on page 50.

Required Privilege Level interface—To view this statement in the configuration.
interface-control—To add this statement to the configuration.

classifiers (Definition)

Syntax

```
classifiers {
  type classifier-name {
    import (classifier-name | default);
    forwarding-class class-name {
      loss-priority level code-points [ aliases ] [ bit-patterns ];
    }
  }
}
```

Hierarchy Level [edit class-of-service],
[edit class-of-service **routing-instances** *routing-instance-name*]

Release Information Statement introduced before Junos OS Release 7.4.
ieee-802.1ad option introduced in Junos OS Release 9.2.

Description Define a CoS aggregate behavior classifier for classifying packets. You can associate the classifier with a forwarding class or code-point mapping, and import a default classifier or one that is previously defined.

Options ***classifier-name***—Name of the aggregate behavior classifier.

type—Traffic type: **dscp**, **dscp-ipv6**, **exp**, **ieee-802.1**, **ieee-802.1ad**, **inet-precedence**.

Required Privilege Level interface—To view this statement in the configuration.
interface-control—To add this statement to the configuration.

Related Documentation

- Overview of BA Classifier Types on page 42
- Example: Configuring CoS for a PBB Network on MX Series Routers

code-point

Syntax code-point [*aliases*] [*bit-patterns*];

Hierarchy Level [edit class-of-service rewrite-rules *type rewrite-name* **forwarding-class** *class-name*]

Release Information Statement introduced before Junos OS Release 7.4.

Description Specify one or more code-point aliases or bit sets for association with a forwarding class.

Options ***aliases***—Name of each alias.

bit-patterns—Value of the code-point bits, in decimal form.

Required Privilege Level interface—To view this statement in the configuration.
interface-control—To add this statement to the configuration.

Related Documentation

- Configuring Rewrite Rules on page 242

code-point-aliases

Syntax

```
code-point-aliases {
  type {
    alias-name bits;
  }
}
```

Hierarchy Level [edit class-of-service]

Release Information Statement introduced before Junos OS Release 7.4.

Description Define an alias for a CoS marker.

Options ***alias-name***—Name of the code-point alias.

bits—6-bit value of the code-point bits, in decimal form.

type—CoS marker type.

Values: dscp, **dscp-ipv6**, **exp**, **ieee-802.1**, **inet-precedence**

Usage Guidelines See “Defining Code Point Aliases for Bit Patterns” on page 68.

Required Privilege Level interface—To view this statement in the configuration.
interface-control—To add this statement to the configuration.

code-points

Syntax code-points ([*aliases*] | [*bit-patterns*]);

Hierarchy Level [edit class-of-service classifiers *type classifier-name* forwarding-class *class-name*]

Release Information Statement introduced before Junos OS Release 7.4.

Description Specify one or more DSCP code-point aliases or bit sets for association with a forwarding class.

Options ***aliases***—Name of the DSCP alias.

bit-patterns—Value of the code-point bits, in six-bit binary form.

Required Privilege Level interface—To view this statement in the configuration.
interface-control—To add this statement to the configuration.

Related Documentation

- Overview of BA Classifier Types on page 42
- Example: Configuring CoS for a PBB Network on MX Series Routers

copy-tos-to-outer-ip-header

Syntax copy-tos-to-outer-ip-header;

Hierarchy Level [edit interfaces at-*fpc/pic/port* unit *logical-unit-number*],
[edit logical-systems *logical-system-name* interfaces at-*fpc/pic/port* unit *logical-unit-number*]

Release Information Statement introduced in Junos OS Release 8.2.

Description For GRE tunnel interfaces only, enables the inner IP header's ToS bits to be copied to the outer IP packet header.

Default If you omit this statement, the ToS bits in the outer IP header are set to 0.

Required Privilege Level interface—To view this statement in the configuration.
interface-control—To add this statement to the configuration.

Related Documentation

- Example: Configuring a GRE Tunnel to Copy ToS Bits to the Outer IP Header on page 279

data

Syntax

```
data {
  dscp (alias | bits);
  forwarding-class class-name;
}
```

Hierarchy Level [edit services cos **application-profile** *profile-name* ftp]

Release Information Statement introduced in Junos OS Release 9.3.

Description Set the appropriate **dscp** and **forwarding-class** value for FTP data.

Default By default, the system will not alter the DSCP or forwarding class for FTP data traffic.

Required Privilege Level interface—To view this statement in the configuration.
interface-control—To add this statement to the configuration.

Related Documentation

- Configuring Application Profiles on page 285
- **video on page 590**
- **voice on page 591**

delay-buffer-rate

Syntax delay-buffer-rate (percent *percentage* | *rate*);

Hierarchy Level [edit class-of-service **traffic-control-profiles** *profile-name*]

Release Information Statement introduced in Junos OS Release 7.6.

Description For Gigabit Ethernet IQ, Channelized IQ PICs, and AS PIC FRF.16 LSQ interfaces only, base the delay-buffer calculation on a delay-buffer rate.

Default If you do not include this statement, the delay-buffer calculation is based on the guaranteed rate if one is configured, or the shaping rate if no guaranteed rate is configured. For more information, see Table 30 on page 187.

Options **percent *percentage***—For LSQ interfaces, delay-buffer rate as a percentage of the available interface bandwidth.

Range: 1 through 100 percent

rate—For IQ and IQ2 interfaces, delay-buffer rate, in bits per second (bps). You can specify a value in bits per second either as a complete decimal number or as a decimal number followed by the abbreviation **k** (1000), **m** (1,000,000), or **g** (1,000,000,000).

Range: 1000 through 160,000,000,000 bps

Usage Guidelines See "Oversubscribing Interface Bandwidth" on page 184, "Providing a Guaranteed Minimum Rate" on page 191, and "Configuring Traffic Control Profiles for Shared Scheduling and Shaping" on page 334.

Required Privilege Level interface—To view this statement in the configuration.
interface-control—To add this statement to the configuration.

Related Documentation

- **output-traffic-control-profile on page 528**

destination

Syntax destination *address*;

Hierarchy Level [edit interfaces *interface-name* unit *logical-unit-number* family inet **address** *address*],
[edit logical-systems *logical-system-name* interfaces *interface-name* unit *logical-unit-number* family inet **address** *address*]

Release Information Statement introduced before Junos OS Release 7.4.

Description For CoS on ATM interfaces, specify the remote address of the connection.

Options ***address***—Address of the remote side of the connection.

Required Privilege Level interface—To view this statement in the configuration.
interface-control—To add this statement to the configuration.

Related Documentation

- Configuring Linear RED Profiles on ATM Interfaces on page 412
- *Junos OS Network Interfaces Configuration Guide*

destination-address

Syntax destination-address (*address* | any-unicast) <except>;

Hierarchy Level [edit **services** cos rule *rule-name* term *term-name* **from**]

Release Information Statement introduced in Junos OS Release 8.1.

Description Specify the destination address for rule matching.

Options ***address***—Destination IP address or prefix value.

Required Privilege Level interface—To view this statement in the configuration.
interface-control—To add this statement to the configuration.

Related Documentation

- Configuring Match Conditions in a CoS Rule on page 284

discard

Syntax discard;

Hierarchy Level [edit class-of-service **forwarding-policy next-hop-map** *map-name* **forwarding-class** *class-name*]

Release Information Statement introduced in Junos OS Release 9.1.

Description Discard traffic sent to this forwarding class for the next-hop map referenced by this forwarding policy.

Usage Guidelines See “Configuring CoS-Based Forwarding” on page 130.

Required Privilege Level interface—To view this statement in the configuration.
interface-control—To add this statement to the configuration.

Related Documentation

- **non-lsp-next-hop on page 525**

drop-probability

See the following sections:

- **drop-probability (Interpolated Value) on page 462**
- **drop-probability (Percentage) on page 462**

drop-probability (Interpolated Value)

Syntax drop-probability [*values*];

Hierarchy Level [edit class-of-service **drop-profiles** *profile-name* **interpolate**]

Release Information Statement introduced before Junos OS Release 7.4.

Description Define up to 64 values for interpolating drop probabilities.

Options ***values***—Data points for interpolated packet drop probability.

Range: 0 through 100

Required Privilege Level interface—To view this statement in the configuration.
interface-control—To add this statement to the configuration.

Related Documentation

- Default Drop Profile on page 233

drop-probability (Percentage)

Syntax drop-probability *percentage*;

Hierarchy Level [edit class-of-service drop-profiles *profile-name*]

Release Information Statement introduced before Junos OS Release 7.4.

Description Define drop probability percentage.

Options ***percentage***—Probability that a packet is dropped, expressed as a percentage. A value of 0 means that a packet is never dropped, and a value of 100 means that all packets are dropped.

Range: 0 through 100 percent

Usage Guidelines See “Default Drop Profile” on page 233.

Required Privilege Level interface—To view this statement in the configuration.
interface-control—To add this statement to the configuration.

drop-profile

Syntax drop-profile *profile-name*;

Hierarchy Level [edit class-of-service schedulers *scheduler-name* drop-profile-map loss-priority (any | low | medium-low | medium-high | high) protocol (any | non-tcp | tcp)]

Release Information Statement introduced before Junos OS Release 7.4.

Description Define drop profiles for RED. When a packet arrives, RED checks the queue fill level. If the fill level corresponds to a nonzero drop probability, the RED algorithm determines whether to drop the arriving packet.

Options ***profile-name***—Name of the drop profile.

Required Privilege Level interface—To view this statement in the configuration.
interface-control—To add this statement to the configuration.

Related Documentation

- RED Drop Profiles Overview on page 231

drop-profile-map

Syntax drop-profile-map **loss-priority** (any | low | medium-low | medium-high | high) **protocol**(any | non-tcp | tcp) **drop-profile** *profile-name*;

Hierarchy Level [edit class-of-service schedulers *scheduler-name*]

Release Information Statement introduced before Junos OS Release 7.4.

Description Define loss priority value for drop profile.

The statements are explained separately.

Required Privilege Level interface—To view this statement in the configuration.
interface-control—To add this statement to the configuration.

Related Documentation

- Default Schedulers on page 147

drop-profiles

Syntax

```
drop-profiles {
  profile-name {
    fill-level percentage drop-probability percentage;
    interpolate {
      drop-probability [ values ];
      fill-level [ values ]
    }
  }
}
```

Hierarchy Level [edit class-of-service]

Release Information Statement introduced before Junos OS Release 7.4.

Description Define drop profiles for RED.

For a packet to be dropped, it must match the drop profile. When a packet arrives, RED checks the queue fill level. If the fill level corresponds to a nonzero drop probability, the RED algorithm determines whether to drop the arriving packet.

Options ***profile-name***—Name of the drop profile.

The remaining statements are explained separately.

Required Privilege Level interface—To view this statement in the configuration.
interface-control—To add this statement to the configuration.

Related Documentation

- Configuring RED Drop Profiles on page 233

drop-timeout

Syntax drop-timeout *milliseconds*;

Hierarchy Level [edit **class-of-service fragmentation-map** *map-name* **forwarding-class** *class-name*]

Release Information Statement introduced in Junos OS Release 8.2.

Description Disable or set the resequencing timeout interval for each forwarding class of a multiclass MLPPP.

Default If you do not include this statement, the default sequencing timeouts for T1 speeds (500 ms) or lower (1500 ms) apply.

Options ***milliseconds***—Time to wait for fragments. A value of 0 disables the resequencing logic for that forwarding class.

Range: 0 through 500 milliseconds for bundles with bandwidths or T1 speeds or higher or 1500 ms for bundles with bandwidths of less than T1 speeds.

Usage Guidelines See “Example: Configuring Drop Timeout Interval by Forwarding Class” on page 142.

Required Privilege Level interface—To view this statement in the configuration.
interface-control—To add this statement to the configuration.

dscp

See the following sections:

- **dscp (AS PIC Classifiers) on page 466**
- **dscp (Multifield Classifier) on page 467**
- **dscp (Rewrite Rules) on page 468**

dscp (AS PIC Classifiers)

Syntax dscp (*alias* | *bits*);

Hierarchy Level [edit **services** cos **application-profile** *profile-name* (**ftp** | **sip**) (**data** | **video** | **voice**)],
[edit **services** cos rule *rule-name* term *term-name* **then**],
[edit **services** cos rule *rule-name* term *term-name* then (**reflexive** | **reverse**)]

Release Information Statement introduced in Junos OS Release 8.1.

Description Define the Differentiated Services code point (DSCP) mapping that is applied to the packets.

Options ***alias***—Name assigned to a set of CoS markers.

bits—Mapping value in the packet header.

Required Privilege Level interface—To view this statement in the configuration.
interface-control—To add this statement to the configuration.

Related Documentation
- Configuring Actions in a CoS Rule on page 285

dscp (Multifield Classifier)

Syntax dscp [0 | *value*];

Hierarchy Level [edit firewall family *family-name* filter *filter-name* term *term-name* then]

Release Information Statement introduced in Junos OS Release 7.4.

Description For M320 and T Series routers, set the DSCP field of incoming or outgoing packets to **000000**. On the same packets, you can use a behavior aggregate (BA) classifier and a rewrite rule to rewrite the MPLS EXP field.

For MX Series routers with MPCs, the DSCP field can be set from a numeric range.

For MX Series routers, if you configure a firewall filter with a DSCP action or traffic-class action on a DPC, the commit does not fail, but a warning displays and an entry is made in the syslog.

Options ***value***—For MX Series routers with MPCs, specify the field of incoming or outgoing packets in the range from **0** through **63**.

Required Privilege Level firewall—To view this statement in the configuration.
firewall-control—To add this statement to the configuration.

Related Documentation

- Applying Tricolor Marking Policers to Firewall Filters on page 99

dscp (Rewrite Rules)

Syntax dscp (*rewrite-name* | default);

Hierarchy Level [edit class-of-service interfaces *interface-name* unit *logical-unit-number* **rewrite-rules**]

Release Information Statement introduced before Junos OS Release 7.4.

Description For IPv4 traffic, apply a Differentiated Services (DiffServ) code point (DSCP) rewrite rule.

NOTE: DSCP rewrite is not supported on the 10-port 10-Gigabit Oversubscribed Ethernet PIC.

Options *rewrite-name*—Name of a **rewrite-rules** mapping configured at the **[edit class-of-service rewrite-rules dscp]** hierarchy level.

default—The default mapping.

Required Privilege Level interface—To view this statement in the configuration.
interface-control—To add this statement to the configuration.

Related Documentation

- Configuring Rewrite Rules on page 242
- **dscp-ipv6 on page 469**
- **exp on page 475**
- **exp-push-push-push on page 476**
- **exp-swap-push-push on page 477**
- **ieee-802.1 on page 500**
- **ieee-802.1ad on page 501**
- **inet-precedence on page 504**
- **rewrite-rules (Definition) on page 546**

dscp-code-point

Syntax dscp-code-point *value*;

Hierarchy Level [edit class-of-service **host-outbound-traffic**]

Release Information Statement introduced in Junos OS Release 8.4.

Description Set the value of the DSCP code point in the ToS field of the packet generated by the Routing Engine (host).

Usage Guidelines See “Changing the Routing Engine Outbound Traffic Defaults” on page 269.

Required Privilege Level interface—To view this statement in the configuration.
interface-control—To add this statement to the configuration.

dscp-ipv6

Syntax dscp-ipv6 (*rewrite-name* | default);

Hierarchy Level [edit class-of-service interfaces *interface-name* unit *logical-unit-number* **rewrite-rules**]

Release Information Statement introduced before Junos OS Release 7.4.

Description For IPv6 traffic, apply a DSCP rewrite rule.

Options *rewrite-name*—Name of a **rewrite-rules** mapping configured at the **[edit class-of-service rewrite-rules dscp-ipv6]** hierarchy level.

default—The default mapping.

Required Privilege Level interface—To view this statement in the configuration.
interface-control—To add this statement to the configuration.

Related Documentation

- Configuring Rewrite Rules on page 242
- **dscp (Rewrite Rules) on page 468**
- **exp on page 475**
- **exp-push-push-push on page 476**
- **exp-swap-push-push on page 477**
- **ieee-802.1 on page 500**
- **ieee-802.1ad on page 501**
- **inet-precedence on page 504**
- **rewrite-rules (Definition) on page 546**

egress-shaping-overhead

Syntax egress-shaping-overhead *number*;

Hierarchy Level [edit chassis fpc *slot-number* pic *pic-number* **traffic-manager**],
[edit chassis lcc *number* fpc *slot-number* pic *pic-number* **traffic-manager**]

Release Information Statement introduced in Junos OS Release 8.3.

Description Number of bytes to add to packet to determine shaped session packet length.

Options ***number***—Number of bytes added to shaped packets.

Range: –63 through 192

Usage Guidelines See "Configuring CoS for L2TP Tunnels on ATM Interfaces" on page 426.

Required Privilege Level interface—To view this statement in the configuration.
interface-control—To add this statement to the configuration.

Related Documentation

- **mode on page 523**, **ingress-shaping-overhead on page 505**

epd-threshold

Syntax epd-threshold *cells* plp1 *cells*;

Hierarchy Level [edit interfaces at-*fpc/pic/port* atm-options scheduler-maps *map-name* forwarding-class *class-name*]

Release Information Statement introduced before Junos OS Release 7.4.

Description For ATM2 IQ interfaces only, define the EPD threshold on a VC. The EPD threshold is a limit on the number of transmit packets that can be queued. Packets that exceed the limit are discarded.

Default If you do not include either the **epd-threshold** or the **linear-red-profile** statement in the forwarding class configuration, the Junos OS uses an EPD threshold based on the available bandwidth and other parameters.

Options *cells*—Maximum number of cells.

Range: For 1-port and 2-port OC12 interfaces, 1 through 425,984 cells. For 1-port OC48 interfaces, 1 through 425,984 cells.For 2-port OC3, DS3, and E3 interfaces, 1 through 212,992 cells. For 4-port DS3 and E3 interfaces, 1 through 106,496 cells.

plp1 *cells*—Early packet drop threshold value for PLP 1.

Range: For 1-port and 2-port OC12 interfaces, 1 through 425,984 cells. For 1-port OC48 interfaces, 1 through 425,984 cells.For 2-port OC3, DS3, and E3 interfaces, 1 through 212,992 cells. For 4-port DS3 and E3 interfaces, 1 through 106,496 cells.

Required Privilege Level interface—To view this statement in the configuration.
interface-control—To add this statement to the configuration.

Related Documentation

- Configuring Scheduler Maps on ATM Interfaces on page 416
- **linear-red-profile on page 514**

excess-bandwidth-share

Syntax excess-bandwidth-share (proportional *value* | equal);

Hierarchy Level [edit class-of-service **interfaces interface-set** *interface-set-name*]

Release Information Statement introduced in Junos OS Release 8.5.

Description Determines the method of sharing excess bandwidth in a hierarchical scheduler environment. If you do not include this statement, the node shares excess bandwidth proportionally at 32.64 Mbps.

Options **proportional *value***—(Default) Share excess bandwidth proportionally (default value is 32.64 Mbps).

equal—Share excess bandwidth equally.

Required Privilege Level interface—To view this statement in the configuration.
interface-control—To add this statement to the configuration.

Related Documentation

- Configuring MDRR on Enhanced Queuing DPCs on page 362

excess-priority

Syntax excess-priority *value*;

Hierarchy Level [edit class-of-service **schedulers** *scheduler-name*]

Release Information Statement introduced in Junos OS Release 9.3.

Description Determine the priority of excess bandwidth traffic on a scheduler.

NOTE: For Link Services IQ (LSQ) PICs or Multiservices PIC (MS-PICs), the **excess-priority** statement is allowed for consistency, but ignored. If an explicit priority is not configured for these interfaces, a default low priority is used. This default priority is also used in the excess region.

Options **low**—Excess traffic for this scheduler has low priority.

medium-low—Excess traffic for this scheduler has medium-low priority.

medium-high—Excess traffic for this scheduler has medium-high priority.

high—Excess traffic for this scheduler has high priority.

Required Privilege Level interface—To view this statement in the configuration.
interface-control—To add this statement to the configuration.

Related Documentation

- Configuring Excess Bandwidth Sharing on IQE PICs on page 295
- Bandwidth Sharing on Nonqueuing Packet Forwarding Engines Overview on page 223
- Excess Bandwidth Distribution on the Trio MPC/MIC Overview on page 380

excess-rate

Syntax excess-rate (percent *percentage* | proportion *value*);

Hierarchy Level [edit class-of-service **schedulers** *scheduler-name*],
[edit class-of-service **traffic-control-profiles** *traffic-control-profile-name*]

Release Information Statement introduced in Junos OS Release 9.3.
Application to the Multiservices PIC added in Junos OS Release 9.5.
Application to the Trio MPC/MIC interfaces added in Junos OS Release 10.1.

Description For an Enhanced IQ PIC, Multiservices PIC, or a Trio MPC/MIC interface, determine the percentage of excess bandwidth traffic to share.

Options ***percentage***—Percentage of the excess bandwidth to share.

Range: 0 through 100 percent

value—Proportion of the excess bandwidth to share. Option available at the **[edit class-of-service traffic-class-profiles *traffic-control-profile-name*]** hierarchy level only.

Range: 0 through 1000

Required Privilege Level interface—To view this statement in the configuration.
interface-control—To add this statement to the configuration.

Related Documentation

- Configuring Excess Bandwidth Sharing on IQE PICs on page 295
- Allocating Excess Bandwidth Among Frame Relay DLCIs on Multiservices PICs on page 287
- Excess Bandwidth Distribution on the Trio MPC/MIC Overview on page 380

exp

Syntax exp (*rewrite-name* | default)**protocol** *protocol-types*;

Hierarchy Level [edit class-of-service interfaces *interface-name* unit *logical-unit-number* **rewrite-rules**]

Release Information Statement introduced before Junos OS Release 7.4.

Description Apply an MPLS experimental (EXP) rewrite rule.

Options *rewrite-name*—Name of a **rewrite-rules** mapping configured at the **[edit class-of-service rewrite-rules exp]** hierarchy level.

default—The default mapping.

By default, IP precedence rewrite rules alter the first three bits on the type-of-service (ToS) byte while leaving the last three bits unchanged. This default behavior applies to rewrite rules you configure for MPLS packets with IPv4 payloads. You configure these types of rewrite rules by including the **mpls-inet-both** or **mpls-inet-both-non-vpn** option at the **[edit class-of-service interfaces interface *interface-name* unit *logical-unit-number* rewrite-rules exp *rewrite-rule-name* protocol]** hierarchy level.

The remaining statement is explained separately.

Required Privilege Level interface—To view this statement in the configuration.
interface-control—To add this statement to the configuration.

Related Documentation

- Rewriting the EXP Bits of All Three Labels of an Outgoing Packet on page 251
- **dscp (Rewrite Rules) on page 468**
- **dscp-ipv6 on page 469**
- **exp-push-push-push on page 476**
- **exp-swap-push-push on page 477**
- **ieee-802.1 on page 500**
- **ieee-802.1ad on page 501**
- **inet-precedence on page 504**
- **rewrite-rules (Definition) on page 546**

exp-push-push-push

Syntax exp-push-push-push default;

Hierarchy Level [edit class-of-service interfaces *interface-name* unit *logical-unit-number* rewrite-rules]

Release Information Statement introduced before Junos OS Release 7.4.

Description For M Series routers, rewrite the EXP bits of all three labels of an outgoing packet, thereby maintaining CoS of an incoming non-MPLS packet.

Options **default**—Apply the default MPLS EXP rewrite table.

Required Privilege Level interface—To view this statement in the configuration.
interface-control—To add this statement to the configuration.

Related Documentation

- Rewriting the EXP Bits of All Three Labels of an Outgoing Packet on page 251
- **dscp (Rewrite Rules) on page 468**
- **dscp-ipv6 on page 469**
- **exp on page 475**
- **exp-swap-push-push on page 477**
- **ieee-802.1 on page 500**
- **ieee-802.1ad on page 501**
- **inet-precedence on page 504**
- **rewrite-rules (Definition) on page 546**

exp-swap-push-push

Syntax exp-swap-push-push default;

Hierarchy Level [edit class-of-service interfaces *interface-name* unit *logical-unit-number* rewrite-rules]

Release Information Statement introduced before Junos OS Release 7.4.

Description For M Series routers, rewrite the EXP bits of all three labels of an outgoing packet, thereby maintaining CoS of an incoming MPLS packet.

Options **default**—Apply the default MPLS EXP rewrite table.

Required Privilege Level interface—To view this statement in the configuration.
interface-control—To add this statement to the configuration.

Related Documentation

- Rewriting the EXP Bits of All Three Labels of an Outgoing Packet on page 251
- **dscp (Rewrite Rules) on page 468**
- **dscp-ipv6 on page 469**
- **exp on page 475**
- **exp-push-push-push on page 476**
- **ieee-802.1 on page 500**
- **ieee-802.1ad on page 501**
- **inet-precedence on page 504**
- **rewrite-rules (Definition) on page 546**

fabric

Syntax

```
fabric {
  scheduler-map {
    priority (high | low) scheduler scheduler-name;
  }
}
```

Hierarchy Level [edit class-of-service]

Release Information Statement introduced before Junos OS Release 7.4.

Description For M320 and T Series routers only, associate a scheduler with a fabric priority.

The remaining statements are explained separately.

Usage Guidelines See “Associating Schedulers with Fabric Priorities” on page 200.

Required Privilege Level interface—To view this statement in the configuration.
interface-control—To add this statement to the configuration.

family

See the following sections:

- **family (CoS on ATM Interfaces) on page 479**
- **family (Multifield Classifier) on page 480**

family (CoS on ATM Interfaces)

Syntax

```
family family {
  address address {
    destination address;
  }
}
```

Hierarchy Level

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

Release Information

Statement introduced before Junos OS Release 7.4.

Description

For CoS on ATM interfaces, configure the protocol family.

Options

family—Protocol family:

- **ccc**—Circuit cross-connect parameters
- **inet**—IPv4 parameters
- **inet6**—IPv6 protocol parameters
- **iso**—OSI ISO protocol parameters
- **mlppp**—Multilink PPP protocol parameters
- **mpls**—MPLS protocol parameters
- **tcc**—Translational cross-connect parameters
- **vpls**—Virtual private LAN service parameters.

The remaining statements are explained separately.

Required Privilege Level

interface—To view this statement in the configuration.
interface-control—To add this statement to the configuration.

Related Documentation

- CoS on ATM Interfaces Overview on page 411
- *Junos OS Network Interfaces Configuration Guide*

family (Multifield Classifier)

Syntax

```
family family-name {
  filter filter-name {
    term term-name {
      from {
        match-conditions;
      }
      then {
        dscp 0;
        forwarding-class class-name;
        loss-priority (high | low);
        three-color-policer {
          (single-rate | two-rate) policer-name;
        }
      }
    }
  }
}
```

Hierarchy Level [edit firewall]

Release Information Statement introduced before Junos OS Release 7.4.

Description Configure a firewall filter for IP version 4 (IPv4) or IP version 6 (IPv6) traffic.

Options *family-name*—Protocol family:

- **ccc**—Circuit cross-connect parameters
- **inet**—IPv4 parameters
- **inet6**—IPv6 protocol parameters
- **iso**—OSI ISO protocol parameters
- **mlppp**—Multilink PPP protocol parameters
- **mpls**—MPLS protocol parameters
- **tcc**—Translational cross-connect parameters
- **vpls**—Virtual private LAN service parameters.

The remaining statements are explained separately.

Required Privilege Level interface—To view this statement in the configuration.
interface-control—To add this statement to the configuration.

Related Documentation

- Configuring Multifield Classifiers on page 72
- *Junos OS Policy Framework Configuration Guide*

fill-level

See the following sections:

- **fill-level (Interpolated Value) on page 481**
- **fill-level (Percentage) on page 481**

fill-level (Interpolated Value)

Syntax fill-level [*values*];

Hierarchy Level [edit class-of-service **drop-profiles** *profile-name* interpolate]

Release Information Statement introduced before Junos OS Release 7.4.

Description Define up to 64 values for interpolating queue fill level.

Options ***values***—Data points for mapping queue fill percentage.

Range: 0 through 100

Usage Guidelines See "Configuring RED Drop Profiles" on page 233.

Required Privilege Level interface—To view this statement in the configuration.
interface-control—To add this statement to the configuration.

fill-level (Percentage)

Syntax fill-level *percentage*;

Hierarchy Level [edit class-of-service **drop-profiles** *profile-name*]

Release Information Statement introduced before Junos OS Release 7.4.

Description When configuring RED, map the fullness of a queue to a drop probability.

Options ***percentage***—How full the queue is, expressed as a percentage. You configure the **fill-level** and **drop-probability** statements in pairs. To specify multiple fill levels, include multiple **fill-level** and **drop-probability** statements. The values you assign to each statement pair must increase relative to the previous pair's values. This is shown in the "Segmented" graph in "RED Drop Profiles Overview" on page 231.

Range: 0 through 100 percent

Required Privilege Level interface—To view this statement in the configuration.
interface-control—To add this statement to the configuration.

Related Documentation

- Configuring RED Drop Profiles on page 233
- **drop-probability (Interpolated Value) on page 462**

filter

See the following sections:

- **filter (Applying to an Interface) on page 482**
- **filter (Configuring) on page 483**

filter (Applying to an Interface)

Syntax

```
filter {
  input filter-name;
  output filter-name;
}
```

Hierarchy Level

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

Release Information

Statement introduced before Junos OS Release 7.4.

Description

Apply a filter to an interface. You can also use filters for encrypted traffic. When you configure filters, you can configure the family **inet**, **inet6**, **mpls**, or **vpls** only.

Options

input *filter-name*—Name of one filter to evaluate when packets are received on the interface.

output *filter-name*—Name of one filter to evaluate when packets are transmitted on the interface.

Usage Guidelines

See “Configuring Multifield Classifiers” on page 72 and “Using Multifield Classifiers to Set PLP” on page 103; for a general discussion of this statement, see the *Junos OS Network Interfaces Configuration Guide*.

Required Privilege Level

interface—To view this statement in the configuration.
interface-control—To add this statement to the configuration.

Related Documentation

- **simple-filter on page 571**

filter (Configuring)

Syntax

```
filter filter-name {
  term term-name {
    from {
      match-conditions;
    }
    then {
      dscp 0;
      forwarding-class class-name;
      loss-priority (high | low);
      policer policer-name;
      three-color-policer {
        (single-rate | two-rate) policer-name;
      }
    }
  }
}
```

Hierarchy Level [edit firewall **family** *family-name*]

Release Information Statement introduced before Junos OS Release 7.4.

Description Configure firewall filters.

Options *filter-name*—Name that identifies the filter. The name can contain letters, numbers, and hyphens (-) and can be up to 255 characters long. To include spaces in the name, enclose it in quotation marks (“ ”).

The remaining statements are explained separately.

Required Privilege Level firewall—To view this statement in the configuration.
firewall-control—To add this statement to the configuration.

Related Documentation

- Configuring Multifield Classifiers on page 72
- Using Multifield Classifiers to Set PLP on page 103
- *Junos OS Policy Framework Configuration Guide*
- **simple-filter (Configuring) on page 572**

firewall

Syntax firewall { ... }

Hierarchy Level [edit]

Release Information Statement introduced before Junos OS Release 7.4.

Description Configure firewall filters.

The statements are explained separately.

Usage Guidelines See “Configuring Multifield Classifiers” on page 72 and “Using Multifield Classifiers to Set PLP” on page 103; for a general discussion of this statement, see the *Junos OS Policy Framework Configuration Guide*.

Required Privilege Level firewall—To view this statement in the configuration.
firewall-control—To add this statement to the configuration.

forwarding-class

See the following sections:

forwarding-class (AS PIC Classifiers)

Syntax forwarding-class *class-name*;

Hierarchy Level [edit **services** cos application-profile *profile-name* (**ftp** | **sip**) (**data** | **video** | **voice**)],
[edit **services** cos rule *rule-name* term *term-name* **then**],
[edit **services** cos rule *rule-name* term *term-name* then (**reflexive** | **reverse**)]

Release Information Statement introduced in Junos OS Release 8.1.

Description Define the forwarding class to which packets are assigned.

Options ***class-name***—Name of the target application.

Required Privilege Level interface—To view this statement in the configuration.
interface-control—To add this statement to the configuration.

Related Documentation
- Configuring Actions in a CoS Rule on page 285

forwarding-class (ATM2 IQ Scheduler Maps)

Syntax

forwarding-class *class-name* {
  epd-threshold *cells* plp1 *cells*;
  linear-red-profile *profile-name*;
  priority (high | low);
  transmit-weight (cells *number* percent *number*);
}

Hierarchy Level [edit interfaces at-*fpc/pic/port* **atm-options scheduler-maps** *map-name*]

Release Information Statement introduced before Junos OS Release 7.4.

Description For ATM2 IQ interfaces only, define forwarding class name and option values.

Options ***class-name***—Name of the forwarding class.

The statements are explained separately.

Required Privilege Level interface—To view this statement in the configuration.
interface-control—To add this statement to the configuration.

Related Documentation

- Configuring Scheduler Maps on ATM Interfaces on page 416

forwarding-class (BA Classifiers)

Syntax

forwarding-class *class-name* {
  loss-priority *level* **code-points** [*aliases*] [*bit-patterns*];
}

Hierarchy Level [edit class-of-service classifiers *type classifier-name*]

Release Information Statement introduced before Junos OS Release 7.4.

Description Define forwarding class name and option values.

Options ***class-name***—Name of the forwarding class.

The remaining statements are explained separately.

Required Privilege Level interface—To view this statement in the configuration.
interface-control—To add this statement to the configuration.

Related Documentation

- Defining Classifiers on page 49
- Example: Configuring CoS for a PBB Network on MX Series Routers

forwarding-class (Forwarding Policy)

Syntax

```
forwarding-class class-name {
   next-hop [ next-hop-name ];
   lsp-next-hop [ lsp-regular-expression ];
   non-lsp-next-hop;
   discard;
}
```

Hierarchy Level [edit class-of-service **forwarding-policy next-hop-map** *map-name*]

Release Information Statement introduced before Junos OS Release 7.4.

Description Define forwarding class name and associated next hops.

Options ***class-name***—Name of the forwarding class.

The remaining statement is explained separately.

Usage Guidelines See “Overriding the Input Classification” on page 132.

Required Privilege Level interface—To view this statement in the configuration.
interface-control—To add this statement to the configuration.

forwarding-class (Fragmentation)

Syntax

forwarding-class *class-name* {
  drop-timeout *milliseconds*;
  fragment-threshold *bytes*;
  multilink-class *number*;
  no-fragmentation;
}

Hierarchy Level [edit class-of-service **fragmentation-maps** *map-name*];

Release Information Statement introduced before Junos OS Release 7.4.

Description For AS PIC link services IQ interfaces (**lsq**) only, define a forwarding class name and associated fragmentation properties within a fragmentation map.

The **fragment-threshold** and **no-fragmentation** statements are mutually exclusive.

Default If you do not include this statement, the traffic in forwarding class ***class-name*** is fragmented.

Options ***class-name***—Name of the forwarding class.

The remaining statements are explained separately.

Usage Guidelines See “Configuring Fragmentation by Forwarding Class” on page 140.

Required Privilege Level interface—To view this statement in the configuration.
interface-control—To add this statement to the configuration.

forwarding-class (Interfaces)

Syntax forwarding-class *class-name*;

Hierarchy Level [edit class-of-service interfaces *interface-name* unit *logical-unit-number*]

Release Information Statement introduced before Junos OS Release 7.4.

Description Associate a forwarding class configuration or default mapping with a specific interface.

Options ***class-name***—Name of the forwarding class.

Usage Guidelines See “Applying Forwarding Classes to Interfaces” on page 115.

Required Privilege Level interface—To view this statement in the configuration.
interface-control—To add this statement to the configuration.

forwarding-class (Multifield Classifiers)

Syntax forwarding-class *class-name*;

Hierarchy Level [edit firewall family *family-name* filter *filter-name* term *term-name* then]

Release Information Statement introduced before Junos OS Release 7.4.

Description Set the forwarding class of incoming packets.

Options ***class-name***—Name of the forwarding class.

Required Privilege Level interface—To view this statement in the configuration.
interface-control—To add this statement to the configuration.

Related Documentation

- Configuring Multifield Classifiers on page 72
- *Junos OS Policy Framework Configuration Guide*

forwarding-class (Restricted Queues)

Syntax forwarding-class *class-name* queue *queue-number*;

Hierarchy Level [edit class-of-service restricted-queues]

Release Information Statement introduced before Junos OS Release 7.4.

Description For M320 and T Series routers only, map forwarding classes to restricted queues. You can map up to eight forwarding classes to restricted queues.

Options ***class-name***—Name of the forwarding class.

The remaining statement is explained separately.

Usage Guidelines

Required Privilege Level interface—To view this statement in the configuration.
interface-control—To add this statement to the configuration.

forwarding-classes

Syntax

```
forwarding-classes {
  class queue-num queue-number priority (high | low);
  queue queue-number class-name priority (high | low) [ policing-priority (premium | normal) ];
}
```

Hierarchy Level [edit **class-of-service**]

Release Information Statement introduced before Junos OS Release 7.4.
policing-priority option introduced in Junos OS Release 9.5.

Description Associate the forwarding class with a queue name and number. For M320, MX Series, and T Series routers only, you can configure fabric priority queuing by including the **priority** statement. For Enhanced IQ PICs, you can include the **policing-priority** option.

The statements are explained separately.

Usage Guidelines See “Configuring Forwarding Classes” on page 115, “Overriding Fabric Priority Queuing” on page 120, and Example: Configuring CoS for a PBB Network on MX Series Routers. For the **policing-priority** option, see “Configuring Layer 2 Policers on IQE PICs” on page 322. For classification by egress interface, see “Classifying Packets by Egress Interface” on page 116.

Required Privilege Level interface—To view this statement in the configuration.
interface-control—To add this statement to the configuration.

forwarding-classes-interface-specific

Syntax

```
forwarding-classes-interface-specific forwarding-class-map-name {
  class class-name queue-num queue-number [ restricted-queue queue-number ];
}
```

Hierarchy Level [edit **class-of-service**]

Release Information Statement introduced in Junos OS Release 9.6.

Description For the IQ, IQE, LSQ and ATM2 PICs in the T Series routers only, configure a forwarding class map for unicast and multicast traffic and a user-configured queue number for an egress interface.

Options ***class-name***—Name of the forwarding class.

forwarding-class-map-name—Name of the forwarding class map for traffic.

queue-number—Number of the egress queue.

Range: 0 through 3 or 7, depending on chassis and configuration

Usage Guidelines See “Configuring Forwarding Classes” on page 115 and “Classifying Packets by Egress Interface” on page 116.

Required Privilege Level interface—To view this statement in the configuration.
interface-control—To add this statement to the configuration.

Related Documentation

- **output-forwarding-class-map on page 526**

forwarding-policy

Syntax

```
forwarding-policy {
  next-hop-map map-name {
    forwarding-class class-name {
      next-hop [ next-hop-name ];
      lsp-next-hop [ lsp-regular-expression ];
      non-lsp-next-hop;
      discard;
    }
  }
  class class-name {
    classification-override {
      forwarding-class class-name;
    }
  }
}
```

Hierarchy Level [edit class-of-service]

Release Information Statement introduced before Junos OS Release 7.4.

Description Define CoS-based forwarding policy options.

The statements are explained separately.

Usage Guidelines See "Configuring CoS-Based Forwarding" on page 130.

Required Privilege Level interface—To view this statement in the configuration.
interface-control—To add this statement to the configuration.

fragment-threshold

Syntax fragment-threshold *bytes*;

Hierarchy Level [edit class-of-service **fragmentation-maps** *map-name* **forwarding-class** *class-name*]

Release Information Statement introduced before Junos OS Release 7.4.

Description For AS PIC link services IQ interfaces (**lsq**) only, set the fragmentation threshold for an individual forwarding class.

Default If you do not include this statement, the fragmentation threshold you set at the **[edit interfaces *interface-name*** unit ***logical-unit-number*]** or **[edit interfaces *interface-name* mlfr-uni-nni-bundle-options]** hierarchy level is the default for all forwarding classes. If you do not set a maximum fragment size anywhere in the configuration, packets are fragmented if they exceed the smallest MTU of all the links in the bundle.

Options ***bytes***—Maximum size, in bytes, for multilink packet fragments.

Range: 80 through 16,320 bytes

Usage Guidelines See “Configuring Fragmentation by Forwarding Class” on page 140.

Required Privilege Level interface—To view this statement in the configuration.
interface-control—To add this statement to the configuration.

fragmentation-map

Syntax fragmentation-map *map-name*;

Hierarchy Level [edit class-of-service interfaces *interface-name* unit *logical-unit-number*]

Release Information Statement introduced before Junos OS Release 7.4.

Description For AS PIC link services IQ (**lsq**) and virtual LSQ redundancy (**rlsq**) interfaces, associate a fragmentation map with a multilink PPP interface or MLFR FRF.16 DLCI.

Default If you do not include this statement, traffic in all forwarding classes is fragmented.

Options ***map-name***—Name of the fragmentation map.

Usage Guidelines See “Configuring Fragmentation by Forwarding Class” on page 140 and the *Junos OS Services Interfaces Configuration Guide*.

Required Privilege Level interface—To view this statement in the configuration.
interface-control—To add this statement to the configuration.

fragmentation-maps

Syntax

```
fragmentation-maps {
  map-name {
    forwarding-class class-name {
      drop-timeout milliseconds;
      fragment-threshold bytes;
      multilink-class number;
      no-fragmentation;
    }
  }
}
```

Hierarchy Level [edit **class-of-service**]

Release Information Statement introduced before Junos OS Release 7.4.

Description For AS PIC link services IQ (**lsq**) and virtual LSQ redundancy (**rlsq**) interfaces, define fragmentation properties for individual forwarding classes.

Default If you do not include this statement, traffic in all forwarding classes is fragmented.

Options ***map-name***—Name of the fragmentation map.

The remaining statements are explained separately.

Usage Guidelines See "Configuring Fragmentation by Forwarding Class" on page 140 and the *Junos OS Services Interfaces Configuration Guide*.

Required Privilege Level interface—To view this statement in the configuration.
interface-control—To add this statement to the configuration.

from

Syntax

```
from {
  applications [ application-name ];
  application-sets [ set-name ];
  destination-address address;
  source-address address;
}
```

Hierarchy Level [edit services cos rule *rule-name* term *term-name*]

Release Information Statement introduced in Junos OS Release 8.1.

Description Specify input conditions for a CoS term.

Options The remaining statements are explained separately.

For information on match conditions, see the description of firewall filter match conditions in the *Junos OS Policy Framework Configuration Guide*.

Required Privilege Level interface—To view this statement in the configuration.
interface-control—To add this statement to the configuration.

Related Documentation

- Configuring CoS Rule Sets on page 286

ftp

Syntax

```
ftp {
  data {
    dscp (alias | bits);
    forwarding-class class-name;
  }
}
```

Hierarchy Level [edit services cos **application-profile** *profile-name* ftp]

Release Information Statement introduced in Junos OS Release 9.3.

Description Set the appropriate **dscp** and **forwarding-class** value for FTP.

Default By default, the system does not alter the DSCP or forwarding class for FTP traffic.

Required Privilege Level interface—To view this statement in the configuration.
interface-control—To add this statement to the configuration.

Related Documentation

- Configuring Application Profiles on page 285
- **sip on page 573**

guaranteed-rate

Syntax guaranteed-rate (percent *percentage* | *rate*);

Hierarchy Level [edit class-of-service **traffic-control-profiles** *profile-name*]

Release Information Statement introduced in Junos OS Release 7.6.

Description For Gigabit Ethernet IQ, Channelized IQ PICs, and AS PIC FRF.16 LSQ interfaces only, configure a guaranteed minimum rate for a logical interface.

Default If you do not include this statement and you do not include the **delay-buffer-rate** statement, the logical interface receives a minimal delay-buffer rate and minimal bandwidth equal to 2 MTU-sized packets.

Options **percent *percentage***—For LSQ interfaces, guaranteed rate as a percentage of the available interface bandwidth.

Range: 1 through 100 percent

rate—For IQ and IQ2 interfaces, guaranteed rate, in bits per second (bps). You can specify a value in bits per second either as a complete decimal number or as a decimal number followed by the abbreviation **k** (1000), **m** (1,000,000), or **g** (1,000,000,000).

Range: 1000 through 160,000,000,000 bps

Usage Guidelines See “Providing a Guaranteed Minimum Rate” on page 191 and “Configuring Traffic Control Profiles for Shared Scheduling and Shaping” on page 334.

Required Privilege Level interface—To view this statement in the configuration.
interface-control—To add this statement to the configuration.

Related Documentation
- **output-traffic-control-profile on page 528**

hierarchical-scheduler

Syntax hierarchical-scheduler;

Hierarchy Level [edit class-of-service **interfaces**]

Release Information Statement introduced in Junos OS Release 8.5.

Description On MX Series, M Series, and T Series routers with IQ2E PIC, enables the use of hierarchical schedulers.

Default If you do not include this statement, the interfaces on the MX Series router cannot use hierarchical interfaces.

Required Privilege Level interface—To view this statement in the configuration.
interface-control—To add this statement to the configuration.

Related Documentation

- Configuring Hierarchical Schedulers for CoS on page 207

high-plp-max-threshold

Syntax high-plp-max-threshold *percent*;

Hierarchy Level [edit interfaces at-*fpc/pic/port* **atm-options linear-red-profiles** *profile-name*]

Release Information Statement introduced before Junos OS Release 7.4.

Description For ATM2 IQ interfaces only, define the drop profile fill-level for the high PLP CoS VC. When the fill level exceeds the defined percentage, all packets are dropped.

Options *percent*—Fill-level percentage when linear random early detection (RED) is applied to cells with PLP.

Required Privilege Level interface—To view this statement in the configuration.
interface-control—To add this statement to the configuration.

Related Documentation

- Configuring Linear RED Profiles on ATM Interfaces on page 412
- **low-plp-max-threshold on page 520**
- **low-plp-threshold on page 520**
- **queue-depth on page 543**

high-plp-threshold

Syntax high-plp-threshold *percent*;

Hierarchy Level [edit interfaces at-*fpc/pic/port* atm-options linear-red-profiles *profile-name*]

Release Information Statement introduced before Junos OS Release 7.4.

Description For ATM2 IQ interfaces only, define CoS VC drop profile fill-level percentage when linear RED is applied to cells with high PLP. When the fill level exceeds the defined percentage, packets with high PLP are randomly dropped by RED. This statement is mandatory.

Options *percent*—Fill-level percentage when linear RED is applied to cells with PLP.

Required Privilege Level interface—To view this statement in the configuration.
interface-control—To add this statement to the configuration.

Related Documentation

- Configuring Linear RED Profiles on ATM Interfaces on page 412
- **high-plp-max-threshold on page 497**
- **low-plp-max-threshold on page 520**
- **low-plp-threshold on page 520**
- **queue-depth on page 543**

host-outbound-traffic

Syntax

```
host-outbound-traffic {
  forwarding-class class-name;
  dscp-code-point value;
}
```

Hierarchy Level [edit class-of-service]

Release Information Statement introduced in Junos OS Release 8.4.

Description Allow queue selection for all traffic generated by the Routing Engine (host). The selected queue must be configured properly. The configuration of specific DSCP code point bits for the ToS field of the generated packets is also allowed. Transit packets are not affected; only packets originating on the Routing Engine are affected. By default, the forwarding class (queue) and DSCP bits are set according to those given in “Default Routing Engine Protocol Queue Assignments” on page 267. This feature is not available on J Series routers.

Options The statements are explained separately.

Usage Guidelines See “Changing the Routing Engine Outbound Traffic Defaults” on page 269.

Required Privilege Level interface—To view this statement in the configuration.
interface-control—To add this statement to the configuration.

ieee-802.1

Syntax ieee-802.1 (*rewrite-name* | default) **vlan-tag** (outer | outer-and-inner);

Hierarchy Level [edit class-of-service interfaces *interface-name* unit *logical-unit-number* **rewrite-rules**]

Release Information Statement introduced before Junos OS Release 7.4.
vlan-tag statement introduced in Junos OS Release 8.1.

Description Apply an IEEE-802.1 rewrite rule. For IQ PICs, you can only configure one IEEE 802.1 rewrite rule on a physical port. All logical ports (units) on that physical port should apply the same IEEE 802.1 rewrite rule.

Options *rewrite-name*—Name of a **rewrite-rules** mapping configured at the **[edit class-of-service rewrite-rules ieee-802.1]** hierarchy level.

default—The default mapping.

Required Privilege Level interface—To view this statement in the configuration.
interface-control—To add this statement to the configuration.

Related Documentation

- Configuring Rewrite Rules on page 242
- Example: Configuring CoS for a PBB Network on MX Series Routers
- **dscp (Rewrite Rules) on page 468**
- **dscp-ipv6 on page 469**
- **exp on page 475**
- **exp-push-push-push on page 476**
- **exp-swap-push-push on page 477**
- **ieee-802.1ad on page 501**
- **inet-precedence on page 504**
- **rewrite-rules (Definition) on page 546**

ieee-802.1ad

Syntax ieee-802.1ad (*rewrite-name* | default) **vlan-tag** (outer | outer-and-inner);

Hierarchy Level [edit class-of-service interfaces *interface-name* unit *logical-unit-number* **rewrite-rules**]

Release Information Statement introduced in Junos OS Release 9.2.

Description Apply a IEEE-802.1ad rewrite rule.

Options *rewrite-name*—Name of a **rewrite-rules** mapping configured at the **[edit class-of-service rewrite-rules ieee-802.1ad]** hierarchy level.

default—The default rewrite bit mapping.

vlan-tag—The rewrite rule is applied to the **outer** or **outer-and-inner** VLAN tag.

Required Privilege Level interface—To view this statement in the configuration.
interface-control—To add this statement to the configuration.

Related Documentation

- Configuring Rewrite Rules on page 242
- Example: Configuring CoS for a PBB Network on MX Series Routers
- **dscp (Rewrite Rules) on page 468**
- **dscp-ipv6 on page 469**
- **exp on page 475**
- **exp-push-push-push on page 476**
- **exp-swap-push-push on page 477**
- **ieee-802.1 on page 500**
- **inet-precedence on page 504**
- **rewrite-rules (Definition) on page 546**

if-exceeding

Syntax

```
if-exceeding {
  bandwidth-limit rate;
  bandwidth-percent number;
  burst-size-limit bytes;
}
```

Hierarchy Level [edit firewall **policer** *policer-name*]

Release Information Statement introduced before Junos OS Release 7.4.

Description Configure policer rate limits.

Options **bandwidth-limit** *bps*—Traffic rate, in bits per second (bps). There is no minimum value, but any value below 61,040 bps results in an effective rate of 30,520 bps.

Range: 8,000 through 50,000,000,000 bps (on the MX Series, M7i, M10i, M120, M320, T640, T1600, and TX-Matrix routers)

Range: 32,000 through 50,000,000,000 bps (on any platform except for the MX Series, M7i, M10i, M120, M320, T640, T1600, and TX-Matrix routers)

Default: None

bandwidth-percent *number*—Port speed, in decimal percentage number.

Range: 1 through 100

Default: None

burst-size-limit *bytes*—Maximum burst size, in bytes. The minimum recommended value is the maximum transmission unit (MTU) of the IP packets being policed.

Range: 1500 through 100,000,000 bytes

Default: None

NOTE: On M120, M320, and T Series routers, you can specify a minimum bandwidth limit of 8k (8000 bps). On the MX Series routers, you can specify a minimum bandwidth limit of 65,535 bps. Values below 65,535, even if allowed, will generate a commit error.

Usage Guidelines See "Configuring Multifield Classifiers" on page 72, "Using Multifield Classifiers to Set PLP" on page 103, and "Configuring Schedulers for Priority Scheduling" on page 165; for a general discussion of this statement, see the *Junos OS Policy Framework Configuration Guide*.

Required Privilege Level firewall—To view this statement in the configuration.
firewall-control—To add this statement to the configuration.

Related Documentation

- **filter (Configuring) on page 483**, **priority (Schedulers) on page 539**

import

See the following sections:

- **import (Classifiers) on page 503**
- **import (Rewrite Rules) on page 503**

import (Classifiers)

Syntax import (*classifier-name* | default);

Hierarchy Level [edit class-of-service classifiers *type classifier-name*]

Release Information Statement introduced before Junos OS Release 7.4.

Description Specify a default or previously defined classifier.

Options ***classifier-name***—Name of the classifier mapping configured at the **[edit class-of-service classifiers]** hierarchy level.

default—The default classifier mapping.

Required Privilege Level interface—To view this statement in the configuration.
interface-control—To add this statement to the configuration.

Related Documentation

- Overview of BA Classifier Types on page 42

import (Rewrite Rules)

Syntax import (*rewrite-name* | default);

Hierarchy Level [edit class-of-service rewrite-rules *type rewrite-name*]

Release Information Statement introduced before Junos OS Release 7.4.

Description Specify a default or previously defined **rewrite-rules** mapping to import.

Options ***rewrite-name***—Name of a rewrite-rules mapping configured at the **[edit class-of-service rewrite-rules]** hierarchy level.

default—The default **rewrite-rules** mapping.

Required Privilege Level interface—To view this statement in the configuration.
interface-control—To add this statement to the configuration.

Related Documentation

- Configuring Rewrite Rules on page 242

inet-precedence

Syntax inet-precedence (*rewrite-name* | default);

Hierarchy Level [edit class-of-service interfaces *interface-name* unit *logical-unit-number* **rewrite-rules**]

Release Information Statement introduced before Junos OS Release 7.4.

Description Apply a IPv4 precedence rewrite rule.

Options *rewrite-name*—Name of a **rewrite-rules** mapping configured at the **[edit class-of-service rewrite-rules inet-precedence]** hierarchy level.

default—The default mapping. By default, IP precedence rewrite rules alter the first three bits on the type of service (ToS) byte while leaving the last three bits unchanged.

Required Privilege Level interface—To view this statement in the configuration.
interface-control—To add this statement to the configuration.

Related Documentation

- Configuring Rewrite Rules on page 242
- **dscp (Rewrite Rules) on page 468**
- **dscp-ipv6 on page 469**
- **exp on page 475**
- **exp-push-push-push on page 476**
- **exp-swap-push-push on page 477**
- **ieee-802.1 on page 500**
- **ieee-802.1ad on page 501**
- **rewrite-rules (Definition) on page 546**

ingress-shaping-overhead

Syntax ingress-shaping-overhead *number*;

Hierarchy Level [edit chassis fpc *slot-number* pic *pic-number* **traffic-manager**],
[edit chassis lcc *number* fpc *slot-number* pic *pic-number* **traffic-manager**]

Release Information Statement introduced in Junos OS Release 8.3.

Description Number of bytes to add to packet to determine shaped session packet length.

Options *number*—Number of bytes added to shaped packets.

Range: –63 through 192

Required Privilege Level interface—To view this statement in the configuration.
interface-control—To add this statement to the configuration.

Related Documentation

- Configuring CoS for L2TP Tunnels on ATM Interfaces on page 426
- **egress-shaping-overhead on page 470**
- **mode on page 523**

input-excess-bandwith-share

Syntax input-excess-bandwidth-share (proportional *value* | equal);

Hierarchy Level [edit class-of-service **interfaces** *interface-name*],
[edit class-of-service **interfaces interface-set** *interface-set-name*]

Release Information Statement introduced in Junos OS Release 9.0.

Description Determines the method of sharing excess bandwidth on the ingress interface in a hierarchical scheduler environment. If you do not include this statement, the node shares excess bandwidth proportionally at 32.64 Mbps.

Options **proportional** *value*—(Default) Share ingress excess bandwidth proportionally (default value is 32.64 Mbps).

equal—Share ingress excess bandwidth equally.

Required Privilege Level interface—To view this statement in the configuration.
interface-control—To add this statement to the configuration.

Related Documentation

- Configuring Ingress Hierarchical CoS on Enhanced Queuing DPCs on page 368

input-policer

Syntax input-policer *policer-name*;

Hierarchy Level [edit interfaces *interface-name* unit *logical-unit-number* **layer2-policer**]

Release Information Statement introduced in Junos OS Release 8.2.

Description Associate a Layer 2 policer with a logical interface. The **input-policer** and **input-three-color** statements are mutually exclusive.

Options *policer-name*—Name of the policer that you define at the **[edit firewall]** hierarchy level.

Usage Guidelines See “Applying Layer 2 Policers to Gigabit Ethernet Interfaces” on page 101.

Required Privilege Level interface—To view this statement in the configuration.
interface-control—To add this statement to the configuration.

Related Documentation

- **output-policer on page 526**

input-scheduler-map

Syntax input-scheduler-map *map-name*;

Hierarchy Level [edit class-of-service interfaces *interface-name*],
[edit class-of-service interfaces *interface-name* **unit** *logical-unit-number*]

Release Information Statement introduced in Junos OS Release 7.6.

Description Associate a scheduler map with a physical or logical interface. The **input-scheduler-map** and **input-traffic-control-profile** statements are mutually exclusive.

Options *map-name*—Name of scheduler map that you define at the **[edit interfaces** ***interface-name*** **atm-options scheduler-maps]** hierarchy level.

default—The default scheduler mapping.

Required Privilege Level interface—To view this statement in the configuration.
interface-control—To add this statement to the configuration.

Related Documentation

- Configuring a Separate Input Scheduler for Each Interface on page 338
- Configuring Ingress Hierarchical CoS on Enhanced Queuing DPCs on page 368
- **input-traffic-control-profile on page 509**

input-shaping-rate

See the following sections:

- **input-shaping-rate (Logical Interface) on page 507**
- **input-shaping-rate (Physical Interface) on page 508**

input-shaping-rate (Logical Interface)

Syntax input-shaping-rate (percent *percentage* | *rate*);

Hierarchy Level [edit class-of-service interfaces *interface-name* unit *logical-unit-number*]

Release Information Statement introduced in Junos OS Release 7.6.

Description For Gigabit Ethernet IQ2 interfaces, configure input traffic shaping by specifying the amount of bandwidth to be allocated to the logical interface. You can configure hierarchical shaping, meaning you can apply an input shaping rate to both the physical and logical interface.

Default If you do not include this statement, logical interfaces 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.

Options **percent *percentage***—Shaping rate as a percentage of the available interface bandwidth.

Range: 0 through 100 percent

rate—Peak rate, in bits per second (bps). You can specify a value in bits per second either as a complete decimal number or as a decimal number followed by the abbreviation **k** (1000), **m** (1,000,000), or **g** (1,000,000,000).

Range: 1000 through 160,000,000,000 bps.

Usage Guidelines See "Configuring Hierarchical Input Shapers" on page 341, "Configuring Ingress Hierarchical CoS on Enhanced Queuing DPCs" on page 368.

Required Privilege Level interface—To view this statement in the configuration.
interface-control—To add this statement to the configuration.

Related Documentation

- **input-traffic-control-profile on page 509**

input-shaping-rate (Physical Interface)

Syntax
input-shaping-rate *rate*;

Hierarchy Level
[edit class-of-service interfaces *interface-name*]

Release Information
Statement introduced in Junos OS Release 7.6.

Description
For Gigabit Ethernet IQ2 interfaces, configure input traffic shaping by specifying the amount of bandwidth to be allocated to the physical interface. You can configure hierarchical shaping, meaning you can apply an input shaping rate to both the physical and logical interface.

Options
rate—Peak rate, in bits per second (bps). You can specify a value in bits per second either as a complete decimal number or as a decimal number followed by the abbreviation **k** (1000), **m** (1,000,000), or **g** (1,000,000,000).

Range: 1000 through 160,000,000,000 bps.

Required Privilege Level
interface—To view this statement in the configuration.
interface-control—To add this statement to the configuration.

Related Documentation
- Configuring Hierarchical Input Shapers on page 341
- Configuring Ingress Hierarchical CoS on Enhanced Queuing DPCs on page 368
- **input-traffic-control-profile on page 509**

input-three-color

Syntax
input-three-color *policer-name*;

Hierarchy Level
[edit interfaces *interface-name* unit *logical-unit-number* layer2-policer]

Release Information
Statement introduced in Junos OS Release 8.2.

Description
Associate a Layer 2, three-color policer with a logical interface. The **input-three-color** and **input-policer** statements are mutually exclusive.

Options
policer-name—Name of the three-color policer that you define at the **[edit firewall]** hierarchy level.

Usage Guidelines
See “Applying Layer 2 Policers to Gigabit Ethernet Interfaces” on page 101.

Required Privilege Level
interface—To view this statement in the configuration.
interface-control—To add this statement to the configuration.

Related Documentation
- **output-three-color on page 527**

input-traffic-control-profile

Syntax input-traffic-control-profile *profiler-name* **shared-instance** *instance-name*;

Hierarchy Level [edit class-of-service interfaces *interface-name* **unit** *logical-unit-number*]

Release Information Statement introduced in Junos OS Release 7.6.

Description For Gigabit Ethernet IQ2 and IQ2E PICs and Enhanced Queuing DPCs on MX Series routers, apply an input traffic scheduling and shaping profile to the logical interface.

Options ***profile-name***—Name of the traffic-control profile to be applied to this interface.

Required Privilege Level interface—To view this statement in the configuration.
interface-control—To add this statement to the configuration.

Related Documentation

- Configuring Traffic Control Profiles for Shared Scheduling and Shaping on page 334
- Configuring Ingress Hierarchical CoS on Enhanced Queuing DPCs on page 368
- **input-shaping-rate (Logical Interface) on page 507**
- **traffic-control-profiles on page 581**

input-traffic-control-profile-remaining

Syntax input-traffic-control-profile-remaining *profile-name*;

Hierarchy Level [edit class-of-service interfaces *interface-name*],
[edit class-of-service interfaces *interface-name* **interface-set** *interface-set-name*]

Release Information Statement introduced in Junos OS Release 9.0.

Description For Enhanced Queuing DPCs on MX Series routers and M Series and T Series routers with IQ2E PIC, apply an input traffic scheduling and shaping profile for remaining traffic to the logical interface or interface set.

Options ***profile-name***—Name of the traffic-control profile for remaining traffic to be applied to this interface or interface set.

Required Privilege Level interface—To view this statement in the configuration.
interface-control—To add this statement to the configuration.

Related Documentation

- Configuring Ingress Hierarchical CoS on Enhanced Queuing DPCs on page 368
- **input-traffic-control-profile on page 509**

interfaces

Syntax

```
interfaces {
  interface-name {
    input-scheduler-map map-name;
    input-shaping-rate rate;
    irb {
      unit logical-unit-number {
        classifiers {
          type (classifier-name | default);
        }
        rewrite-rules {
          dscp (rewrite-name | default);
          dscp-ipv6 (rewrite-name | default);
          exp (rewrite-name | default)protocol protocol-types;
          ieee-802.1 (rewrite-name | default) vlan-tag (outer | outer-and-inner);
          inet-precedence (rewrite-name | default);
        }
      }
    }
    member-link-scheduler (replicate | scale);
    scheduler-map map-name;
    scheduler-map-chassis map-name;
    shaping-rate rate;
    unit logical-unit-number {
      classifiers {
        type (classifier-name | default) family (mpls | inet);
      }
      forwarding-class class-name;
      fragmentation-map map-name;
      input-shaping-rate (percent percentage | rate);
      input-traffic-control-profile profiler-name shared-instance instance-name;
      output-traffic-control-profile profile-name shared-instance instance-name;
      per-session-scheduler;
      rewrite-rules {
        dscp (rewrite-name | default);
        dscp-ipv6 (rewrite-name | default);
        exp (rewrite-name | default) protocol protocol-types;
        exp-push-push-push default;
        exp-swap-push-push default;
        ieee-802.1 (rewrite-name | default) vlan-tag (outer | outer-and-inner);
        inet-precedence (rewrite-name | default);
      }
      scheduler-map map-name;
      shaping-rate rate;
      translation-table (to-dscp-from-dscp | to-dscp-ipv6-from-dscp-ipv6 | to-exp-from-exp
        | to-inet-precedence-from-inet-precedence) table-name;
    }
  }
  interface-set interface-set-name {
    excess-bandwith-share;
    internal-node;
    output-traffic-control-profile profile-name;
    output-traffic-control-profile-remaining profile-name;
```



```
    }
}
```

Hierarchy Level [edit class-of-service]

Release Information Statement introduced before Junos OS Release 7.4.
Interface-set level added in Junos OS Release 8.5.

Description Configure interface-specific CoS properties for incoming packets.

Options The remaining statements are explained separately.

Required Privilege Level interface—To view this statement in the configuration.
interface-control—To add this statement to the configuration.

Related Documentation

- Overview of BA Classifier Types on page 42
- Configuring Rewrite Rules on page 242

interface-set

Syntax

```
interface-set interface-set-name {
    excess-bandwith-share (proportional value | equal);
    internal-node;
    output-traffic-control-profile profile-name;
    output-traffic-control-profile-remaining profile-name;
}
```

Hierarchy Level [edit class-of-service **interfaces**]

Release Information Statement introduced in Junos OS Release 8.5.

Description For MX Series routers with Enhanced Queuing DPCs and M Series and T Series routers with IQ2E PIC, configure hierarchical schedulers.

Options ***interface-set-name***—Name of the interface set.

The remaining statements are explained separately.

Required Privilege Level interface—To view this statement in the configuration.
interface-control—To add this statement to the configuration.

Related Documentation

- Configuring Hierarchical Schedulers for CoS on page 207

internal-node

Syntax internal-node;

Hierarchy Level [edit class-of-service **interfaces interface-set** *interface-set-name*]

Release Information Statement introduced in Junos OS Release 8.5.

Description The statement is used 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.

Default If you do not include this statement, the node is internal only if its children have a traffic control profile configured.

Required Privilege Level interface—To view this statement in the configuration.
interface-control—To add this statement to the configuration.

Related Documentation

- Configuring Internal Scheduler Nodes on page 219

interpolate

Syntax

```
interpolate {
  drop-probability [ values ];
  fill-level [ values ];
}
```

Hierarchy Level [edit class-of-service **drop-profiles** *profile-name*]

Release Information Statement introduced before Junos OS Release 7.4.

Description Specify values for interpolating relationship between queue fill level and drop probability.

The statements are explained separately.

Usage Guidelines See "Configuring RED Drop Profiles" on page 233.

Required Privilege Level interface—To view this statement in the configuration.
interface-control—To add this statement to the configuration.

irb

Syntax

```
irb {
  unit logical-unit-number {
    classifiers {
      type (classifier-name | default);
    }
    rewrite-rules {
      dscp (rewrite-name | default);
      dscp-ipv6 (rewrite-name | default);
      exp (rewrite-name | default)protocol protocol-types;
      ieee-802.1 (rewrite-name | default) vlan-tag (outer | outer-and-inner);
      inet-precedence (rewrite-name | default);
    }
  }
}
```

Hierarchy Level [edit class-of-service **interfaces**]

Release Information Statement introduced in Junos OS Release 8.4.

Description On the MX Series routers, you can apply classifiers or rewrite rules to an integrated bridging and routing (IRB) interface. All types of classifiers and rewrite rules are allowed. These classifiers and rewrite rules are independent of others configured on the MX Series router.

The statements are explained separately.

Usage Guidelines See "MX Series Router CoS Hardware Capabilities and Limitations" on page 266.

Required Privilege Level interface—To view this statement in the configuration.
interface-control—To add this statement to the configuration.

layer2-policer

Syntax

```
layer2-policer {
  input-policer policer-name;
  input-three-color policer-name;
  output-policer policer-name;
  output-three-color policer-name;
}
```

Hierarchy Level [edit interfaces ge-*fpc/pic/port* unit *logical-unit-number*]

Release Information Statement introduced in Junos OS Release 8.2.

Description For Gigabit Ethernet interfaces only, apply an input or output policer at Layer 2. The policer must be properly defined at the **[edit firewall]** hierarchy level.

Options The statements are explained separately.

Usage Guidelines See “Applying Layer 2 Policers to Gigabit Ethernet Interfaces” on page 101.

Required Privilege Level interface—To view this statement in the configuration.
interface-control—To add this statement to the configuration.

linear-red-profile

Syntax linear-red-profile *profile-name*;

Hierarchy Level [edit interfaces at-*fpc/pic/port* atm-options scheduler-maps *map-name* **forwarding-class** *class-name*]

Release Information Statement introduced before Junos OS Release 7.4.

Description For ATM2 IQ interfaces only, assign a linear RED profile to a specified forwarding class. To define the linear RED profiles, include the **linear-red-profiles** statement at the **[edit interfaces at-*fpc/pic/port* atm-options]** hierarchy level.

Default If you do not include either the **epd-threshold** or the **linear-red-profile** statement in the forwarding class configuration, the Junos OS uses an EPD threshold based on the available bandwidth and other parameters.

Options ***profile-name***—Name of the linear RED profile.

Required Privilege Level interface—To view this statement in the configuration.
interface-control—To add this statement to the configuration.

Related Documentation

- Configuring Scheduler Maps on ATM Interfaces on page 416
- **epd-threshold on page 471**
- **linear-red-profiles on page 515**

linear-red-profiles

Syntax

linear-red-profiles *profile-name* {
  high-plp-threshold *percent*;
  low-plp-threshold *percent*;
  queue-depth *cells*;
}

Hierarchy Level [edit interfaces at-*fpc/pic/port* **atm-options**]

Release Information Statement introduced before Junos OS Release 7.4.

Description For ATM2 IQ interfaces only, define CoS virtual circuit drop profiles for RED. When a packet arrives, RED checks the queue fill level. If the fill level corresponds to a nonzero drop probability, the RED algorithm determines whether to drop the arriving packet.

Options ***profile-name***—Name of the drop profile.

The statements are explained separately.

Required Privilege Level interface—To view this statement in the configuration.
interface-control—To add this statement to the configuration.

Related Documentation

- Configuring Linear RED Profiles on ATM Interfaces on page 412

logical-bandwidth-policer

Syntax logical-bandwidth-policer;

Hierarchy Level [edit firewall **policer** *policer-name*]

Release Information Statement introduced in Junos OS Release 8.2.

Description Extend the policer rate limits to logical interfaces. The policer rate limit is based on the shaping rate defined on the logical interface.

Usage Guidelines See “Configuring Logical Bandwidth Policers” on page 81.

Required Privilege Level interface—To view this statement in the configuration.
interface-control—To add this statement to the configuration.

Related Documentation

- **shaping-rate on page 563**

logical-interface-policer

Syntax logical-interface-policer;

Hierarchy Level [edit firewall **three-color-policer** *policer-name*]

Release Information Statement introduced in Junos OS Release 8.2.

Description Apply a policer to a logical interface in the ingress or egress direction as part of a configuration using tricolor marking to discard high loss priority traffic.

Required Privilege Level interface—To view this statement in the configuration.
interface-control—To add this statement to the configuration.

Related Documentation

- Configuring Tricolor Marking Policers on page 98
- **action on page 443**

loss-priority

See the following sections:

loss-priority (BA Classifiers)

Syntax loss-priority *level*;

Hierarchy Level [edit class-of-service classifiers *type classifier-name* forwarding-class *class-name*]

Release Information Statement introduced before Junos OS Release 7.4.

Description Specify packet loss priority value for a specific set of code-point aliases and bit patterns.

Options *level* can be one of the following:

- **high**—Packet has high loss priority.
- **medium-high**—Packet has medium-high loss priority.
- **medium-low**—Packet has medium-low loss priority.
- **low**—Packet has low loss priority.

Required Privilege Level interface—To view this statement in the configuration.
interface-control—To add this statement to the configuration.

Related Documentation

- Overview of BA Classifier Types on page 42
- Example: Configuring CoS for a PBB Network on MX Series Routers
- Configuring Tricolor Marking on page 89

loss-priority (Normal Filter)

Syntax loss-priority (high | low);

Hierarchy Level [edit firewall family *family-name* filter *filter-name* term *term-name* then]

Release Information Statement introduced before Junos OS Release 7.4.

Description Set the loss priority of incoming packets.

Required Privilege Level interface—To view this statement in the configuration.
interface-control—To add this statement to the configuration.

Related Documentation

- Configuring Multifield Classifiers on page 72
- *Junos OS Policy Framework Configuration Guide*

loss-priority (Rewrite Rules)

Syntax loss-priority *level*;

Hierarchy Level [edit class-of-service rewrite-rules *type rewrite-name* forwarding-class *class-name*]

Release Information Statement introduced before Junos OS Release 7.4.

Description Specify a loss priority to which to apply a rewrite rule. The rewrite rule sets the code-point aliases and bit patterns for a specific forwarding class and packet loss priority (PLP). The inputs for the map are the forwarding class and the PLP. The output of the map is the code-point alias or bit pattern.

Options *level* can be one of the following:

- **high**—The rewrite rule applies to packets with high loss priority.
- **low**—The rewrite rule applies to packets with low loss priority.
- **medium-high**—(For J Series routers only) The rewrite rule applies to packets with medium-high loss priority.
- **medium-low**—(For J Series routers only) The rewrite rule applies to packets with medium-low loss priority.

Usage Guidelines See “Configuring Rewrite Rules” on page 242, “Overview of BA Classifier Types” on page 42, “Configuring Tricolor Marking” on page 89, and Example: Configuring CoS for a PBB Network on MX Series Routers.

Required Privilege Level interface—To view this statement in the configuration.
interface-control—To add this statement to the configuration.

loss-priority (Scheduler Drop Profiles)

Syntax loss-priority (any | low | medium-low | medium-high | high);

Hierarchy Level [edit class-of-service schedulers *scheduler-name* **drop-profile-map**]

Release Information Statement introduced before Junos OS Release 7.4.

Description Specify a loss priority to which to apply a drop profile. The drop profile map sets the drop profile for a specific PLP and protocol type. The inputs for the map are the PLP designation and the protocol type. The output is the drop profile.

Options **any**—The drop profile applies to packets with any PLP.

high—The drop profile applies to packets with high PLP.

medium—The drop profile applies to packets with medium PLP.

low—The drop profile applies to packets with low PLP.

Required Privilege Level interface—To view this statement in the configuration.
interface-control—To add this statement to the configuration.

Related Documentation

- Default Schedulers on page 147
- Configuring Tricolor Marking on page 89
- **protocol (Schedulers) on page 541**

loss-priority (Simple Filter)

Syntax loss-priority (high | low | medium);

Hierarchy Level [edit firewall family *family-name* simple-filter *filter-name* term *term-name* **then**]

Release Information Statement introduced in Junos OS Release 7.6.

Description Set the loss priority of incoming packets.

Required Privilege Level interface—To view this statement in the configuration.
interface-control—To add this statement to the configuration.

Related Documentation

- Configuring Multifield Classifiers on page 72
- *Junos OS Policy Framework Configuration Guide*

low-plp-max-threshold

Syntax low-plp-max-threshold *percent*;

Hierarchy Level [edit interfaces at-*fpc/pic/port* **atm-options linear-red-profiles** *profile-name*]

Release Information Statement introduced before Junos OS Release 7.4.

Description For ATM2 IQ interfaces only, define the drop profile fill-level for the low PLP CoS VC. When the fill level exceeds the defined percentage, all packets are dropped.

Options *percent*—Fill-level percentage when linear RED is applied to cells with PLP.

Required Privilege Level interface—To view this statement in the configuration.
interface-control—To add this statement to the configuration.

Related Documentation

- Configuring Linear RED Profiles on ATM Interfaces on page 412
- **high-plp-max-threshold on page 497**
- **low-plp-threshold on page 520**
- **queue-depth on page 543**

low-plp-threshold

Syntax low-plp-threshold *percent*;

Hierarchy Level [edit interfaces at-*fpc/pic/port* **atm-options linear-red-profiles** *profile-name*]

Release Information Statement introduced before Junos OS Release 7.4.

Description For ATM2 IQ interfaces only, define the CoS VC drop profile fill-level percentage when linear RED is applied to cells with low PLP. When the fill level exceeds the defined percentage, packets with low PLP are randomly dropped by RED. This statement is mandatory.

Options *percent*—Fill-level percentage when linear RED is applied to cells with low PLP.

Required Privilege Level interface—To view this statement in the configuration.
interface-control—To add this statement to the configuration.

Related Documentation

- Configuring Linear RED Profiles on ATM Interfaces on page 412
- **high-plp-max-threshold on page 497**
- **high-plp-threshold on page 498**
- **low-plp-max-threshold on page 520**
- **queue-depth on page 543**

lsp-next-hop

Syntax lsp-next-hop [*lsp-regular-expression*];

Hierarchy Level [edit class-of-service forwarding-policy next-hop-map *map-name* forwarding-class *class-name*]

Release Information Statement introduced before Junos OS Release 7.4.

Description Specify the LSP regular expression to which to map forwarded traffic.

Options ***lsp-regular-expression***—Next-hop LSP label.

Usage Guidelines See “Configuring CoS-Based Forwarding” on page 130.

Required Privilege Level interface—To view this statement in the configuration.
interface-control—To add this statement to the configuration.

match-direction

Syntax match-direction (input | output | input-output);

Hierarchy Level [edit services cos rule *rule-name*]

Release Information Statement introduced in Junos OS Release 8.1.

Description Specify the direction in which the rule match is applied.

Options **input**—Apply the rule match on the input side of the interface.

output—Apply the rule match on the output side of the interface.

input-output—Apply the rule match bidirectionally.

Required Privilege Level interface—To view this statement in the configuration.
interface-control—To add this statement to the configuration.

Related Documentation

- Configuring CoS Rules on page 282

max-queues-per-interface

Syntax max-queues-per-interface (4 | 8);

Hierarchy Level [edit chassis fpc *slot-number* pic *pic-number*],
[edit chassis lcc *number* fpc *slot-number* pic *pic-number*]

Release Information Statement introduced before Junos OS Release 7.4.
Support for TX Matrix and TX Matrix Plus added in Junos OS Release 9.6.
On MX Series routers, configure eight egress queues on Trio MPC/MIC interfaces.

Description On M320, T Series, and TX Matrix routers, configure eight egress queues on interfaces.

On MX Series routers, configure eight egress queues on Trio MPC/MIC interfaces.

Required Privilege Level interface—To view this statement in the configuration.
interface-control—To add this statement to the configuration.

Related Documentation

- Configuring Up to 16 Forwarding Classes on page 120
- Enabling Eight Queues on ATM Interfaces on page 418
- Configuring the Maximum Number of Queues for Trio MPC/MIC Interfaces on page 378

member-link-scheduler

Syntax member-link-scheduler (replicate | scale);

Hierarchy Level [edit class-of-service interfaces],
[edit logical-systems *logical-system-name* class-of-service interfaces *interface-name*],
[edit routing-instances *routing-instance-name* class-of-service interfaces *interface-name*]

Release Information Statement introduced in Junos OS Release 9.6.

Description Determines whether scheduler parameters for aggregated interface member links are applied in a replicated or scaled manner.

Default By default, scheduler parameters are scaled (in “equal division mode”) among aggregated interface member links.

Options **replicate**—Scheduler parameters are copied to each level of the aggregated interface member links.

scale—Scheduler parameters are scaled based on number of member links and applied each level of the aggregated interface member links.

Usage Guidelines See “Configuring Hierarchical Schedulers for CoS” on page 207.

Required Privilege Level view-level—To view this statement in the configuration.
control-level—To add this statement to the configuration.

mode

Syntax mode *session-shaping*;

Hierarchy Level [edit chassis fpc *slot-number* pic *pic-number* **traffic-manager**],
[edit chassis lcc *number* fpc *slot-number* pic *pic-number* **traffic-manager**]

Release Information Statement introduced in Junos OS Release 8.3.

Description Enable shaping on an L2TP session.

Options ***session-shaping***—Perform shaping instead of policing on this interface.

Required Privilege Level interface—To view this statement in the configuration.
interface-control—To add this statement to the configuration.

Related Documentation

- Configuring CoS for L2TP Tunnels on ATM Interfaces on page 426
- **ingress-shaping-overhead on page 505**

multilink-class

Syntax multilink-class *number*;

Hierarchy Level [edit class-of-service **fragmentation-maps** *map-name* **forwarding-class** *class-name*]

Release Information Statement introduced before Junos OS Release 7.4.

Description For AS PIC link services IQ interfaces (**lsq**) only, map a forwarding class into a multiclass MLPPP (MCML).

The **multilink-class** statement and **no-fragmentation** statements are mutually exclusive.

Options ***number***—The multilink class assigned to this forwarding class.

Range: 0 through 7

Default: None

Usage Guidelines See "Configuring Fragmentation by Forwarding Class" on page 140 and the *Junos OS Services Interfaces Configuration Guide*.

Required Privilege Level interface—To view this statement in the configuration.
interface-control—To add this statement to the configuration.

next-hop

Syntax next-hop [*next-hop-name*];

Hierarchy Level [edit class-of-service forwarding-policy next-hop-map *map-name* forwarding-class *class-name*]

Release Information Statement introduced before Junos OS Release 7.4.

Description Specify the next-hop name or address to which to map forwarded traffic.

Options ***next-hop-name***—Next-hop alias or IP address.

Usage Guidelines See “Configuring CoS-Based Forwarding” on page 130.

Required Privilege Level interface—To view this statement in the configuration.
interface-control—To add this statement to the configuration.

next-hop-map

Syntax

```
next-hop-map map-name {
  forwarding-class class-name {
    next-hop next-hop-name;
    lsp-next-hop [ lsp-regular-expression ];
    non-lsp-next-hop;
    discard;
  }
}
```

Hierarchy Level [edit class-of-service forwarding-policy]

Release Information Statement introduced before Junos OS Release 7.4.

Description Specify the map for CoS forwarding routes.

Options ***map-name***—Map that defines next-hop routes.

Usage Guidelines See “Configuring CoS-Based Forwarding” on page 130.

Required Privilege Level interface—To view this statement in the configuration.
interface-control—To add this statement to the configuration.

no-fragmentation

Syntax no-fragmentation;

Hierarchy Level [edit class-of-service **fragmentation-maps** *map-name* **forwarding-class** *class-name*]

Release Information Statement introduced before Junos OS Release 7.4.

Description For AS PIC link services IQ (**lsq**) interfaces only, set traffic on a queue to be interleaved, rather than fragmented. This statement specifies that no extra fragmentation header is prepended to the packets received on this queue and that static-link load balancing is used to ensure in-order packet delivery.

Static-link load balancing is done based on packet payload. For IPv4 and IPv6 traffic, the link is chosen based on a hash computed from the source address, destination address, and protocol. If the IP payload is TCP or UDP traffic, the hash also includes the source port and destination port. For MPLS traffic, the hash includes all MPLS labels and fields in the payload, if the MPLS payload is IPv4 or IPv6.

Default If you do not include this statement, the traffic in forwarding class ***class-name*** is fragmented.

Usage Guidelines See “Configuring Fragmentation by Forwarding Class” on page 140 and the *Junos OS Services Interfaces Configuration Guide*.

Required Privilege Level interface—To view this statement in the configuration.
interface-control—To add this statement to the configuration.

non-lsp-next-hop

Syntax non-lsp-next-hop;

Hierarchy Level [edit class-of-service **forwarding-policy next-hop-map** *map-name* **forwarding-class** *class-name*]

Release Information Statement introduced before Junos OS Release 9.0.

Description Use a non-LSP next hop for traffic sent to this forwarding class next-hop map of this forwarding policy.

Usage Guidelines See “Configuring CoS-Based Forwarding” on page 130.

Required Privilege Level interface—To view this statement in the configuration.
interface-control—To add this statement to the configuration.

output-forwarding-class-map

Syntax output-forwarding-class-map *forwarding-class-map-name*;

Hierarchy Level [edit **class-of-service forwarding-classes-interface-specific**]

Release Information Statement introduced in Junos OS Release 9.6.

Description Apply a configured forwarding class map to a logical interface.

Options *forwarding-class-map-name*—Name of a forwarding class mapping configured at the **[edit class-of-service forwarding-classes-interface-specific]** hierarchy level.

Usage Guidelines “Classifying Packets by Egress Interface” on page 116

Required Privilege Level interface-control—To add this statement to the configuration.

Related Documentation

- **forwarding-classes-interface-specific on page 491**

output-policer

Syntax output-policer *policer-name*;

Hierarchy Level [edit interfaces *interface-name* unit *logical-unit-number* **layer2-policer**]

Release Information Statement introduced in Junos OS Release 8.2.

Description Associate a Layer 2 policer with a logical interface. The **output-policer** and **output-three-color** statements are mutually exclusive.

Options *policer-name*—Name of the policer that you define at the **[edit firewall]** hierarchy level.

Usage Guidelines See “Applying Layer 2 Policers to Gigabit Ethernet Interfaces” on page 101.

Required Privilege Level interface—To view this statement in the configuration.
interface-control—To add this statement to the configuration.

Related Documentation

- **input-policer on page 506**

output-three-color

Syntax output-three-color *policer-name*;

Hierarchy Level [edit interfaces *interface-name* unit *logical-unit-number* **layer2-policer**]

Release Information Statement introduced in Junos OS Release 8.2.

Description Associate a Layer 2, three-color policer with a logical interface. The **output-three-color** and **output-policer** statements are mutually exclusive.

Options *policer-name*—Name of the three-color policer that you define at the **[edit firewall]** hierarchy level.

Usage Guidelines See “Applying Layer 2 Policers to Gigabit Ethernet Interfaces” on page 101.

Required Privilege Level interface—To view this statement in the configuration.
interface-control—To add this statement to the configuration.

Related Documentation

- **input-three-color on page 508**

output-traffic-control-profile

Syntax output-traffic-control-profile *profile-name* **shared-instance** *instance-name*;

Hierarchy Level [edit class-of-service interfaces *interface-name* unit *logical-unit-number*],
[edit class-of-service interfaces *interface-name* **interface-set** *interface-set-name*]

Release Information Statement introduced in Junos OS Release 7.6.
Interface set option for Enhanced Queuing DPCs on MX Series routers introduced in Junos OS Release 8.5

Description For Channelized IQ PICs, Gigabit Ethernet IQ, Gigabit Ethernet IQ2, and IQ2E PICs, link services IQ (LSQ) interfaces on AS PICs, and Enhanced Queuing DPCs on MX Series routers, apply an output traffic scheduling and shaping profile to the logical interface.

The **shared-instance** statement is supported on Gigabit Ethernet IQ2 PICs only.

Options ***profile-name***—Name of the traffic-control profile to be applied to this interface

Required Privilege Level interface—To view this statement in the configuration.
interface-control—To add this statement to the configuration.

Related Documentation

- Oversubscribing Interface Bandwidth on page 184
- Configuring Traffic Control Profiles for Shared Scheduling and Shaping on page 334
- Example: Configuring CoS for a PBB Network on MX Series Routers
- Configuring Hierarchical Schedulers for CoS on page 207 (Enhanced Queuing DPCs on MX Series routers)
- **output-traffic-control-profile-remaining on page 529**
- **traffic-control-profiles on page 581**

output-traffic-control-profile-remaining

Syntax output-traffic-control-profile-remaining *profile-name*;

Hierarchy Level [edit class-of-service interfaces *interface-name*],
[edit class-of-service interfaces *interface-name* **interface-set** *interface-set-name*]

Release Information Statement introduced in Junos OS Release 8.5.

Description For Enhanced Queuing DPCs on MX Series routers and IQ2E PIC on M Series and T Series routers, apply an output traffic scheduling and shaping profile for remaining traffic to the logical interface or interface set.

Options ***profile-name***—Name of the traffic-control profile for remaining traffic to be applied to this interface or interface set.

Required Privilege Level interface—To view this statement in the configuration.
interface-control—To add this statement to the configuration.

Related Documentation

- Configuring Hierarchical Schedulers for CoS on page 207
- Configuring Remaining Common Queues on Trio MPC/MIC Interfaces on page 379
- **output-traffic-control-profile on page 528**

overhead-accounting

Syntax overhead-accounting (frame-mode | cell-mode) <bytes *byte-value*>;

Hierarchy Level [edit class-of-service **traffic-control-profiles** *profile-name*]

Release Information Statement introduced in Junos OS Release 10.2.

Description Configure the mode to shape downstream ATM traffic based on either frames or cells.

Default The default is **frame-mode**.

Options **frame-mode**—Shaping based on the number of bytes in the frame, without regard to cell encapsulation or padding overhead.

cell-mode—Shaping based on the number of bytes in cells, and accounts for the ATM cell encapsulation and padding overhead. The resulting traffic stream conforms to the policing rates configured in downstream ATM switches, reducing the number of packet drops in the Ethernet network

byte-value—Byte adjustment value for frame or cell shaping mode.

Range: –120 through 124 bytes

Required Privilege Level interface—To view this statement in the configuration.
interface-control—To add this statement to the configuration.

Related Documentation

- Configuring Static Shaping Parameters to Account for Overhead in Downstream Traffic Rates on page 393
- Bandwidth Management for Downstream Traffic in Edge Networks Overview on page 391
- **egress-shaping-overhead on page 470**

per-session-scheduler

Syntax per-session-scheduler;

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

Release Information Statement introduced in Junos OS Release 8.3.

Description Enable session-aware CoS shaping on this L2TP interface.

Required Privilege Level interface—To view this statement in the configuration.
interface-control—To add this statement to the configuration.

Related Documentation

- Configuring CoS for L2TP Tunnels on ATM Interfaces on page 426
- **ingress-shaping-overhead on page 505**
- **mode on page 523**

per-unit-scheduler

Syntax per-unit-scheduler;

Hierarchy Level [edit interfaces *interface-name*]

Release Information Statement introduced before Junos OS Release 7.4.

Description Enable multiple queues for each logical interface. When this statement is included, you can associate an output scheduler with each logical interface. This statement and the **shared-scheduler** statement are mutually exclusive.

Usage Guidelines See “Applying Scheduler Maps Overview” on page 167.

Required Privilege Level interface—To view this statement in the configuration.
interface-control—To add this statement to the configuration.

plp-copy-all

Syntax plp-copy-all;

Hierarchy Level [edit class-of-service]

Release Information Statement introduced in Junos OS Release 10.3

Description On T Series routers with different Packet Forwarding Engines (non-Enhanced Scaling and Enhanced Scaling FPCs), enables PLP bit copying for ingress and egress unicast and multicast traffic.

Required Privilege Level interface—To view this statement in the configuration.
interface-control—To add this statement to the configuration.

Related Documentation

- Packet Loss Priority Configuration Overview on page 234
- Setting Packet Loss Priority on page 60

plp-to-clp

Syntax plp-to-clp;

Hierarchy Level [edit interfaces at-*fpc/pic/port* **atm-options**],
[edit interfaces at-*fpc/pic/port* **unit** *logical-unit-number*],
[edit logical-systems *logical-system-name* interfaces at-*fpc/pic/port* **unit** *logical-unit-number*]

Release Information Statement introduced before Junos OS Release 7.4.

Description For ATM2 IQ interfaces only, enable the PLP setting to be copied to the cell loss priority (CLP) bit.

Default If you omit this statement, the Junos OS does not copy the PLP setting to the CLP bit.

Required Privilege Level interface—To view this statement in the configuration.
interface-control—To add this statement to the configuration.

Related Documentation

- Copying the PLP Setting to the CLP Bit on ATM Interfaces on page 424

policer

See the following sections:

- **policer (Applying to an Interface) on page 533**
- **policer (Configuring) on page 534**

policer (Applying to an Interface)

Syntax

```
policer {
  input policer-name;
  output policer-name;
}
```

Hierarchy Level [edit interfaces *interface-name* unit *logical-unit-number* **family** *family*],
[edit logical-systems *logical-system-name* interfaces *interface-name* unit *logical-unit-number* **family** *family*]

Release Information Statement introduced before Junos OS Release 7.4.

Description Apply a rate policer to an interface.

Options **input** ***policer-name***—Name of one policer to evaluate when packets are received on the interface.

output ***policer-name***—Name of one policer to evaluate when packets are transmitted on the interface.

Usage Guidelines See “Configuring Multifield Classifiers” on page 72, “Using Multifield Classifiers to Set PLP” on page 103, and “Default Schedulers” on page 147; for a general discussion of this statement, see the *Junos OS Network Interfaces Configuration Guide*.

Required Privilege Level interface—To view this statement in the configuration.
interface-control—To add this statement to the configuration.

Related Documentation

- **simple-filter on page 572**

policer (Configuring)

Syntax

```
policer policer-name {
  logical-bandwidth-policer;
  if-exceeding {
    bandwidth-limit rate;
    bandwidth-percent number;
    burst-size-limit bytes;
  }
  then {
    policer-action;
  }
}
```

Hierarchy Level [edit firewall]

Release Information Statement introduced before Junos OS Release 7.4.
The **out-of-profile** policer action added in Junos OS Release 8.1.
The **logical-bandwidth-policer** statement added in Junos OS Release 8.2.

Description Configure policer rate limits and actions. To activate a policer, you must include the **policer** action modifier in the **then** statement in a firewall filter term or on an interface.

Options ***policer-action***—One or more actions to take:

- **discard**—Discard traffic that exceeds the rate limits.
- **forwarding-class *class-name***—Specify the particular forwarding class.
- **loss-priority**—Set the packet loss priority (PLP) to low or high.
- **out-of-profile**—On J Series routers with strict priority queuing, prevent starvation of other queues by rate limiting the data stream entering the strict priority queue, marking the packets that exceed the rate limit as out-of-profile, and dropping the out-of-profile packets if the physical interface is congested.

policer-name—Name that identifies the policer. The name can contain letters, numbers, and hyphens (-), and can be up to 255 characters long. To include spaces in the name, enclose it in quotation marks (“ ”).

then—Actions to take on matching packets.

The remaining statements are explained separately.

Required Privilege Level firewall—To view this statement in the configuration.
firewall-control—To add this statement to the configuration.

Related Documentation

- Configuring Multifield Classifiers on page 72
- Using Multifield Classifiers to Set PLP on page 103
- Default Schedulers on page 147
- *Junos OS Policy Framework Configuration Guide*

- **filter (Configuring) on page 483**
- **priority (Schedulers) on page 539**

priority

See the following sections:

- **priority (ATM2 IQ Schedulers) on page 536**
- **priority (Fabric Queues, Schedulers) on page 537**
- **priority (Fabric Priority) on page 538**
- **priority (Schedulers) on page 539**

priority (ATM2 IQ Schedulers)

Syntax priority (high | low);

Hierarchy Level [edit interfaces at-*fpc/pic/port* atm-options scheduler-maps *map-name* **forwarding-class** *class-name*]

Release Information Statement introduced before Junos OS Release 7.4.

Description For ATM2 IQ interfaces only, assign queuing priority to a forwarding class.

Options **low**—Forwarding class has low priority.

high—Forwarding class has high priority.

Required Privilege Level interface—To view this statement in the configuration.
interface-control—To add this statement to the configuration.

Related Documentation

- Configuring Scheduler Maps on ATM Interfaces on page 416

priority (Fabric Queues, Schedulers)

Syntax priority (high | low)**scheduler** *scheduler-name*;

Hierarchy Level [edit class-of-service **fabric scheduler-map**]

Release Information Statement introduced before Junos OS Release 7.4.

Description For M320, MX Series, and T Series routers only, specify the fabric priority with which a scheduler is associated.

For a scheduler that you associate with a fabric priority, you cannot include the **buffer-size**, **transmit-rate**, or **priority** statements at the **[edit class-of-service schedulers *scheduler-name*]** hierarchy level.

Options **low**—Scheduler has low priority.

high—Scheduler has high priority.

The remaining statements are explained separately.

Usage Guidelines See “Associating Schedulers with Fabric Priorities” on page 200.

Required Privilege Level interface—To view this statement in the configuration.
interface-control—To add this statement to the configuration.

priority (Fabric Priority)

Syntax priority (high | low);

Hierarchy Level [edit class-of-service **forwarding-classes class** *class-name* queue-num *queue-number*],
[edit class-of-service **forwarding-classes queue** *queue-number class-name*]

Release Information Statement introduced before Junos OS Release 7.4.
[edit class-of-service forwarding-classes class *class-name* queue-num *queue-number*] hierarchy level added in Junos OS Release 8.1.

Description For M320 routers, MX Series routers, and T Series routers only, specify a fabric priority value.

The two hierarchy levels are mutually exclusive. To configure up to eight forwarding classes with one-to-one mapping between forwarding classes and output queues, include this statement at the **[edit class-of-service forwarding-classes queue *queue-number class-name*]** hierarchy level. To configure up to 16 forwarding classes with multiple forwarding classes mapped to single queues, include this statement at the **[edit class-of-service forwarding-classes class *class-name* queue-num *queue-number*]** hierarchy level.

Options **low**—Forwarding class's fabric queuing has low priority.

high—Forwarding class's fabric queuing has high priority.

Usage Guidelines See "Overriding Fabric Priority Queuing" on page 120 and "Configuring Up to 16 Forwarding Classes" on page 120.

Required Privilege Level interface—To view this statement in the configuration.
interface-control—To add this statement to the configuration.

priority (Schedulers)

Syntax priority *priority-level*;

Hierarchy Level [edit class-of-service **schedulers** *scheduler-name*]

Release Information Statement introduced before Junos OS Release 7.4.

Description Specify packet-scheduling priority value.

Options ***priority-level*** can be one of the following:

- **low**—Scheduler has low priority.
- **medium-low**—Scheduler has medium-low priority.
- **medium-high**—Scheduler has medium-high priority.
- **high**—Scheduler has high priority. Assigning high priority to a queue prevents the queue from being underserved.
- **strict-high**—Scheduler has strictly high priority. Configure a **high** priority queue with unlimited transmission bandwidth available to it. As long as it has traffic to send, the **strict-high** priority queue receives precedence over **low**, **medium-low**, and **medium-high** priority queues, but not **high** priority queues. You can configure **strict-high** priority on only one queue per interface.

Required Privilege Level interface—To view this statement in the configuration.
interface-control—To add this statement to the configuration.

Related Documentation

- Configuring Schedulers for Priority Scheduling on page 165

protocol

See the following sections:

- **protocol (Rewrite Rules) on page 540**
- **protocol (Schedulers) on page 541**

protocol (Rewrite Rules)

Syntax protocol *protocol-types*;

Hierarchy Level [edit class-of-service interfaces *interface-name* unit *logical-unit-number* rewrite-rules exp *rewrite-name*],
[edit class-of-service interfaces *interface-name* unit *logical-unit-number* rewrite-rules dscp *rewrite-name*],
[edit class-of-service interfaces *interface-name* unit *logical-unit-number* rewrite-rules inet-prec *rewrite-name*]

Release Information Statement introduced before Junos OS Release 7.4.
Option for **dscp** and **inet-prec** introduced in Junos OS Release 8.4.

Description Apply a rewrite rule to MPLS packets only, and write the CoS value to MPLS headers only; or apply a rewrite rule to MPLS and IPv4 packets, and write the CoS value to MPLS and IPv4 headers.

Options *protocol-types* can be one of the following:

- **mpls**—Apply a rewrite rule to MPLS packets and write the CoS value to MPLS headers.
- **mpls-inet-both**—Apply a rewrite rule to VPN MPLS packets with IPv4 payloads. On M120, M320, MX Series, and T Series routers, write the CoS value to the MPLS and IPv4 headers. On M Series routers, initialize all ingress MPLS LSP packets with IPv4 payloads with 000 code points for the MPLS EXP value, and the configured rewrite code point for IP precedence.
- **mpls-inet-both-non-vpn**—Apply a rewrite rule to non-VPN MPLS packets with IPv4 payloads. On M120, M320, MX Series, and T Series routers, write the CoS value to the MPLS and IPv4 headers. On M Series routers, initialize all ingress MPLS LSP packets with IPv4 payloads with 000 code points for the MPLS EXP value, and the configured rewrite code point for IP precedence.

Required Privilege Level interface—To view this statement in the configuration.
interface-control—To add this statement to the configuration.

Related Documentation
- Rewriting MPLS and IPv4 Packet Headers on page 248

protocol (Schedulers)

Syntax protocol (any | non-tcp | tcp);

Hierarchy Level [edit class-of-service **schedulers** *scheduler-name* **drop-profile-map**]

Release Information Statement introduced before Junos OS Release 7.4.

Description Specify the protocol type for the specified scheduler.

Options **any**—Accept any protocol type.

non-tcp—Accept any protocol type other than TCP/IP.

any—Accept TCP/IP protocol type.

NOTE: On MX Series routers, you can only configure the any option.

Required Privilege Level interface—To view this statement in the configuration.
interface-control—To add this statement to the configuration.

Related Documentation

- Configuring Schedulers on page 148

q-pic-large-buffer

Syntax

```
q-pic-large-buffer {
  [ large-scale | small-scale ];
}
```

Hierarchy Level [edit chassis fpc *slot-number* pic *pic-number*],
[edit chassis llc *number* fpc *slot-number* pic *pic-number*]

Release Information Statement introduced in Junos OS Release 7.4.
Support for TX Matrix and TX Matrix Plus hierarchy added in Junos OS Release 9.6.

Description Enable configuration of large delay buffer size for slower interfaces (T1, E1, and *N*xDS0 interfaces configured on channelized IQ PICs).

Options ***large-scale***—Supports a large number of interfaces.

small-scale—Supports a small number of interfaces.

Default: ***small-scale***

Usage Guidelines See “Configuring Schedulers” on page 148.

Required Privilege Level interface—To view this statement in the configuration.
interface-control—To add this statement to the configuration.

queue

See the following sections:

- **queue (Global Queues) on page 542**
- **queue (Restricted Queues) on page 543**

queue (Global Queues)

Syntax queue *queue-number class-name*;

Hierarchy Level [edit class-of-service **forwarding-classes**]

Release Information Statement introduced before Junos OS Release 7.4.

Description Specify the output transmission queue to which to map all input from an associated forwarding class.

On M120, M320, MX Series, and T Series routers, this statement enables you to configure up to eight forwarding classes with one-to-one mapping to output queues. If you want to configure up to 16 forwarding classes with multiple forwarding classes mapped to single output queues, include the **class** statement instead of the **queue** statement at the **[edit class-of-service forwarding-classes]** hierarchy level.

Options *class-name*—Name of forwarding class.

queue-number—Output queue number.

Range: For M Series routers routers, 0 through 3. For M120, M320, MX Series, and T Series routers, 0 through 7. Some T Series router PICs are restricted to 0 through 3.

Usage Guidelines See "Configuring Forwarding Classes" on page 115.

Required Privilege Level interface—To view this statement in the configuration.
interface-control—To add this statement to the configuration.

Related Documentation

- **class (Forwarding Classes) on page 452**

queue (Restricted Queues)

Syntax queue *queue-number*;

Hierarchy Level [edit class-of-service **restricted-queues forwarding-class** *class-name*]

Release Information Statement introduced before Junos OS Release 7.4.

Description For M320, MX Series, and T Series routers only, map forwarding classes to restricted queues.

Options ***queue-number***—Output queue number.

Range: 0 through 3.

Required Privilege Level interface—To view this statement in the configuration.
interface-control—To add this statement to the configuration.

queue-depth

Syntax queue-depth *cells*;

Hierarchy Level [edit interfaces *interface-name* **atm-options linear-red-profiles** *profile-name*]

Release Information Statement introduced before Junos OS Release 7.4.

Description For ATM2 IQ interfaces only, define maximum queue depth in the CoS VC drop profile. Packets are always dropped beyond the defined maximum. This statement is mandatory; there is no default configuration.

Options ***cells***—Maximum number of cells the queue can contain.

Range: 1 through 64,000 cells

Required Privilege Level interface—To view this statement in the configuration.
interface-control—To add this statement to the configuration.

Related Documentation

- Configuring Linear RED Profiles on ATM Interfaces on page 412
- **high-plp-threshold on page 498**
- **low-plp-threshold on page 520**

red-buffer-occupancy

Syntax

```
red-buffer-occupancy {
  weighted-averaged [ instant-usage-weight-exponent ] weight-value;
}
```

Hierarchy Level [edit chassis fpc *slot-number* pic *pic-number*],
[edit chassis lcc *number* fpc *slot-number* pic *pic-number*]

Release Information Statement introduced in Junos OS Release 8.3.

Description Configure weighted RED (WRED) buffer occupancy on an IQ-PIC.

Options **instant-usage-weight-exponent** ***weight-value***—Establish an exponent to use for weighted average calculations of buffer occupancy.

weighted-averaged ***weight-value***—Establish a value to use for weighted average calculations of buffer occupancy.

Range: For IQ-PICs, 0 through 31. Values in excess of 31 are configurable, and appear in **show** commands, but are replaced with the *operational* maximum value of 31 on IQ-PICs.

Usage Guidelines See "Configuring Weighted RED Buffer Occupancy" on page 236.

Required Privilege Level interface—To view this statement in the configuration.
interface-control—To add this statement to the configuration.

(reflexive | reverse)

Syntax

```
(reflexive | reverse) {
  application-profile profile-name;
  dscp (alias | bits);
  forwarding-class class-name;
  syslog;
}
```

Hierarchy Level [edit **services** cos rule *rule-name* term *term-name* **then**]

Release Information Statement introduced in Junos OS Release 8.1.

Description **reflexive**—Applies the equivalent reverse CoS action to flows in the opposite direction.

reverse—Allows you to define CoS behavior for flows in the reverse direction.

The remaining statements are explained separately.

Required Privilege Level interface—To view this statement in the configuration.
interface-control—To add this statement to the configuration.

Related Documentation

- Configuring CoS Rules on page 282

restricted-queues

Syntax

restricted-queues {
  forwarding-class *class-name* **queue** *queue-number*;
}

Hierarchy Level

[edit class-of-service]

Release Information

Statement introduced before Junos OS Release 7.4.

Description

For M320, MX Series, and T Series routers only, map forwarding classes to restricted queues.

The remaining statements are explained separately.

Required Privilege Level

interface—To view this statement in the configuration.
interface-control—To add this statement to the configuration.

rewrite-rules

See the following sections:

- **rewrite-rules (Definition) on page 546**
- **rewrite-rules (Interfaces) on page 547**

rewrite-rules (Definition)

Syntax

```
rewrite-rules {
  type rewrite-name{
    import (rewrite-name | default);
    forwarding-class class-name {
      loss-priority level code-point [ aliases ] [ bit-patterns ];
    }
  }
}
```

Hierarchy Level [edit class-of-service]

Release Information Statement introduced before Junos OS Release 7.4.
ieee-802.1ad option introduced in Junos OS Release 9.2.

Description Specify a rewrite-rules mapping for the traffic that passes through all queues on the interface.

Options ***rewrite-name***—Name of a **rewrite-rules** mapping.

type—Traffic type.

Values: **dscp**, **dscp-ipv6**, **exp**, **frame-relay-de** (J Series routers only), **ieee-802.1**, **ieee-802.1ad**, **inet-precedence**

The remaining statements are explained separately.

Required Privilege Level interface—To view this statement in the configuration.
interface-control—To add this statement to the configuration.

Related Documentation

- Configuring Rewrite Rules on page 242
- Example: Configuring CoS for a PBB Network on MX Series Routers
- J Series router documentation

rewrite-rules (Interfaces)

Syntax

rewrite-rules {
  dscp (*rewrite-name* | default);
  dscp-ipv6 (*rewrite-name* | default);
  exp (*rewrite-name* | default)**protocol** *protocol-types*;
  exp-push-push-push default;
  exp-swap-push-push default;
  ieee-802.1 (*rewrite-name* | default) **vlan-tag** (outer | outer-and-inner);
  ieee-802.1ad (*rewrite-name* | default) **vlan-tag** (outer | outer-and-inner);
  inet-precedence (*rewrite-name* | default);
}

Hierarchy Level [edit class-of-service interfaces *interface-name* **unit** *logical-unit-number*]

Release Information Statement introduced before Junos OS Release 7.4.

Description Associate a rewrite-rules configuration or default mapping with a specific interface. On a MX Series router, **exp-push-push-push**, **exp-swap-push-push**, and **frame-relay-de** are not supported on an integrated bridging and routing (IRB) interface.

Options *rewrite-name*—Name of a **rewrite-rules** mapping configured at the **[edit class-of-service rewrite-rules]** hierarchy level.

default—The default mapping.

The remaining statements are explained separately.

Required Privilege Level interface—To view this statement in the configuration.
interface-control—To add this statement to the configuration.

Related Documentation

- Configuring Rewrite Rules on page 242
- J Series router documentation
- **rewrite-rules (Definition) on page 546**

routing-instances

Syntax

```
routing-instances routing-instance-name {
  classifiers {
    exp (classifier-name | default);
  }
}
```

Hierarchy Level [edit class-of-service]

Release Information Statement introduced before Junos OS Release 7.4.

Description For routing instances with VRF table labels enabled, apply a custom MPLS EXP classifier to the routing instance. You can apply the default MPLS EXP classifier or one that is previously defined.

Default If you do not include this statement, the default MPLS EXP classifier is applied to the routing instance.

Options ***routing-instance-name***—Name of a routing instance.

classifier-name—Name of the MPLS EXP classifier.

Usage Guidelines See “Applying MPLS EXP Classifiers to Routing Instances” on page 56.

Required Privilege Level interface—To view this statement in the configuration.
interface-control—To add this statement to the configuration.

rtvbr

Syntax rtvbr peak *rate* sustained *rate* burst *length*;

Hierarchy Level [edit interfaces *interface-name* atm-options vpi *vpi-identifier* **shaping**],
[edit interfaces *interface-name* unit *logical-unit-number* address *address* family *family* multipoint-destination *address* **shaping**],
[edit interfaces *interface-name* unit *logical-unit-number* **shaping**],
[edit logical-systems *logical-system-name* interfaces *interface-name* **unit** *logical-unit-number* **shaping**],
[edit logical-systems *logical-system-name* interfaces *interface-name* unit *logical-unit-number* address *address* family *family* multipoint-destination *address* **shaping**]

Release Information Statement introduced before Junos OS Release 7.4.

Description For ATM2 IQ PICs only, define the real-time variable bandwidth utilization in the traffic-shaping profile.

When you configure the real-time bandwidth utilization, you must specify all three options (**burst**, **peak**, and **sustained**). You can specify ***rate*** in bits per second either as a complete decimal number or as a decimal number followed by the abbreviation **k** (1000), **m** (1,000,000), or **g** (1,000,000,000). You can also specify ***rate*** in cells per second by entering a decimal number followed by the abbreviation **c**; values expressed in cells per second are converted to bits per second using the formula 1 cps = 384 bps.

Default If the **rtvbr** statement is not included, bandwidth utilization is unlimited.

Options **burst *length***—Burst length, in cells. If you set the length to 1, the peak traffic rate is used.

Range: 1 through 4000 cells

peak *rate*—Peak rate, in bits per second or cells per second.

Range: For ATM2 IQ OC3 and OC12 interfaces, 33 Kbps through 542,526,792 bps.For ATM2 IQ OC48 interfaces, 33 Kbps through 2,170,107,168 bps. For ATM2 IQ DS3 and E3 interfaces, 33 Kbps through the maximum rate, which depends on the ATM encapsulation and framing you configure. For more information, see the *Junos OS Network Interfaces Configuration Guide*.

sustained *rate*—Sustained rate, in bits per second or cells per second.

Range: For ATM2 IQ OC3 and OC12 interfaces, 33 Kbps through 542,526,792 bps. For ATM2 IQ OC48 interfaces, 33 Kbps through 2,170,107,168 bps. For ATM2 IQ DS3 and E3 interfaces, from 33 Kbps through the maximum rate, which depends on the ATM encapsulation and framing you configure. For more information, see the *Junos OS Network Interfaces Configuration Guide*.

Required Privilege Level interface—To view this statement in the configuration.
interface-control—To add this statement to the configuration.

Related Documentation

- Applying Scheduler Maps to Logical ATM Interfaces on page 424
- **cbr on page 450**

• **vbr on page 587**

rule

Syntax

```
rule rule-name {
  match-direction (input | output | input-output);
  term term-name {
    from {
      applications [ application-names ];
      application-sets [ set-names ];
      destination-address address;
      source-address address;
    }
    then {
      application-profile profile-name;
      dscp (alias | bits);
      forwarding-class class-name;
      syslog;
      (reflexive | reverse) {
        application-profile profile-name;
        dscp (alias | bits);
        forwarding-class class-name;
        syslog;
      }
    }
  }
}
```

Hierarchy Level [edit services cos],
[edit services cos rule-set *rule-set-name*]

Release Information Statement introduced in Junos OS Release 8.1.

Description Specify the rule the router uses when applying this service.

Options ***rule-name***—Identifier for the collection of terms that constitute this rule.

The remaining statements are explained separately.

Required Privilege Level interface—To view this statement in the configuration.
interface-control—To add this statement to the configuration.

Related Documentation

• Configuring CoS Rules on page 282

rule-set

Syntax

```
rule-set rule-set-name {
  [ rule rule-name ];
}
```

Hierarchy Level [edit services cos]

Release Information Statement introduced in Junos OS Release 8.1.

Description Specify the rule set the router uses when applying this service.

Options ***rule-set-name***—Identifier for the collection of rules that constitute this rule set.

Required Privilege Level interface—To view this statement in the configuration.
interface-control—To add this statement to the configuration.

Related Documentation

- Configuring CoS Rule Sets on page 286

scheduler

See the following sections:

- **scheduler (Fabric Queues) on page 552**
- **scheduler (Scheduler Map) on page 552**

scheduler (Fabric Queues)

Syntax scheduler *scheduler-name*;

Hierarchy Level [edit class-of-service **fabric scheduler-map priority** (high | low)]

Release Information Statement introduced before Junos OS Release 7.4.

Description For M320, MX Series, and T Series routers only, specify a scheduler to associate with a fabric queue. For fabric CoS configuration, schedulers are restricted to transmit rates and drop profiles.

Options ***scheduler-name***—Name of the scheduler configuration block.

Usage Guidelines See "Associating Schedulers with Fabric Priorities" on page 200.

Required Privilege Level interface—To view this statement in the configuration.
interface-control—To add this statement to the configuration.

scheduler (Scheduler Map)

Syntax scheduler *scheduler-name*;

Hierarchy Level [edit class-of-service scheduler-maps *map-name*]

Release Information Statement introduced before Junos OS Release 7.4.

Description Associate a scheduler with a scheduler map.

Options ***scheduler-name***—Name of the scheduler configuration block.

Usage Guidelines See "Configuring Schedulers" on page 148 and Example: Configuring CoS for a PBB Network on MX Series Routers.

Required Privilege Level interface—To view this statement in the configuration.
interface-control—To add this statement to the configuration.

scheduler-map

See the following sections:

- **scheduler-map (Fabric Queues) on page 553**
- **scheduler-map (Interfaces and Traffic-Control Profiles) on page 553**

scheduler-map (Fabric Queues)

Syntax scheduler-map **priority** (high | low) **scheduler** *scheduler-name*;

Hierarchy Level [edit class-of-service **fabric**]

Release Information Statement introduced before Junos OS Release 7.4.

Description For M320, MX Series, and T Series routers only, associate a scheduler with a fabric priority.

The statements are explained separately.

Usage Guidelines See "Associating Schedulers with Fabric Priorities" on page 200.

Required Privilege Level interface—To view this statement in the configuration.
interface-control—To add this statement to the configuration.

scheduler-map (Interfaces and Traffic-Control Profiles)

Syntax scheduler-map *map-name*;

Hierarchy Level [edit class-of-service interfaces *interface-name*],
[edit class-of-service interfaces *interface-name* **unit** *logical-unit-number*],
[edit class-of-service **traffic-control-profiles**]

Release Information Statement introduced before Junos OS Release 7.4.

Description For Gigabit Ethernet IQ, Channelized IQ PICs, and AS PIC FRF.16 LSQ interfaces only, associate a scheduler map name with an interface or with a traffic-control profile.

For channelized OC12 intelligent queuing (IQ), channelized T3 IQ, channelized E1 IQ, and Gigabit Ethernet IQ interfaces only, you can associate a scheduler map name with a logical interface.

Options ***map-name***—Name of the scheduler map.

Usage Guidelines See "Configuring Schedulers" on page 148 and "Oversubscribing Interface Bandwidth" on page 184.

Required Privilege Level interface—To view this statement in the configuration.
interface-control—To add this statement to the configuration.

Related Documentation

- **output-traffic-control-profile on page 528**

scheduler-map-chassis

Syntax scheduler-map-chassis (derived | *map-name);*

Hierarchy Level [edit class-of-service interfaces *interface-type-fpc/pic/**]

Release Information Statement introduced before Junos OS Release 7.4.

Description For IQ and IQ2 interfaces, assign a custom scheduler to the packet forwarding component queues that control the aggregated traffic transmitted into the entire PIC.

Default If you do not include this statement, on IQ and IQ2 interfaces the aggregated traffic that is fed from the packet forwarding components into the PIC is automatically queued according to the scheduler configuration for each logical unit in the PIC.

Options **derived**—Sets the chassis queues to derive their scheduling configuration from the associated logical interface scheduling configuration.

map-name—Name of the scheduler map configured at the **[edit class-of-service scheduler-maps]** hierarchy level.

Usage Guidelines See “Applying Scheduler Maps to Packet Forwarding Component Queues” on page 194.

Required Privilege Level interface—To view this statement in the configuration.
interface-control—To add this statement to the configuration.

Related Documentation

- **scheduler-map (Fabric Queues) on page 553**

scheduler-maps

See the following sections:

- **scheduler-maps (For ATM2 IQ Interfaces) on page 555**
- **scheduler-maps (For Most Interface Types) on page 556**

scheduler-maps (For ATM2 IQ Interfaces)

Syntax

```
scheduler-maps map-name {
  forwarding-class (class-name | assured-forwarding | best-effort | expedited-forwarding
    | network-control);
  vc-cos-mode (alternate | strict);
}
```

Hierarchy Level [edit interfaces at-*fpc/pic/port* **atm-options**]

Release Information Statement introduced before Junos OS Release 7.4.

Description For ATM2 IQ interfaces only, define CoS parameters assigned to forwarding classes.

Options ***map-name***—Name of the scheduler map.

The remaining statements are explained separately.

Required Privilege Level interface—To view this statement in the configuration.
interface-control—To add this statement to the configuration.

Related Documentation

- Configuring Scheduler Maps on ATM Interfaces on page 416
- **atm-scheduler-map on page 448**

scheduler-maps (For Most Interface Types)

Syntax

```
scheduler-maps {
  map-name {
    forwarding-class class-name scheduler scheduler-name;
  }
}
```

Hierarchy Level [edit class-of-service]

Release Information Statement introduced before Junos OS Release 7.4.

Description Specify a scheduler map name and associate it with the scheduler configuration and forwarding class.

Options ***map-name***—Name of the scheduler map.

The remaining statements are explained separately.

Usage Guidelines See “Configuring Schedulers” on page 148 and Example: Configuring CoS for a PBB Network on MX Series Routers.

Required Privilege Level interface—To view this statement in the configuration.
interface-control—To add this statement to the configuration.

schedulers

See the following sections:

- **schedulers (Class-of-Service) on page 557**
- **schedulers (Interfaces) on page 558**

schedulers (Class-of-Service)

Syntax

```
schedulers {
  scheduler-name {
    buffer-size (seconds | percent percentage | remainder | temporal microseconds);
    drop-profile-map loss-priority (any | low | medium-low | medium-high | high) protocol
      (any | non-tcp | tcp) drop-profile profile-name;
    excess-rate percent percentage
    priority priority-level;
    shaping-rate (percent percentage | rate);
    transmit-rate (percent percentage | rate | remainder) <exact | rate-limit>;
  }
}
```

Hierarchy Level [edit **class-of-service**]

Release Information Statement introduced before Junos OS Release 7.4.

Description Specify scheduler name and parameter values.

Options ***scheduler-name***—Name of the scheduler to be configured.

The remaining statements are explained separately.

Required Privilege Level interface—To view this statement in the configuration.
interface-control—To add this statement to the configuration.

Related Documentation

- Configuring Schedulers on page 148
- Example: Configuring CoS for a PBB Network on MX Series Routers

schedulers (Interfaces)

Syntax schedulers *number*;

Hierarchy Level [edit interfaces]

Release Information Statement introduced in Junos OS Release 8.2.

Description Specify number of schedulers for Ethernet IQ2 PIC port interfaces.

Default If you omit this statement, the 1024 schedulers are distributed equally over all ports in multiples of 4.

Options *number*—Number of schedulers to configure on the port.

Range: 1 through 1024

Usage Guidelines See “Configuring the Number of Schedulers for Ethernet IQ2 PICs” on page 201.

Required Privilege Level interface—To view this statement in the configuration.
interface-control—To add this statement to the configuration.

services

Syntax services cos { ... }

Hierarchy Level [edit]

Release Information Statement introduced in Junos OS Release 8.1.

Description Define the service rules to be applied to traffic.

Options **cos**—Identifies the class-of-service set of rules statements.

Usage Guidelines See “Configuring CoS Rule Sets” on page 286.

Required Privilege Level interface—To view this statement in the configuration.
interface-control—To add this statement to the configuration.

shaping

Syntax

shaping {
(**cbr** *rate* | **rtvbr** peak *rate* sustained *rate* burst *length* | **vbr** peak *rate* sustained *rate* burst *length*);
}

Hierarchy Level

[edit interfaces *interface-name* atm-options vpi *vpi-identifier*],
[edit interfaces *interface-name* **unit** *logical-unit-number*],
[edit interfaces *interface-name* unit *logical-unit-number* address *address* family *family* multipoint-destination *address*],
[edit logical-systems *logical-system-name* interfaces *interface-name* **unit** *logical-unit-number*],
[edit logical-systems *logical-system-name* interfaces *interface-name* unit *logical-unit-number* address *address* family *family* multipoint-destination *address*]

Release Information

Statement introduced before Junos OS Release 7.4.

Description

For ATM encapsulation only, define the traffic-shaping profile.

For ATM2 IQ interfaces, changing or deleting VP tunnel traffic shaping causes all logical interfaces on a VP to be deleted and then added again.

VP tunnels are not supported on multipoint interfaces.

The statements are explained separately.

Required Privilege Level

interface—To view this statement in the configuration.
interface-control—To add this statement to the configuration.

Related Documentation

- Applying Scheduler Maps to Logical ATM Interfaces on page 424

shaping-rate

See the following sections:

- shaping-rate (Applying to an Interface) on page 561
- shaping-rate (Limiting Excess Bandwidth Usage) on page 563
- shaping-rate (Oversubscribing an Interface) on page 564

shaping-rate (Applying to an Interface)

Syntax shaping-rate *rate*;

Hierarchy Level [edit class-of-service interfaces *interface-name*],
[edit class-of-service interfaces *interface-name* unit *logical-unit-number*]

Release Information Statement introduced before Junos OS Release 7.4.
[edit class-of-service interfaces interface *interface-name*] hierarchy level added in Junos OS Release 7.5.

Description For logical interfaces on which you configure packet scheduling, configure traffic shaping by specifying the amount of bandwidth to be allocated to the logical interface.

For physical interfaces on IQ PICs, configure traffic shaping based on the rate-limited bandwidth of the total interface bandwidth.

NOTE: The **shaping-rate** statement cannot be applied to a physical interface on J Series routers.

Logical and physical interface traffic shaping is mutually exclusive. This means you can include the **shaping-rate** statement at the **[edit class-of-service interfaces interface *interface-name*]** hierarchy level or the **[edit class-of-service interfaces interface *interface-name* unit *logical-unit-number*]** hierarchy level, but not both.

NOTE: For MX Series routers, the shaping rate value for the physical interface at the **[edit class-of-service interfaces *interface-name*]** hierarchy level must be a minimum of 160 Kbps. If the value is less than the sum of the logical interface guaranteed rates, the user is not allowed to apply the shaping rate to a physical interface.

Alternatively, you can configure a shaping rate for a 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” on page 184.

For FRF.16 bundles on link services interfaces, only shaping rates based on percentage are supported.

Default If you do not include this statement at the **[edit class-of-service interfaces interface *interface-name* unit *logical-unit-number*]** hierarchy level, the default logical interface bandwidth is the average of unused bandwidth for the number of logical interfaces that require default bandwidth treatment. If you do not include this statement at the **[edit class-of-service interfaces interface *interface-name*]** hierarchy level, the default physical interface bandwidth is the average of unused bandwidth for the number of physical interfaces that require default bandwidth treatment.

Options ***rate***—Peak rate, in bits per second (bps). You can specify a value in bits per second either as a complete decimal number or as a decimal number followed by the abbreviation **k** (1000), **m** (1,000,000), or **g** (1,000,000,000).

Range: For logical interfaces, 1000 through 32,000,000,000 bps.

For physical interfaces, 1000 through 160,000,000,000 bps.

NOTE: For all MX series interfaces, the rate can be from 65,535 through 160,000,000,000 bps.

Usage Guidelines See “Applying Scheduler Maps Overview” on page 167.

Required Privilege Level interface—To view this statement in the configuration.
interface-control—To add this statement to the configuration.

shaping-rate (Limiting Excess Bandwidth Usage)

Syntax shaping-rate (percent *percentage* | *rate*);

Hierarchy Level [edit class-of-service **schedulers** *scheduler-name*]

Release Information Statement introduced before Junos OS Release 7.4.

Description For J Series routers only, define a limit on excess bandwidth usage.

The **transmit-rate** statement at the **[edit class-of-service schedulers *scheduler-name*** hierarchy level configures the minimum bandwidth allocated to a queue. The transmission bandwidth can be configured as an exact value or allowed to exceed the configured rate if additional bandwidth is available from other queues. For J Series routers only, you limit the excess bandwidth usage with this statement.

You should configure the shaping rate as an absolute maximum usage and not the additional usage beyond the configured transmit rate.

Default If you do not include this statement, the default shaping rate is 100 percent, which is the same as no shaping at all.

Options **percent *percentage***—Shaping rate as a percentage of the available interface bandwidth.

Range: 0 through 100 percent

rate—Peak rate, in bits per second (bps). You can specify a value in bits per second either as a complete decimal number or as a decimal number followed by the abbreviation **k** (1000), **m** (1,000,000), or **g** (1,000,000,000).

Range: 3200 through 32,000,000,000 bps

Usage Guidelines See “Applying Scheduler Maps Overview” on page 167.

Required Privilege Level interface—To view this statement in the configuration.
interface-control—To add this statement to the configuration.

shaping-rate (Oversubscribing an Interface)

Syntax shaping-rate (percent *percentage* | *rate*);

Hierarchy Level [edit class-of-service **traffic-control-profiles** *profile-name*]

Release Information Statement introduced in Junos OS Release 7.6.

Description For Gigabit Ethernet IQ, Channelized IQ PICs, and AS PIC FRF.16 LSQ interfaces only, configure a shaping rate for a logical interface. The sum of the shaping rates for all logical interfaces on the physical interface can exceed the physical interface bandwidth. This practice is known as oversubscription of the peak information rate (PIR).

Default The default behavior depends on various factors. For more information, see Table 30 on page 187.

Options **percent *percentage***—For LSQ interfaces, shaping rate as a percentage of the available interface bandwidth.

Range: 1 through 100 percent

rate—For IQ and IQ2 interfaces, peak rate, in bits per second (bps). You can specify a value in bits per second either as a complete decimal number or as a decimal number followed by the abbreviation **k** (1000), **m** (1,000,000), or **g** (1,000,000,000).

Range: 1000 through 160,000,000,000 bps

Usage Guidelines See "Oversubscribing Interface Bandwidth" on page 184 and "Configuring Traffic Control Profiles for Shared Scheduling and Shaping" on page 334.

Required Privilege Level interface—To view this statement in the configuration.
interface-control—To add this statement to the configuration.

Related Documentation
- **output-traffic-control-profile on page 528**

shaping-rate-excess-high

Syntax shaping-rate-excess-high *rate* [burst-size *bytes*];

Hierarchy Level [edit class-of-service **traffic-control-profiles** *profile-name*]

Release Information Statement introduced in Junos OS Release 10.1.

Description For Trio DPC and MPC/MIC interfaces only, configure a shaping rate and optional burst size for high-priority excess traffic. This can help to make sure higher priority services do not starve lower priority services.

Default If you do not include this statement, the default shaping rate for this priority is determined by the **shaping-rate** statement in the traffic control profile.

Options *rate*—Peak rate, in bits per second (bps). You can specify a value in bits per second either as a complete decimal number or as a decimal number followed by the abbreviation **k** (1000), **m** (1,000,000), or **g** (1,000,000,000).

Range: 1000 through 160,000,000,000

Default: None

burst-size ***bytes***—Maximum burst size, in bytes.

Range: 0 through 1,000,000,000

Required Privilege Level interface—To view this statement in the configuration.
interface-control—To add this statement to the configuration.

Related Documentation

- Per-Priority Shaping on the Trio MPC/MIC Interfaces Overview on page 381
- Oversubscribing Interface Bandwidth on page 184
- Configuring Traffic Control Profiles for Shared Scheduling and Shaping on page 334
- **shaping-rate-excess-low on page 566**
- **shaping-rate-priority-high on page 567**
- **shaping-rate-priority-low on page 568**
- **shaping-rate-priority-medium on page 569**

shaping-rate-excess-low

Syntax shaping-rate-excess-low *rate* [burst-size *bytes*];

Hierarchy Level [edit class-of-service **traffic-control-profiles** *profile-name*]

Release Information Statement introduced in Junos OS Release 10.1.

Description For Trio DPC and MPC/MIC interfaces only, configure a shaping rate and optional burst size for low-priority excess traffic. This can help to make sure higher priority services do not starve lower priority services.

Default If you do not include this statement, the default shaping rate for this priority is determined by the **shaping-rate** statement in the traffic control profile.

Options ***rate***—Peak rate, in bits per second (bps). You can specify a value in bits per second either as a complete decimal number or as a decimal number followed by the abbreviation **k** (1000), **m** (1,000,000), or **g** (1,000,000,000).

Range: 1000 through 160,000,000,000

Default: None

burst-size ***bytes***—Maximum burst size, in bytes.

Range: 0 through 1,000,000,000

Required Privilege Level interface—To view this statement in the configuration.
interface-control—To add this statement to the configuration.

Related Documentation

- Per-Priority Shaping on the Trio MPC/MIC Interfaces Overview on page 381
- Oversubscribing Interface Bandwidth on page 184
- Configuring Traffic Control Profiles for Shared Scheduling and Shaping on page 334
- **shaping-rate-excess-high on page 565**
- **shaping-rate-priority-high on page 567**
- **shaping-rate-priority-low on page 568**
- **shaping-rate-priority-medium on page 569**

shaping-rate-priority-high

Syntax shaping-rate-priority-high *rate* [burst-size *bytes*];

Hierarchy Level [edit class-of-service **traffic-control-profiles** *profile-name*]

Release Information Statement introduced in Junos OS Release 10.1.

Description For Trio DPC and MPC/MIC interfaces only, configure a shaping rate and optional burst size for high priority traffic. This can help to make sure higher priority services do not starve lower priority services.

Default If you do not include this statement, the default shaping rate for this priority is determined by the **shaping-rate** statement in the traffic control profile.

Options *rate*—Peak rate, in bits per second (bps). You can specify a value in bits per second either as a complete decimal number or as a decimal number followed by the abbreviation **k** (1000), **m** (1,000,000), or **g** (1,000,000,000).

Range: 1000 through 160,000,000,000

Default: None

burst-size ***bytes***—Maximum burst size, in bytes.

Range: 0 through 1,000,000,000

Required Privilege Level interface—To view this statement in the configuration.
interface-control—To add this statement to the configuration.

Related Documentation

- Per-Priority Shaping on the Trio MPC/MIC Interfaces Overview on page 381
- Oversubscribing Interface Bandwidth on page 184
- Configuring Traffic Control Profiles for Shared Scheduling and Shaping on page 334
- **shaping-rate-excess-high on page 565**
- **shaping-rate-excess-low on page 566**
- **shaping-rate-priority-low on page 568**
- **shaping-rate-priority-medium on page 569**

shaping-rate-priority-low

Syntax shaping-rate-priority-low *rate* [burst-size *bytes*];

Hierarchy Level [edit class-of-service **traffic-control-profiles** *profile-name*]

Release Information Statement introduced in Junos OS Release 10.1.

Description For Trio DPC and MPC/MIC interfaces only, configure a shaping rate and optional burst size for low priority traffic. This can help to make sure higher priority services do not starve lower priority services.

Default If you do not include this statement, the default shaping rate for this priority is determined by the **shaping-rate** statement in the traffic control profile.

Options *rate*—Peak rate, in bits per second (bps). You can specify a value in bits per second either as a complete decimal number or as a decimal number followed by the abbreviation **k** (1000), **m** (1,000,000), or **g** (1,000,000,000).

Range: 1000 through 160,000,000,000

Default: None

burst-size *bytes*—Maximum burst size, in bytes.

Range: 0 through 1,000,000,000

Required Privilege Level interface—To view this statement in the configuration.
interface-control—To add this statement to the configuration.

Related Documentation

- Per-Priority Shaping on the Trio MPC/MIC Interfaces Overview on page 381
- Oversubscribing Interface Bandwidth on page 184
- Configuring Traffic Control Profiles for Shared Scheduling and Shaping on page 334
- **shaping-rate-excess-high on page 565**
- **shaping-rate-excess-low on page 566**
- **shaping-rate-priority-high on page 567**
- **shaping-rate-priority-medium on page 569**

shaping-rate-priority-medium

Syntax shaping-rate-priority-medium *rate* [burst-size *bytes*];

Hierarchy Level [edit class-of-service **traffic-control-profiles** *profile-name*]

Release Information Statement introduced in Junos OS Release 10.1.

Description For Trio DPC and MPC/MIC interfaces only, configure a shaping rate and optional burst size for medium priority traffic. This can help to make sure higher priority services do not starve lower priority services.

Default If you do not include this statement, the default shaping rate for this priority is determined by the **shaping-rate** statement in the traffic control profile.

Options *rate*—Peak rate, in bits per second (bps). You can specify a value in bits per second either as a complete decimal number or as a decimal number followed by the abbreviation **k** (1000), **m** (1,000,000), or **g** (1,000,000,000).

Range: 1000 through 160,000,000,000

Default: None

burst-size ***bytes***—Maximum burst size, in bytes.

Range: 0 through 1,000,000,000

Required Privilege Level interface—To view this statement in the configuration.
interface-control—To add this statement to the configuration.

Related Documentation

- Per-Priority Shaping on the Trio MPC/MIC Interfaces Overview on page 381
- Oversubscribing Interface Bandwidth on page 184
- Configuring Traffic Control Profiles for Shared Scheduling and Shaping on page 334
- **shaping-rate-excess-high on page 565**
- **shaping-rate-excess-low on page 566**
- **shaping-rate-priority-high on page 567**
- **shaping-rate-priority-low on page 568**

shared-instance

Syntax
shared-instance *instance-name*;

Hierarchy Level
[edit class-of-service interfaces *interface-name* **unit** *logical-unit-number* **input-traffic-control-profile**],
[edit class-of-service interfaces *interface-name* **unit** *logical-unit-number* **output-traffic-control-profile**]

Release Information
Statement introduced in Junos OS Release 7.6.

Description
For Gigabit Ethernet IQ2 and IQ2E PICs only, apply a shared traffic scheduling and shaping profile to the logical interface.

Options
instance-name—Name of the shared scheduler and shaper to be applied to this interface

Required Privilege Level
interface—To view this statement in the configuration.
interface-control—To add this statement to the configuration.

Related Documentation
- Configuring Shaping on 10-Gigabit Ethernet IQ2 PICs on page 328
- **traffic-control-profiles on page 581**

shared-scheduler

Syntax
shared-scheduler;

Hierarchy Level
[edit **interfaces** *interface-name*]

Release Information
Statement introduced in Junos OS Release 7.6.

Description
For Gigabit Ethernet IQ2 PICs only, enable shared schedulers and shapers on this interface. This statement and the **per-unit-scheduler** statement are mutually exclusive. Even so, you can configure one logical interface for each shared instance. This effectively provides the functionality of per-unit scheduling.

Usage Guidelines
See "Configuring Shaping on 10-Gigabit Ethernet IQ2 PICs" on page 328.

Required Privilege Level
interface—To view this statement in the configuration.
interface-control—To add this statement to the configuration.

Related Documentation
- **traffic-control-profiles on page 581**

simple-filter

See the following sections:

- **simple-filter (Applying to an Interface) on page 571**
- **simple-filter (Configuring) on page 572**

simple-filter (Applying to an Interface)

Syntax

```
simple-filter {
  input filter-name;
}
```

Hierarchy Level

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

Release Information

Statement introduced in Junos OS Release 7.6.

Description

Apply a simple filter to an interface. You can apply simple filters to the family **inet** only, and only in the input direction.

Options

input *filter-name*—Name of one filter to evaluate when packets are received on the interface.

Usage Guidelines

See "Configuring Multifield Classifiers" on page 72; for a general discussion of this statement, see the *Junos OS Network Interfaces Configuration Guide*.

Required Privilege Level

interface—To view this statement in the configuration.
interface-control—To add this statement to the configuration.

Related Documentation

- **filter on page 482**

simple-filter (Configuring)

Syntax

```
simple-filter filter-name {
  term term-name {
    from {
      match-conditions;
    }
    then {
      forwarding-class class-name;
      loss-priority (high | low | medium);
    }
  }
}
```

Hierarchy Level [edit firewall family inet **filter** *filter-name*]

Release Information Statement introduced in Junos OS Release 7.6.

Description Define a simple filter. Simple filters are recommended for metropolitan Ethernet applications.

Options **from**—Match packet fields to values. If the **from** option is not included, all packets are considered to match and the actions and action modifiers in the **then** statement are taken.

match-conditions—One or more conditions to use to make a match. The conditions are described in the *Junos OS Policy Framework Configuration Guide*.

term-name—Name that identifies the term. The name can contain letters, numbers, and hyphens (-), and can be up to 255 characters long. To include spaces in the name, enclose it in quotation marks (“ ”).

then—Actions to take on matching packets. If the **then** option is not included and a packet matches all the conditions in the **from** statement, the packet is accepted.

The remaining statements are explained separately. Only **forwarding-class** and **loss-priority** are valid in a simple filter configuration.

Required Privilege Level firewall—To view this statement in the configuration.
firewall-control—To add this statement to the configuration.

Related Documentation

- Configuring Multifield Classifiers on page 72
- *Junos OS Policy Framework Configuration Guide*
- **filter (Applying to an Interface) on page 482**
- **simple-filter (Applying to an Interface) on page 571**

sip

Syntax

```
sip {
  video {
    dscp (alias | bits);
    forwarding-class class-name;
  }
  voice {
    dscp (alias | bits);
    forwarding-class class-name;
  }
}
```

Hierarchy Level [edit services cos **application-profile** *profile-name*]

Release Information Statement introduced in Junos OS Release 9.3.

Description Set the appropriate **dscp** and **forwarding-class** value for SIP traffic.

Default By default, the system will not alter the DSCP or forwarding class for SIP traffic.

Required Privilege Level interface—To view this statement in the configuration.
interface-control—To add this statement to the configuration.

Related Documentation

- Configuring Application Profiles on page 285
- **ftp on page 495**

source-address

Syntax source-address *address*;

Hierarchy Level [edit **services** cos rule *rule-name* term *term-name* **from**]

Release Information Statement introduced in Junos OS Release 8.1.

Description Source address for rule matching.

Options ***address***—Source IP address or prefix value.

Required Privilege Level interface—To view this statement in the configuration.
interface-control—To add this statement to the configuration.

Related Documentation

- Configuring Match Conditions in a CoS Rule on page 284

syslog

Syntax syslog;

Hierarchy Level [edit **services** cos rule *rule-name* term *term-name* **then**],
[edit **services** cos rule *rule-name* term *term-name* then **(reflexive | reverse)**]

Release Information Statement introduced in Junos OS Release 8.1.

Description Enable system logging. The system log information from the AS PIC is passed to the kernel for logging in the **/var/log** directory. This setting overrides any **syslog** statement setting included in the service set or interface default configuration.

Required Privilege Level interface—To view this statement in the configuration.
interface-control—To add this statement to the configuration.

Related Documentation

- Configuring Actions in a CoS Rule on page 285

term

See the following sections:

- **term (AS PIC Classifiers) on page 575**
- **term (Normal Filter) on page 576**
- **term (Simple Filter) on page 577**

term (AS PIC Classifiers)

Syntax

```
term term-name {
  from {
    applications [ application-names ];
    application-sets [ set-names ];
    destination-address address;
    source-address address;
  }
  then {
    application-profile profile-name;
    dscp (alias | bits);
    forwarding-class class-name;
    syslog;
    (reflexive | reverse) {
      application-profile profile-name;
      dscp (alias | bits);
      forwarding-class class-name;
      syslog;
    }
  }
}
```

Hierarchy Level [edit **services cos rule** *rule-name*]

Release Information Statement introduced in Junos OS Release 8.1.

Description Define the CoS term properties.

Options ***term-name***—Identifier for the term.

The remaining statements are explained separately.

Required Privilege Level interface—To view this statement in the configuration.
interface-control—To add this statement to the configuration.

Related Documentation
- Configuring CoS Rules on page 282

term (Normal Filter)

Syntax

```
term term-name {
  from {
    match-conditions;
  }
  then {
    forwarding-class class-name;
    loss-priority (high | low);
    three-color-policer {
      (single-rate | two-rate) policer-name;
    }
  }
}
```

Hierarchy Level [edit firewall family *family-name* **filter** *filter-name*]

Release Information Statement introduced before Junos OS Release 7.4.

Description Define a firewall filter term.

Options

from—Match packet fields to values. If not included, all packets are considered to match and the actions and action modifiers in the **then** statement are taken.

match-conditions—One or more conditions to use to make a match. The conditions are described in the *Junos OS Policy Framework Configuration Guide*.

term-name—Name that identifies the term. The name can contain letters, numbers, and hyphens (-), and can be up to 255 characters long. To include spaces in the name, enclose it in quotation marks (“ ”).

then—Actions to take on matching packets. If not included and a packet matches all the conditions in the **from** statement, the packet is accepted. For CoS, only the actions listed are allowed. These statements are explained separately.

Required Privilege Level

firewall—To view this statement in the configuration.
firewall-control—To add this statement to the configuration.

Related Documentation

- Configuring Multifield Classifiers on page 72
- *Junos OS Policy Framework Configuration Guide*

term (Simple Filter)

Syntax

```
term term-name {
  from {
    match-conditions;
  }
  then {
    forwarding-class class-name;
    loss-priority (high | low | medium);
  }
}
```

Hierarchy Level [edit firewall family inet **simple-filter** *filter-name*]

Release Information Statement introduced in Junos OS Release 7.6.

Description Define a simple filter term.

Options **from**—Match packet fields to values. If the **from** option is not included, all packets are considered to match and the actions and action modifiers in the **then** statement are taken.

match-conditions—One or more conditions to use to make a match. The conditions are described in the *Junos OS Policy Framework Configuration Guide*.

term-name—Name that identifies the term. The name can contain letters, numbers, and hyphens (-), and can be up to 255 characters long. To include spaces in the name, enclose it in quotation marks (" ").

then—Actions to take on matching packets. If the **then** option is not included and a packet matches all the conditions in the **from** statement, the packet is accepted. For CoS, only the actions listed are allowed. These statements are explained separately.

Usage Guidelines See "Configuring Multifield Classifiers" on page 72; for a general discussion of this statement, see the *Junos OS Policy Framework Configuration Guide*.

Required Privilege Level firewall—To view this statement in the configuration.
firewall-control—To add this statement to the configuration.

then

Syntax

```
then {
  application-profile profile-name;
  dscp (alias | bits);
  forwarding-class class-name;
  syslog;
  (reflexive | reverse) {
    application-profile profile-name;
    dscp (alias | bits);
    forwarding-class class-name;
    syslog;
  }
}
```

Hierarchy Level [edit **services** cos rule *rule-name* **term** *term-name*]

Release Information Statement introduced in Junos OS Release 8.1.

Description Define the CoS term actions.

The remaining statements are explained separately.

Required Privilege Level interface—To view this statement in the configuration.
interface-control—To add this statement to the configuration.

Related Documentation

- Configuring Actions in a CoS Rule on page 285
- *Junos OS Policy Framework Configuration Guide*

three-color-policer

See the following sections:

- **three-color-policer (Applying) on page 579**
- **three-color-policer (Configuring) on page 580**

three-color-policer (Applying)

Syntax

```
three-color-policer {
   (single-rate | two-rate) policer-name;
}
```

Hierarchy Level [edit firewall family *family-name* filter *filter-name* term *term-name* then]

Release Information Statement introduced in Junos OS Release 7.4.
single-rate statement added in Junos OS Release 8.2.

Description For M320 and T Series routers with Enhanced II Flexible PIC Concentrators (FPCs) and the T640 router with Enhanced Scaling FPC4, apply a tricolor marking policer.

Options **single-rate**—Named tricolor policer is a single-rate policer.

two-rate—Named tricolor policer is a two-rate policer.

policer-name—Name of a tricolor policer.

Required Privilege Level firewall—To view this statement in the configuration.
firewall-control—To add this statement to the configuration.

Related Documentation

- Applying Tricolor Marking Policers to Firewall Filters on page 99

three-color-policer (Configuring)

Syntax

```
three-color-policer policer-name {
  action {
    loss-priority high then discard;
  }
  logical-interface-policer;
  single-rate {
    (color-aware | color-blind);
    committed-information-rate bps;
    committed-burst-size bytes;
    excess-burst-size bytes;
  }
  two-rate {
    (color-aware | color-blind);
    committed-information-rate bps;
    committed-burst-size bytes;
    peak-information-rate bps;
    peak-burst-size bytes;
  }
}
```

Hierarchy Level [edit firewall]

Release Information Statement introduced in Junos OS Release 7.4.
The **action** and **single-rate** statements added in Junos OS Release 8.2.

Description For M320, MX Series, and T Series routers with Enhanced II Flexible PIC Concentrators (FPCs), configure a tricolor marking policer.

Options

single-rate—Marking is based on the CIR.

two-rate—Marking is based on the CIR and the PIR.

color-aware—Metering varies by preclassification. Metering can increase a packet's assigned PLP, but cannot decrease it.

color-blind—All packets are evaluated by the CIR or CBS. If a packet exceeds the CIR or CBS, it is evaluated by the PIR or EBS.

committed-burst-size *bytes*—Guaranteed deliverable burst.

Range: 1500 through 100,000,000,000

committed-information-rate *bps*—Guaranteed bandwidth under normal line conditions.

Range: 1500 through 100,000,000,000

excess-burst-size *bytes*—Maximum allowable excess burst.

Range: 1500 through 100,000,000,000

peak-burst-size *bytes*—Maximum allowable burst.

Range: 1500 through 100,000,000,000

peak-information-rate *bps*—Maximum achievable rate.

Range: 1500 through 100,000,000,000

The remaining statements are explained separately.

Required Privilege Level
firewall—To view this statement in the configuration.
firewall-control—To add this statement to the configuration.

Related Documentation
- Configuring Tricolor Marking Policers on page 98

traffic-control-profiles

Syntax

```
traffic-control-profiles profile-name {
  delay-buffer-rate (percent percentage | rate);
  excess-rate (percent percentage  | proportion value );
  guaranteed-rate (percent percentage | rate);
  overhead-accounting (frame-mode | cell-mode) <bytes byte-value>;
  scheduler-map map-name;
  shaping-rate (percent percentage | rate);
  ADD 5 statements for RLI 6132
}
```

Hierarchy Level [edit class-of-service]

Release Information
Statement introduced in Junos OS Release 7.6.
excess-rate statement introduced in Junos OS Release 9.3.

Description For Gigabit Ethernet IQ, Channelized IQ PICs, and AS PIC FRF.16 LSQ interfaces only, configure traffic shaping and scheduling profiles. For Enhanced EQ PICs only, you can include the **excess-rate** statement.

Options *profile-name*—Name of the traffic-control profile.

The remaining statements are explained separately.

Required Privilege Level
interface—To view this statement in the configuration.
interface-control—To add this statement to the configuration.

Related Documentation
- Oversubscribing Interface Bandwidth on page 184
- Example: Configuring CoS for a PBB Network on MX Series Routers
- **output-traffic-control-profile on page 528**

traffic-manager

Syntax

```
traffic-manager {
  egress-shaping-overhead number;
  ingress-shaping-overhead number;
  mode session-shaping;
}
```

Hierarchy Level [edit chassis fpc *slot-number* pic *pic-number*],
[edit chassis lcc *number* fpc *slot-number* pic *pic-number*]

Release Information Statement introduced in Junos OS Release 8.3.

Description Enable shaping on an L2TP session.

The remaining statements are explained separately.

Required Privilege Level interface—To view this statement in the configuration.
interface-control—To add this statement to the configuration.

Related Documentation

- Configuring CoS for L2TP Tunnels on ATM Interfaces on page 426

translation-table

Syntax

```
translation-table {
  (to-dscp-from-dscp | to-dscp-ipv6-from-dscp-ipv6 | to-exp-from-exp |
    to-inet-precedence-from-inet-precedence) table-name {
    to-code-point value from-code-points (* | [ values ]);
  }
}
```

Hierarchy Level [edit class-of-service]

Release Information Statement introduced in Junos OS Release 9.3.
Support on Multiservices PIC added in Junos OS Release 9.5.

Description For an Enhanced IQ PIC or Multiservices PIC, specify the input translation tables. You must also apply the translation table to a logical interface on the Enhanced IQ PIC or Multiservices PIC.

Default If you do not include this statement, the ToS bit values in received packet headers are not changed by the PIC.

Options **to-dscp-from-dscp**—(Optional) Translate incoming IPv4 DSCP values to new values. You must also configure and apply a DSCP classifier.

to-dscp-ipv6-from-dscp-ipv6—(Optional) Translate incoming IPv6 DSCP values to new values. You must also configure and apply an IPv6 DSCP classifier.

to-inet-precedence-from-inet-precedence—(Optional) Translate incoming INET precedence values to new values.

to-exp-from-exp—(Optional) Translate incoming MPLS EXP values to new values.

table-name—The name of the translation table.

value—The bit string to which to translate the incoming bit value.

value(s)—The bit string(s) from which the incoming bit value(s) are translated.

*****—(Optional) This translation matches all bit patterns not explicitly listed.

Usage Guidelines See "Configuring ToS Translation Tables" on page 291 and "Multiservices PIC ToS Translation" on page 289.

Required Privilege Level interface—To view this statement in the configuration.
interface-control—To add this statement to the configuration.

transmit-rate

Syntax transmit-rate (*rate* | percent *percentage* | remainder) <exact | rate-limit>;

Hierarchy Level [edit class-of-service **schedulers** *scheduler-name*]

Release Information Statement introduced before Junos OS Release 7.4.
rate-limit option introduced in Junos OS Release 8.3. Applied to the Multiservices PICs in Junos OS Release 9.4.

Description Specify the transmit rate or percentage for a scheduler.

Default If you do not include this statement, 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.

Options **exact**—(Optional) Enforce the exact transmission rate. Under sustained congestion, a rate-controlled queue that goes into negative credit fills up and eventually drops packets. This value should never exceed the rate-controlled amount.

percent *percentage*—Percentage of transmission capacity. A percentage of zero drops all packets in the queue.

Range: 0 through 100 percent

rate—Transmission rate, in bps. You can specify a value in bits per second either as a complete decimal number or as a decimal number followed by the abbreviation **k** (1000), **m** (1,000,000), or **g** (1,000,000,000).

Range: 3200 through 160,000,000,000 bps

NOTE: For all MX series interfaces, the rate can be from 65,535 through 160,000,000,000 bps.

rate-limit—(Optional) Limit the transmission rate to the rate-controlled amount. In contrast to the **exact** option, the scheduler with the **rate-limit** option shares unused bandwidth above the rate-controlled amount.

remainder—Use remaining rate available.

Required Privilege Level interface—To view this statement in the configuration.
interface-control—To add this statement to the configuration.

Related Documentation

- Configuring Schedulers on page 148
- Example: Configuring CoS for a PBB Network on MX Series Routers

transmit-weight

Syntax transmit-weight (cells *number* | percent *number*);

Hierarchy Level [edit interfaces at-*fpc/pic/port* **atm-options** scheduler-maps *map-name* **forwarding-class**) *class-name*]

Release Information Statement introduced before Junos OS Release 7.4.

Description For ATM2 IQ interfaces only, assign a transmission weight to a forwarding class.

Default 95 percent for queue 0, 5 percent for queue 3.

Options **percent** ***percentage***—Transmission weight of the forwarding class as a percentage of the total bandwidth.

Range: 5 through 100

cells ***number***—Transmission weight of the forwarding class as a number of cells.

Range: 0 through 32,000

Required Privilege Level interface—To view this statement in the configuration.
interface-control—To add this statement to the configuration.

Related Documentation

- Configuring Scheduler Maps on ATM Interfaces on page 416

tri-color

Syntax tri-color;

Hierarchy Level [edit class-of-service]

Release Information Statement introduced in Junos OS Release 7.4.

Description For IPv4 packets on M320, MX Series, and T Series routers with Enhanced II Flexible PIC Concentrators (FPCs), enable two-rate tricolor marking (TCM), as defined in RFC 2698.

Default If you do not include this statement, tricolor marking is not enabled and the medium packet loss priority (PLP) is not configurable.

Required Privilege Level interface—To view this statement in the configuration.
interface-control—To add this statement to the configuration.

Related Documentation

- Configuring Tricolor Marking on page 89

unit

Syntax

```
unit logical-unit-number {
  classifiers {
    type (classifier-name | default) family (mpls | all);
  }
  forwarding-class class-name;
  fragmentation-map map-name;
  input-traffic-control-profile profiler-name shared-instance instance-name;
  output-traffic-control-profile profile-name shared-instance instance-name;
  per-session-scheduler;
  rewrite-rules {
    dscp (rewrite-name | default);
    dscp-ipv6 (rewrite-name | default);
    exp (rewrite-name | default)protocol protocol-types;
    exp-push-push-push default;
    exp-swap-push-push default;
    ieee-802.1 (rewrite-name | default) vlan-tag (outer | outer-and-inner);
    inet-precedence (rewrite-name | default);
  }
  scheduler-map map-name;
  shaping-rate rate;
}
```

Hierarchy Level [edit class-of-service **interfaces** *interface-name*]

Release Information Statement introduced before Junos OS Release 7.4.

Description Configure a logical interface on the physical device. You must configure a logical interface to be able to use the physical device.

Options *logical-unit-number*—Number of the logical unit.

Range: 0 through 16,384

The remaining statements are explained separately.

Required Privilege Level interface—To view this statement in the configuration.
interface-control—To add this statement to the configuration.

Related Documentation

- Overview of BA Classifier Types on page 42
- Configuring Rewrite Rules on page 242

vbr

Syntax vbr peak *rate* sustained *rate* burst *length*;

Hierarchy Level [edit interfaces *interface-name* atm-options vpi *vpi-identifier* **shaping**],
[edit interfaces *interface-name* unit *logical-unit-number* address *address* family *family* multipoint-destination *address* **shaping**],
[edit interfaces *interface-name* unit *logical-unit-number* **shaping**],
[edit logical-systems *logical-system-name* interfaces *interface-name* unit *logical-unit-number* address *address* family *family* multipoint-destination *address* **shaping**],
[edit logical-systems *logical-system-name* interfaces *interface-name* **unit** *logical-unit-number* **shaping**]

Release Information Statement introduced before Junos OS Release 7.4.

Description For ATM encapsulation only, define the variable bandwidth utilization in the traffic-shaping profile.

When you configure the variable bandwidth utilization, you must specify all three options (**burst**, **peak**, and **sustained**). You can specify ***rate*** in bits per second either as a complete decimal number or as a decimal number followed by the abbreviation **k** (1000), **m** (1,000,000), or **g** (1,000,000,000). You can also specify ***rate*** in cells per second by entering a decimal number followed by the abbreviation **c**; values expressed in cells per second are converted to bits per second by means of the formula 1 cps = 384 bps.

Default If the **vbr** statement is not specified, bandwidth utilization is unlimited.

Options **burst *length***—Burst length, in cells. If you set the length to 1, the peak traffic rate is used.

Range: 1 through 4000 cells

peak *rate*—Peak rate, in bits per second or cells per second.

Range: For ATM1 interfaces, 33 Kbps through 135.6 Mbps (ATM OC3); 33 Kbps through 276 Mbps (ATM OC12).For ATM2 IQ OC3 and OC12 interfaces, 33 Kbps through 542,526,792 bps. For ATM2 IQ OC48 interfaces, 33 Kbps through 2,170,107,168 bps. For ATM2 IQ DS3 and E3 interfaces, from 33 Kbps through the maximum rate, which depends on the ATM encapsulation and framing you configure. For more information, see the *Junos OS Network Interfaces Configuration Guide*.

sustained *rate*—Sustained rate, in bits per second or cells per second.

Range: For ATM1 interfaces, 33 Kbps through 135.6 Mbps (ATM OC3); 33 Kbps through 276 Mbps (ATM OC12).For ATM2 IQ OC3 and OC12 interfaces, 33 Kbps through 542,526,792 bps. For ATM2 IQ OC48 interfaces, 33 Kbps through 2,170,107,168 bps. For ATM2 IQ DS3 and E3 interfaces, from 33 Kbps through the maximum rate, which depends on the ATM encapsulation and framing you configure. For more information, see the *Junos OS Network Interfaces Configuration Guide*.

Required Privilege Level interface—To view this statement in the configuration.
interface-control—To add this statement to the configuration.

Related Documentation

- Applying Scheduler Maps to Logical ATM Interfaces on page 424
- **cbr on page 450**
- **rtvbr on page 549**
- **shaping on page 559**

vc-cos-mode

Syntax vc-cos-mode (alternate | strict);

Hierarchy Level [edit interfaces *interface-name* **atm-options scheduler-maps** *map-name*]

Release Information Statement introduced before Junos OS Release 7.4.

Description For ATM2 IQ interfaces only, specify packet-scheduling priority value for ATM2 IQ VC tunnels.

Options **alternate**—VC CoS queue has high priority. The scheduling of the queues alternates between the high-priority queue and the remaining queues, so every other scheduled packet is from the high-priority queue.

strict—VC CoS queue has strictly high priority. A queue with strict high priority is always scheduled before the remaining queues. The remaining queues are scheduled in round-robin fashion.

Default: alternate

Required Privilege Level interface—To view this statement in the configuration.
interface-control—To add this statement to the configuration.

Related Documentation

- Configuring Scheduler Maps on ATM Interfaces on page 416

vci

Syntax vci *vpi-identifier.vci-identifier*;

Hierarchy Level [edit interfaces at-*fpc/pic/port* **unit** *logical-unit-number*],
[edit interfaces at-*fpc/pic/port* **unit** *logical-unit-number* family *family* address *address* multipoint-destination *address*],
[edit logical-systems *logical-system-name* interfaces at- *fpc/pic/port* **unit** *logical-unit-number*],
[edit logical-systems *logical-system-name* interfaces at-*fpc/pic/port* unit *logical-unit-number* family *family* address *address* multipoint-destination *address*]

Release Information Statement introduced before Junos OS Release 7.4.

Description For ATM point-to-point logical interfaces only, configure the virtual circuit identifier (VCI) and virtual path identifier (VPI).

To configure a VPI for a point-to-multipoint interface, specify the VPI in the **multipoint-destination** statement.

VCIs 0 through 31 are reserved for specific ATM values designated by the ATM Forum.

Options ***vci-identifier***—ATM virtual circuit identifier. Unless you configure the interface to use promiscuous mode, this value cannot exceed the largest numbered VC configured for the interface with the **maximum-vcs** option of the **vpi** statement.

Range: 0 through 4089 or 0 through 65,535 with promiscuous mode, with VCIs 0 through 31 reserved.

vpi-identifier—ATM virtual path identifier.

Range: 0 through 255

Default: 0

Required Privilege Level interface—To view this statement in the configuration.
interface-control—To add this statement to the configuration.

Related Documentation

- Applying Scheduler Maps to Logical ATM Interfaces on page 424

video

Syntax

```
video {
  dscp (alias | bits);
  forwarding-class class-name;
}
```

Hierarchy Level [edit services cos **application-profile** *profile-name* sip]

Release Information Statement introduced in Junos OS Release 9.3.

Description Set the appropriate **dscp** and **forwarding-class** values for SIP video traffic.

Default By default, the system will not alter the DSCP or forwarding class for SIP video traffic.

Required Privilege Level interface—To view this statement in the configuration.
interface-control—To add this statement to the configuration.

Related Documentation

- Configuring Application Profiles on page 285
- **voice on page 591**

vlan-tag

Syntax vlan-tag (outer | outer-and-inner);

Hierarchy Level [edit class-of-service interfaces *interface-name* unit *logical-unit-number* rewrite-rules ieee-802.1 (*rewrite-name* | default)]

Release Information Statement introduced in Junos OS Release 8.1.

Description For Gigabit Ethernet IQ2 PICs only, apply this IEEE-802.1 rewrite rule to the outer or outer and inner VLAN tags.

Default If you do not include this statement, the rewrite rule applies to the outer VLAN tag only.

Options **outer**—Apply the rewrite rule to the outer VLAN tag only.

outer-and-inner—Apply the rewrite rule to both the outer and inner VLAN tags.

Required Privilege Level interface—To view this statement in the configuration.
interface-control—To add this statement to the configuration.

Related Documentation

- Applying IEEE 802.1p Rewrite Rules to Dual VLAN Tags on page 245

voice

Syntax

```
voice {
  dscp (alias | bits);
  forwarding-class class-name;
}
```

Hierarchy Level [edit services cos **application-profile** *profile-name* sip]

Release Information Statement introduced in Junos OS Release 9.3.

Description Set the appropriate **dscp** and **forwarding-class** values for SIP voice traffic.

Default By default, the system will not alter the DSCP or forwarding class for SIP voice traffic.

Required Privilege Level interface—To view this statement in the configuration.
interface-control—To add this statement to the configuration.

Related Documentation

- Configuring Application Profiles on page 285
- **video on page 590**

PART 6

Index

Index

Symbols

A

B

E

F

L

R

S

U

V

W

Index of Statements and Commands