JUNIPER NETWORKS

Junos® OS

CoS Output Queuing and Scheduling Feature Guide for Routing Devices

Release

13.2

Published: 2013-07-31

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

Junos® OS CoS Output Queuing and Scheduling Feature Guide for Routing Devices
13.2

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

YEAR 2000 NOTICE

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

END USER LICENSE AGREEMENT

Table of Contents

List of Figures

List of Tables

About the Documentation

Documentation and 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/.

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

Juniper Networks Books publishes books by Juniper Networks engineers and subject matter experts. These books go beyond the technical documentation to explore the nuances of network architecture, deployment, and administration. The current list can be viewed at http://www.juniper.net/books.

Supported Platforms

For the features described in this document, the following platforms are supported:

- T Series
- M Series
- MX Series
- J Series
- PTX Series
- ACX Series

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. The example does not become active until you commit the 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 *CLI User Guide*.

Documentation Conventions

Table 1 on page xiii 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 xiv 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 or emphasizes 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 OS 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; 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*>;**
| (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.	

GUI Conventions

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

Convention	Description	Examples
Bold text like this	Represents 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 menu 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 post-sales 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/
- Search for known bugs: http://www2.juniper.net/kb/

- 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, see http://www.juniper.net/support/requesting-support.html.

PART 1

Overview

CHAPTER 1

Schedulers and Scheduler Maps

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:

NOTE: For PTX Series Packet Transport Routers:

- The **fabric** and **traffic-control-profiles** statements at the **[edit class-of-service]** hierarchy level are not supported.

```
[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 | proportion value);
    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 Overview

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 54.

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;
  }
}
```

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 89
- Applying Scheduler Maps and Shaping Rate to Physical Interfaces on IQ PICs on page 97
- Applying Scheduler Maps and Shaping Rate to DLCIs and VLANs on page 103
- Oversubscribing Interface Bandwidth on page 60
- Providing a Guaranteed Minimum Rate on page 69
- Applying Scheduler Maps to Packet Forwarding Component Queues on page 90
- *Forwarding Classes and Fabric Priority Queues*
- Associating Schedulers with Fabric Priorities on page 72

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 and switches:

- EX Series switches
- 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 3D Universal Edge 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 queuing the packet for transmission. All packets that exceed the rate limit are 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 or EX Series switches, 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.

NOTE: Rate limiting is implemented differently on Enhanced Queuing DPCs and non-queuing Packet Forwarding Engines. On Enhanced Queuing DPCs, rate-limiting is implemented using a single rate two color policer. On non-queuing Packet Forwarding Engines, rate-limiting is achieved by shaping the queue to the transmit rate and keeping the queue delay buffers small to prevent too many packets from being queued once the shaping rate is reached.

CHAPTER 2

Priority Scheduling

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 51. 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 55.
- 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 Library for Routing Devices*.
- 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 3 on page 11 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 3: 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

Related Documentation

- Schedulers Overview on page 3
- Configuring Schedulers for Priority Scheduling on page 55

CHAPTER 3

Scheduling on Specific Interface Types

Applying a Shaping Rate to Physical Interfaces Overview

On T4000 routers with Type 5 FPCs and on EX Series switches, you can configure physical interfaces to shape traffic based on the rate-limited bandwidth of the total interface bandwidth. This allows you to shape the output of the physical interface, so that the interface transmits less traffic than it is physically capable of carrying.

If you do not configure a shaping rate on the physical interface, the default physical interface bandwidth is based on the channel bandwidth and the time slot allocation.

In general, the physical interface speed is the basis for calculating the various queue parameters for a physical interface such as delay buffer size, weighted round- robin (WRR) weight, drop profile, and so forth. However, when you apply a shaping rate by including the **shaping-rate** statement, the shaping rate on that physical interface becomes the basis for calculating all the queue parameters for that physical interface.

On T4000 routers with Type 5 FPCs, the shaping rate value for the physical interface must be a minimum of 292 Kbps. The maximum value of shaping rate is limited by the maximum transmission rate of the interface.

Related Documentation

- Configuring the Shaping Rate for Physical Interfaces on page 59

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 4: RTT Delay Buffers for IQ2 PICs

IQ2 PIC Type	Ingress Buffer (ms)	Egress Buffer (ms)
4–port Gigabit Ethernet (Type 1)	200	300

Table 4: RTT Delay Buffers for IQ2 PICs *(continued)*

IQ2 PIC Type	Ingress Buffer (ms)	Egress Buffer (ms)
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

CHAPTER 4

Hierarchical Schedulers

Hierarchical Schedulers Terminology

Hierarchical schedulers introduce some new terms into a discussion of CoS capabilities. They also use some familiar terms in 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, customer VLANs, or aggregated Ethernet interfaces. 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 5 on page 16 shows how the configuration of an interface set or logical interface affects the terminology of hierarchical scheduler nodes.

Table 5: 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).

Understanding Two-Level and Three-level Hierarchical CoS for Subscriber Interfaces on MX Series Routers

This topic describes hierarchical CoS on MX Series routers running Modular Port Concentrator/Modular Interface Card (MPC/MIC) or Enhanced Queuing Dense Port Concentrator (EQ DPC) interfaces.

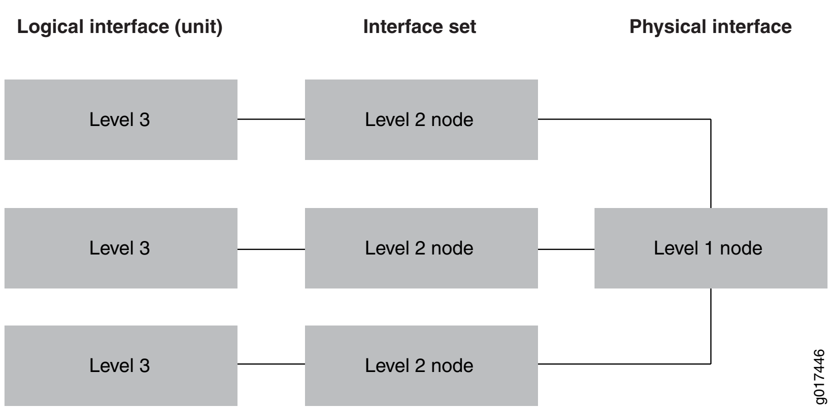
Hierarchical CoS enables you to apply traffic scheduling and queuing parameters and packet transmission scheduling parameters to an individual subscriber interface rather than to all interfaces configured on the port. Hierarchical CoS is supported on MX Series routers with either EQ DPCs or MPC/MICs installed.

On Juniper Networks MX Series routers, MPC/MIC and EQ DPC interfaces support a four-level CoS scheduling hierarchy that, when fully configured, consists of the physical interface (level 1), an interface set or underlying interface (level 2), one or more logical interfaces (level 3), and one or more queues (level 4). Although all CoS scheduling hierarchies are four-level, level 1 is always the physical interface and level 4 is always the queue. Hierarchical scheduling configurations consist of the type of interfaces you configure; for example, a logical interface or an interface set and where those interfaces reside in the scheduling hierarchy, either level 2 or level 3. Because many hierarchical scheduling configurations are possible, we use the terms *two-level hierarchical scheduling* and *three-level hierarchical scheduling* in this discussion.

Two-Level Hierarchical Scheduling

Two-level hierarchical scheduling limits the number of hierarchical levels in the scheduling hierarchy to two (level 2 and level 3) as shown in Figure 1 on page 17. In this configuration, interface sets are not configured and only the logical interfaces have traffic control profiles. Two-level hierarchical scheduling is supported on MX Series routers running either MPC/MIC or EQ DPC interfaces.

Figure 1: Two-Level Hierarchical Scheduling



In a two-level scheduling hierarchy, all logical interfaces and interface sets share a single level 2 node; no hierarchical relationship is formed.

You control two-level hierarchical scheduling by including the **maximum-hierarchy-levels** option under the **[edit interfaces *interface-name* hierarchical-scheduler]** statement:

- When the **maximum-hierarchy-levels** option is not set, interface sets can be at either level 2 or level 3, depending on whether the member logical interfaces within the interface set have a traffic control profile.
- If any member logical interface has a traffic control profile, then the interface set is always a level 2 CoS scheduler node.
- If no member logical interface has a traffic control profile, the interface set is always a level 3 CoS scheduler node.
- If the **maximum-hierarchy-levels** option is set, then the interface set can only be at level 3; it cannot be at level 2. In this case, if you configure a level 2 interface set, you generate Packet Forwarding Engine errors.

Table 6 on page 18 summarizes the interface hierarchy and the CoS scheduler node levels for two-level hierarchical scheduling.

Table 6: Two-Level Hierarchical Scheduling-Interface Hierarchy Versus Scheduling Nodes

Level 1	Level 2	Level 3	Level 4
Physical interface	–	Logical interfaces	One or more queues
Physical interface	–	Interface set	One or more queues
Physical interface	–	Logical interfaces	One or more queues

To configure two-level hierarchical scheduling, include the **hierarchical-scheduler** statement at the **[edit interfaces *interface-name*]** hierarchy level. You can optionally include the **maximum-hierarchy-levels** option. If you choose to set this option, the only supported value is 2.

```
[edit interfaces]
xe-2/0/0 {
  hierarchical-scheduler {
    maximum-hierarchy-levels 2;
  }
}
```

Three-Level Hierarchical Scheduling

Three-level hierarchical scheduling is supported only on MX Series routers running MPC/MIC interfaces. Three-level hierarchical scheduling has up to eight class of service queues. You can configure many different three-level scheduling hierarchies, depending on the location of the interface set and the use of underlying interfaces. In all variations, the physical interface is a level 1 CoS scheduler node and the queues reside at level 4.

NOTE: Three-level hierarchical scheduling is supported only on subscriber interfaces and interface sets running over aggregated Ethernet interfaces on MPC/MIC interfaces in MX Series routers.

When you use three-level hierarchical scheduling, interface sets can reside at either level 2 or level 3. You can also configure an underlying logic interface at level 2 and a logical interface at level 3. Table 7 on page 19 summarizes the most common cases of the interface hierarchy and the CoS scheduler node levels for three-level hierarchical scheduling.

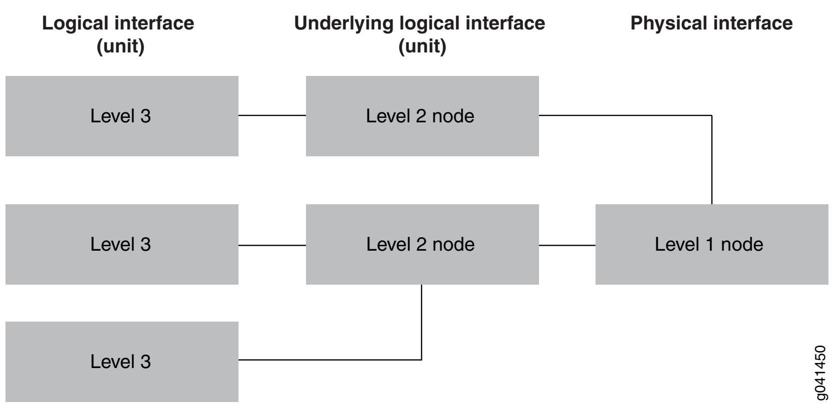
Table 7: Three-Level Hierarchical Scheduling-Interface Hierarchy Versus CoS Scheduling Node Levels

Level 1	Level 2	Level 3	Level 4
Physical interface	Interface set	Logical interfaces	One or more queues
Physical interface	Logical interface	Interface set	One or more queues
Physical interface	Underlying logical interface	Logical interfaces	One or more queues

In three-level hierarchical scheduling, the CoS scheduler nodes at level 1, level 2, and level 3 form a hierarchical relationship; this differs from two-level hierarchical scheduling where no hierarchical relationship is formed.

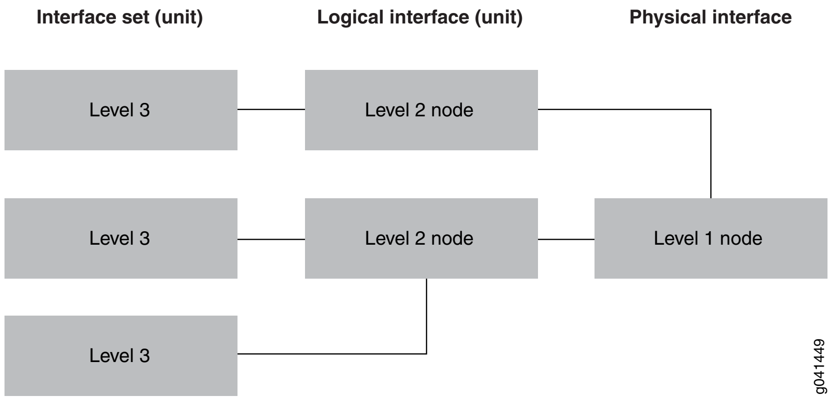
With a three-level hierarchical scheduling, logical interfaces can reside at level 2, or they can reside at level 3, if the logical interface at level 2 is an underlying logical interface. This is shown in Figure 2 on page 19.

Figure 2: Three-Level Hierarchical Scheduling: Logical Interfaces at Level 3 with Underlying Logical Interfaces at Level 2



Another possible configuration for three-level hierarchical scheduling is shown in Figure 3 on page 20. In this configuration, the logical interfaces are located at level 2 and the interface sets are located at level 3.

Figure 3: Three-Level Hierarchical Scheduling: Logical Interfaces at Level 2 with Interface Sets at Level 3



To configure three-level hierarchical scheduling, include the **implicit-hierarchy** option at the **[edit interfaces *interface-name* hierarchical-scheduler]** hierarchy level.

```
[edit interfaces]
xe-2/0/0 {
  hierarchical-scheduler {
    implicit-hierarchy;
  }
}
```

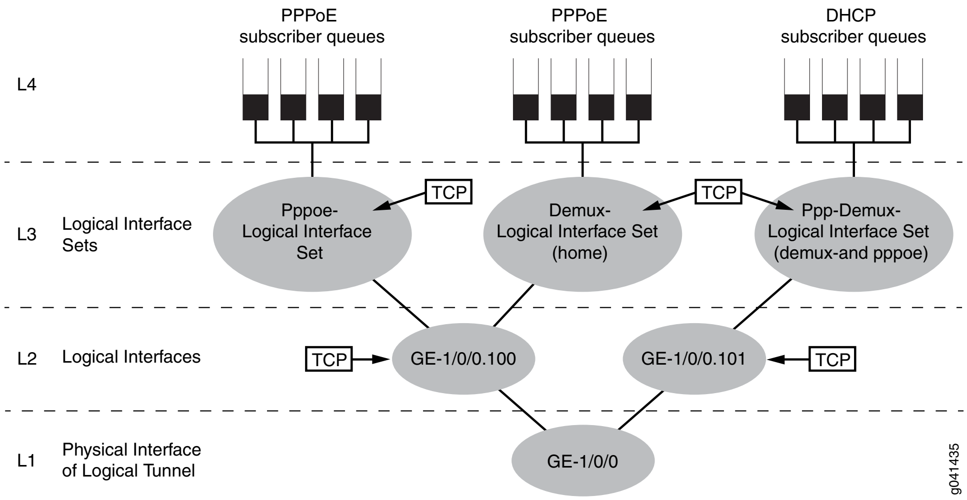
Interface Hierarchy Versus CoS Hierarchy

An interface hierarchy and a CoS scheduling hierarchy are distinctly different. Interface hierarchy refers to the relationship between the various interfaces; for example, the relationship between logical interfaces and an interface set, the relationship between a logical interface and an underlying logical interface, or the relationship between the physical interface and logical interface. CoS scheduling hierarchy refers to the hierarchical relationship between the CoS scheduler nodes. In two-level hierarchical scheduling, no hierarchy is formed between the CoS scheduler nodes; all logical interfaces and interface sets share a single level 2 scheduler node. However, when you use the **implicit-hierarchy** option for three-level hierarchical scheduling, the CoS scheduler nodes form a scheduling hierarchy.

Figure 4 on page 21 and Figure 5 on page 22 provide two scenarios for this discussion. Figure 4 on page 21 shows an interface hierarchy where a Gigabit Ethernet interface (GE-1/0/0) is the physical interface. Two logical interfaces (GE-1/0/0.100 and GE-1/0/0.101) are configured on the physical interface:

- Logical interface GE-1/0/0.100 is a member of a PPPoE interface set and a Demux interface set.
- Logical interface GE-1/0/0.101 is a member of a demux interface set.

Figure 4: Logical Interfaces at Level 2 and Interface Sets at Level 3



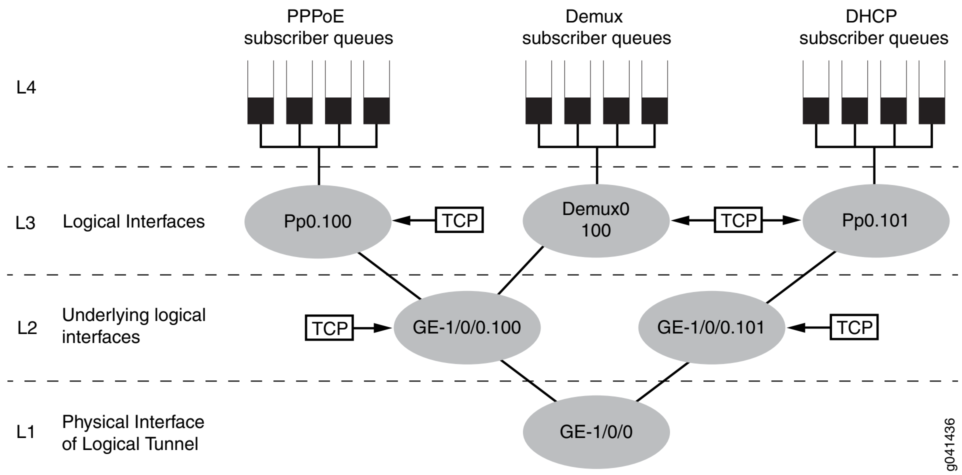
Each interface set has a dedicated queue. The CoS scheduler nodes at level 1 (physical interface), level 2 (underlying logical interfaces), and level 3 (interface sets) form a scheduling hierarchy.

To configure this scenario, you must include the **implicit-hierarchy** option under the **hierarchical-scheduler** statement on physical interface GE-1/0/0 and configure and apply traffic control profiles on each interface set and underlying logical interface.

Figure 5 on page 22 shows an interface hierarchy where Gigabit Ethernet interface GE-1/0/0 is the physical interface. Three logical interfaces are configured:

- Two logical interfaces (Pp0.100 and Demux0.100) reside on the underlying logical interface GE-1/0/0.100.
- A third logical interface (Pp0.101) resides on the underlying logical interface GE-1/0/0.101.

Figure 5: Logical Interfaces at Level 3 and Underlying Logical Interfaces at Level 2



Each logical interface has a dedicated queue. The CoS scheduler nodes at level 1 (physical interface), level 2 (underlying logical interfaces), and level 3 (logical interfaces) form a scheduling hierarchy.

To configure this scenario, you must include the **implicit-hierarchy** option under the **hierarchical-scheduler** statement on physical interface GE-1/0/0 and configure and apply traffic control profiles on each logical interface and underlying logical interface.

You can configure many different three-level scheduling hierarchies; Figure 4 on page 21 and Figure 5 on page 22 present just two possible scenarios. Table 7 on page 19 summarizes the possible interface locations and CoS scheduler nodes.

Related Documentation

- Configuring Hierarchical Schedulers for CoS on page 121
- *Configuring Hierarchical CoS for a Subscriber Interface of Aggregated Ethernet Links*
- *Configuring Hierarchical CoS on a Static PPPoE Subscriber Interface*
- *CoS Three-Level Hierarchical Scheduling on MPLS Pseudowire Subscriber Interfaces*
- **hierarchical-scheduler (Subscriber Interfaces on MX Series Routers) on page 150**

Interface Set Caveats

When configuring interface sets, consider the following guidelines:

- Interface sets can be defined in two major ways: as a list of logical interfaces or groups of aggregated Ethernet 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). You can configure sets of aggregated Ethernet interfaces on MIC or MPC interfaces only.
- You cannot specify an interface set mixing the logical interface, aggregated Ethernet, S-VLAN, or VLAN outer tag list forms of the **interface-set** statement.

- Keep the following guidelines in mind when configuring interface sets of logical interfaces over aggregated Ethernet:
  - Sets of aggregated Ethernet interfaces are supported on MIC and MPC interfaces only.
  - The supported interface stacks for aggregated Ethernet in an interface set include VLAN demux interfaces, IP demux interfaces, and PPPoE logical interfaces over VLAN demux interfaces.
  - The link membership list and scheduler mode of the interface set are inherited from the underlying aggregated Ethernet interface over which the interface set is configured.
  - When an aggregated Ethernet interface operates in link protection mode, or if the scheduler mode is configured to replicate member links, the scheduling parameters of the interface set are copied to each of the member links.
  - If the scheduler mode of the aggregated Ethernet interface is set to scale member links, the scheduling parameters are scaled based on the number of active member links and applied to each of the aggregated interface member links.
- 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;
   }
```



```
}
```

Related Documentation

- Configuring Interface Sets on page 123

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 60 and "Providing a Guaranteed Minimum Rate" on page 69. For more information on scheduler maps, see "Configuring Scheduler Maps" on page 89.

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.

Hierarchical Schedulers on Aggregated Ethernet Interfaces Overview

On MX series routers, you can apply hierarchical schedulers on aggregated ethernet bundles using interface sets. This feature enables you to configure a group of virtual LANs (VLANs) and control their bandwidth. This feature is supported at egress only.

You can configure interface sets for aggregated Ethernet (AE) interfaces created under static configurations. You can configure class-of-service parameters on AE interfaces, in either link-protect or non-link-protect mode. You can configure these parameters at the AE physical interface level. The CoS configuration is fully replicated for all AE member links in link-protect mode. You can control the way these parameters are applied to member links in non-link-protect mode by configuring the AE interface to operate in scaled mode or replicate mode.

The link membership list and scheduler mode of the interface set is inherited from the underlying aggregated Ethernet interface over which the interface set is configured. When an aggregated Ethernet interface operates in link protection mode, or if scheduler mode is configured to replicate member links, the scheduling parameters of the interface set are copied to each of the member links.

If the scheduler mode of the aggregated Ethernet interface is set to scale member links, the scheduling parameters are scaled based on the number of active member links (scaling factor is 1/A where A is the number of active links in the bundle) and applied to each of the AE interface member links.

To configure an interface set, include the **interface-set** statement at the **[edit class-of-service interfaces]** hierarchy level.

To apply scheduling and queuing parameters to the interface set, include the **output-traffic-control-profile** ***profile-name*** statement at the **[edit class-of-service interfaces** ***interface-name*** **interface-set** ***interface-set-name*****]** hierarchy level.

To apply an output traffic scheduling and shaping profile for the remaining traffic to the logical interface or interface set, include the **output-traffic-control-profile-remaining** ***profile-name*** statement at the **[edit class-of-service interfaces** ***interface-name*****]** hierarchy level or the **[edit class-of-service interfaces** ***interface-name*** **interface-set** ***interface-set-name*****]** hierarchy level.

Related Documentation

- Configuring Hierarchical Schedulers on Aggregated Ethernet Interfaces on page 122
- **output-traffic-control-profile-remaining on page 161**
- Controlling Remaining Traffic on page 125

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

Priority propagation is performed for MX Series router output Interfaces on Enhanced Queuing DPCs, MICs, and MPCs, and for M Series and T Series router output interfaces on IQ2E PICs. 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 serviced in the order of their priority. The default priority of a queue is low, and can explicitly configure a queue priority by including the **priority** statement at the **[edit class-of-service schedulers *scheduler-name*]** hierarchy level.

You cannot directly configure the priorities of all hierarchical scheduling elements. The priorities of internal nodes, for example, are determined as follows:

- 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 8 on page 27.

Table 8: 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 9 on page 27.

Table 9: 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

Table 9: Internal Node Queue Priority for CIR Mode *(continued)*

Configured Priority of Highest Active Child Node	Hardware Priority Below Guaranteed Rate	Hardware Priority Above Guaranteed Rate
Medium-high	1	3
Medium-low	1	3
Low	2	3

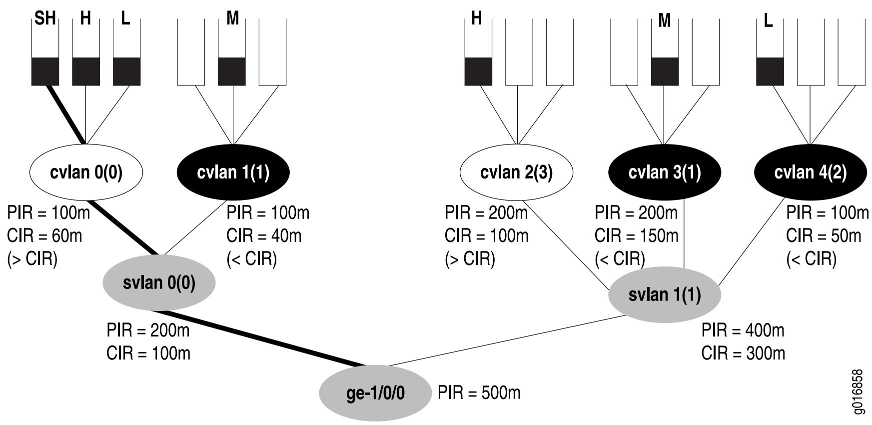
In PIR-only mode, nodes cannot send if they are above the configured shaping rate. The mapping between the configured priority and the hardware priority is for PIR-only mode is shown in Table 10 on page 28.

Table 10: Internal Node Queue Priority for PIR-Only Mode

Configured Priority	Hardware Priority
Strict-high	0
High	0
Medium-high	1
Medium-low	1
Low	2

A physical interface with hierarchical schedulers configured is shown in Figure 6 on page 29. 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 6: 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 9 on page 27 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).

Related Documentation

- *CoS on Enhanced IQ2 PICs Overview*
- *Enhanced Queuing DPC Hardware Properties*
- *CoS Limitations on MIC and MPC Interfaces*
- *Scheduler Node Scaling on MIC and MPC Interfaces Overview*
- priority (Schedulers) on page 165
- schedulers (Class of Service) on page 171

CHAPTER 5

Hierarchical Queues

Understanding Fine-Grained Queuing for Hierarchical Scheduling

This topic covers the following information:

CoS Scheduling with MIC and MPC Interfaces

Interfaces hosted on Modular Interface Card (MIC) and Modular Port Concentrator (MPC) line cards in MX Series 3D Universal Edge Routers support two types of class-of-service (CoS) queuing and scheduling, depending on MIC or MPC type:

Port-based Queuing MPCs

Port-level CoS features are available for interfaces hosted on MICs in MPC1 and MPC2 line cards and on ports on the fixed-configuration 16-port 10-Gigabit Ethernet MPC in MX240, MX480, MX960, MX2010, and MX2020 routers. For interfaces configured on these MPCs, CoS queuing is managed by the MQ chip.

The MQ chip interconnects with all other chips in the chipset, and its basic functionality includes managing packet data memory, interfacing with the ports, and interfacing with the packet header-processing chip and the WAN connectivity chip. On port-based queuing MPCs, the MQ chip also provides for the following port-based queuing capabilities:

- Up to 8 egress queues per port.
- Delay buffer capacities for 100 ms by default, and up to 200 ms maximum delay.
- Rate shaping of the ports and their queues.
- Guaranteed rate enforced at the queues.

Port-based queuing MPCs also support pre-classification of incoming packets to protect high priority packets in the event of congestion. such features include ingress DSCP rewrite and per-VLAN classification, ingress and egress policing, and rewrites.

Hierarchical Queuing MICs and MPCs

Hierarchical CoS features are available for interfaces hosted on MICs in MPC1 Q, MPC2 Q, and MPC2 EQ line cards in MX240, MX480, MX960, MX2010, and MX2020 routers and for interfaces hosted on 2-port or 4-port 10-Gigabit Ethernet MICs in MX5, MX10, MX40, or MX80 modular chassis routers. For interfaces configured on these MICs and MPCs, CoS queuing is managed by the QX chip.

Hierarchical queuing MICs and MPCs support all port-level CoS functionality plus fine-grained queuing abilities over four levels of hierarchical scheduling. The additional CoS functionality is provided by the dense queuing capabilities of the QX chip:

- Hierarchical scheduling with ports, interface-sets, and logical-interfaces.
- Shaping—Committed Information Rate (CIR) and a Peak Information Rate (PIR)—at all scheduling levels, including queues.
- Three normal priority levels and two excess priority levels configurable at all scheduling levels, including queues.
- Per-priority shaping of traffic at Level 1 or Level 2.
- Shaping for unconfigured customer VLANs (C-VLANs) and for service VLAN (S-VLANs).

Hierarchical Scheduling with Hierarchical Queuing MICs and MPCs

Whereas standard CoS scheduling is based on the scheduling and queuing characteristics of a router's egress ports and their queues, hierarchical CoS scheduling is based on the scheduling and queuing characteristics that span a hierarchy of *scheduler nodes* over a port. The hierarchy begins at Level 1, a *root node* at the physical interface (port) level of the CLI hierarchy and terminates at Level 4, a *leaf node* at the queue level. Between the root and leaf nodes of any scheduler hierarchy are one or more *internal nodes*, which are non-root nodes that have other nodes as “children” in the hierarchy.

Whereas you configure standard CoS scheduling by applying a *scheduler map* to each egress port to specify a forwarding class and a queue priority level, you configure hierarchical CoS scheduling with additional parameters. To configure hierarchical CoS scheduling, you apply a scheduler map to the queue level (Level 4) of a scheduler hierarchy, and you can apply a different *traffic control profile* at each of the other levels. A traffic control profile specifies not only a scheduler map (forwarding class and queue priority level) but also optional shaping rate (PIR), guaranteed transmit rate (CIR), burst rate, delay buffer rate, and drop profile.

Table 11 on page 33 illustrates the possible combinations of scheduler nodes and their corresponding node level designations in a hierarchical scheduling environment.

Table 11: Node Levels Designations in Hierarchical Scheduling

Scheduler Configuration for Hierarchical CoS	Hierarchical CoS Scheduler Nodes			
	Root Node	Internal (Non-Leaf) Nodes		Leaf Node
	Level 1	Level 2	Level 3	Level 4
One or more traffic control profiles configured on logical interfaces, but no interface-sets configured.	Physical interface	—	One or more logical interfaces	One or more queues
Interface-sets (collections of logical interfaces) configured, but no traffic-control profiles configured on logical interfaces.	Physical interface	—	Interface-set	One or more queues
Fully configured scheduler nodes:	Physical interface	Interface-set	One or more logical interfaces	One or more queues

For example, if an **interface-set** statement is configured with logical interfaces (such as **unit 0** and **unit 2**) and a queue, then the interface-set is an internal node at Level 2 of the scheduler node hierarchy. However, if there are no traffic control profiles configured on logical interfaces, then the interface set is at Level 3 of the hierarchy.

Related Documentation

- Jitter Reduction in Hierarchical CoS Queues on page 33
- *MX Series MPC Overview*
- *MPCs Supported by MX240, MX480, MX960, MX2010, and MX2020 Routers*
- *MX Series MIC Overview*
- *MICs Supported by MX Series Routers*
- *MX5, MX10, MX40, and MX80 Modular Interface Card Description*

Jitter Reduction in Hierarchical CoS Queues

Queue Jitter as a Function of the Maximum Number of Queues

Each QX chip on a Modular Interface Card (MIC) or Modular Port Concentrator (MPC) internally hosts a *rate wheel thread* that updates the *shaper credits* into the *shapers* available at each level of scheduling hierarchy. At each hierarchy level, the length of this update period determines two key characteristics of scheduling:

- The minimum buffer needed for the queue to pass packets without dropping.
- The degree of jitter encountered in the queue.

At each hierarchy level, the length of the rate wheel update period is dependent upon the number of entities enabled for that node level. Because traffic is queued at Level 4 (queues) and scheduled upwards to Level 1 (the port), the number of entities (queues) enabled at Level 4 determines the number of entities (logical interfaces, interface-sets, or ports) enabled at the other levels of the scheduling hierarchy. By extension, the number of queues enabled for a given scheduler node hierarchy determines the length of the update period at all hierarchy levels. Consequently, limiting the maximum number of queues supported by a hierarchical queuing MIC or MPC can reduce jitter in the queues.

Default Maximum Queues for Hierarchical Queuing MICs and MPCs

The QX chip on a MIC or MPC consists of two symmetrical halves, and each half supports a maximum of 64 K queues (128 K queues per QX chip). The 2-port and 4-port 10-Gigabit Ethernet MICs with XFP and the MPC1_Q line cards have one chipset and can support a maximum of 128 K queues, distributed across the two partitions of the single QX chip. The MPC2 Q and MPC2 EQ line cards have two chipsets and can support a maximum of 256 K queues, distributed across the four partitions of the two QX chips.

Table 12 on page 34 lists the maximum number of queues supported by default and the corresponding rate wheel update period for each hierarchical queuing MIC or MPC.

Table 12: Default Maximum Queues and Corresponding Rate Wheel Update Periods

Router Model	Hierarchical Queuing MIC or MPC	Maximum Queues	Rate Wheel Update Period
MX5, MX10, MX40, and MX80 modular	**2-port or 4-port 10-Gigabit Ethernet MIC with XFP** The chassis base board hosts one chipset-based Packet Forwarding Engine process that operates in standalone mode. The single QX chip is composed of two partitions that each support 64 K queues for egress ports.	128 K	1.6 ms
MX240, MX480, MX960, MX2010, and MX2020	**MPC1 Q** The MPC1 Q line card hosts one chipset-based Packet Forwarding Engine process that operates in fabric mode. The single QX chip is composed of two partitions that each support 64 K queues for egress ports.	128 K	1.6 ms
	MPC2 Q The MPC2 Q line card hosts two chipset-based Packet Forwarding Engine processes that operate in fabric mode. The two QX chips are composed of four partitions that each support 64 K queues for egress ports.	256 K	1.6 ms
	MPC2 EQ The MPC2 EQ line card hosts two chipset-based Packet Forwarding Engine processes that operate in fabric mode. The two QX chips are composed of four partitions that each support 64 K queues for egress ports.	256 K	2.6 ms

You can configure hierarchical queuing MICs and MPCs to support a reduced maximum number of queues. Doing so reduces the rate wheel update period used by the QX chip, which in turn reduces jitter in the queues for the egress interfaces hosted on the line card.

Shaping Rate Granularity as a Function of the Rate Wheel Update Period

Reducing the length of the QX chip rate wheel update period, in addition to reducing jitter in the hierarchical scheduling queues, also indirectly increases the shaping granularity.

For a given port line rate and scheduling hierarchy level, the shaping granularity is a function of the minimum shaper credit size and the rate wheel update period in effect as a result of the number of queues supported by the line card.

shaping granularity = minimum shaper credit size / rate wheel update period

Table 13 on page 35 shows how shaping granularity is calculated for (non-enhanced) hierarchical queuing MIC and MPC line cards with default values for minimum shaper credit size and for rate wheel update period.

Table 13: Default Shaping Granularities on (Non-Enhanced) Queuing MICs and MPCs

Port Type	Hierarchy Level	(Non-Enhanced) Queuing MIC or MPC Defaults		Calculation of Shaping Granularity
		Minimum Credit	Update Period	
1 Gbps Queuing	Level 1 (port), Level 4 (queues)	4 bytes = 32 bits	13.33 ms = 0.01333 sec	32 bits / 0.01333 sec = 2.4 Kbps
	Level 2, Level 3	16 bytes = 128 bits	1.66 ms = 0.00166 sec	128 bits / 0.01333 sec = 9.6 Kbps
10 Gbps Queuing	Level 1 (port), Level 4 (queues)	16 bytes = 128 bits	13.33 ms = 0.01333 sec	128 bits / 0.01333 sec = 9.6 Kbps
	Level 2, Level 3	64 bytes = 512 bits	1.66 ms = 0.00166 sec	512 bits / 0.01333 sec = 38.4 Kbps

Table 14 on page 35 lists shaping granularities on (non-enhanced) hierarchical queuing MIC and MPC line cards with default minimum shaper credit size and all values for maximum queues per card.

Table 14: All Shaping Granularities on (Non-Enhanced) Queuing MICs and MPCs

Port Type	Hierarchy Level	All Shaping Granularities on (Non-Enhanced) Hierarchical Queuing MIC or MPC				
		128 K	64 K	32 K	16 K	8 K
1 Gbps Queuing	Level 1 (port), Level 4 (queues)	2.4 Kbps	4.8 Kbps	9.6 Kbps	19.2 Kbps	38.4 Kbps
	Level 2, Level 3	9.6 Kbps	19.2 Kbps	38.4 Kbps	76.8 Kbps	153.6 Kbps
10 Gbps Queuing	Level 1 (port), Level 4 (queues)	9.6 Kbps	19.2 Kbps	38.4 Kbps	76.8 Kbps	153.6 Kbps
	Level 2, Level 3	38.4 Kbps	76.8 Kbps	153.6 Kbps	307.2 Kbps	614.4 Kbps

Related Documentation

- Understanding Fine-Grained Queuing for Hierarchical Scheduling on page 31
- Example: Reducing Jitter in Hierarchical CoS Queues on page 134
- **max-queues on page 158**

PART 2

Configuration

CHAPTER 6

Configuration for Schedulers

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 40
- Configuring Drop Profile Maps for Schedulers on page 51
- Configuring Scheduler Transmission Rate on page 51
- Configuring Schedulers for Priority Scheduling on page 55

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 15 on page 40.

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 routing device type, as shown in Table 15 on page 40.
- 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 transmission rate of the queue by the configured temporal value. The buffer size temporal value per queue differs by routing device type, as shown in Table 15 on page 40. 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 41.

Table 15: Buffer Size Temporal Value Ranges by Routing Device Type

Routing Devices	Temporal Value Ranges
M320 and T Series router FPCs, Type 1 and Type 2	1 through 80,000 microseconds

Table 15: Buffer Size Temporal Value Ranges by Routing Device Type *(continued)*

Routing Devices	Temporal Value Ranges
M320 and T Series router FPCs, Type 3. All ES cards (Type 1, 2, 3, and 4).	1 through 50,000 microseconds For PICs with greater than 40 Gbps of total bandwidth, the maximum temporal buffer size that can be configured for a scheduler is 40,000 microseconds instead of 50,000 microseconds.
M120 router FEBs and MX Series router nonenhanced Queuing DPCs, and EX Series switches	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
PTX Series Packet Transport Routers	1 through 100,000 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
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 41
- Enabling and Disabling the Memory Allocation Dynamic per Queue on page 49

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 16 on page 42 shows some recommended buffer sizes needed to absorb typical burst sizes from various upstream interface types.

Table 16: 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 17 on page 42.

Table 17: 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

Table 17: Maximum Delay Buffer with q-pic-large-buffer Enabled by Interface *(continued)*

Platform, PIC, or Interface Type	Maximum Buffer Size
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
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 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 103.

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 19 on page 45. To calculate specific buffer sizes for various *N*xDS0 interfaces, see "Maximum Delay Buffer for NxDS0 Interfaces" on page 45.

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 18 on page 44.

Table 18: 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

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;
		}

Table 18: Delay-Buffer Calculations *(continued)*

Delay Buffer Configuration	Formula	Example
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 45
- Example: Configuring Large Delay Buffers for Slower Interfaces on page 47

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 19 on page 45 shows the delay-buffer calculations for 1xDS0 through 32xDS0 interfaces.

Table 19: 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

Table 19: NxDS0 Transmission Rates and Delay Buffers *(continued)*

Interface Speed	Delay Buffer Size
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
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

Table 19: NxDS0 Transmission Rates and Delay Buffers *(continued)*

Interface Speed	Delay Buffer Size
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 18 on page 44.

```
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 19 on page 45.

```
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 3D Universal Edge Routers, and T Series Core Routers and EX Series switches 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 and EX Series switches' FPCs and Packet Forwarding Engines support MAD. No Modular Port Concentrators (MPCs) and 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;
    }
  }
}
```

Related Documentation

- buffer-size (Schedulers) on page 142

- schedulers (Class of Service) on page 171
- *q-pic-large-buffer*
- schedulers (Class of Service) on page 171

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*.

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*.

NOTE: **On Juniper Network MX Series 3D Universal Edge Routers, T4000 Core Routers, EX Series switches, and PTX Series Packet Transport Routers, you can configure only the any option for the protocol statement.**

For each scheduler, you can configure separate drop profile maps for each loss priority.

You can configure a maximum of 32 different drop profiles.

Related Documentation

- *Configuring RED 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 55.

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 and EX Series switch 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 3D Universal Edge Routers.
- The configuration of the **transmit-rate percent 0 exact** statement at the **[edit class-of-service schedulers** ***scheduler-name*****]** hierarchy is ineffective on T4000 routers with Type 5 FPCs.

- **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 39.

For more information about scheduler transmission rate, see the following sections:

- Example: Configuring Scheduler Transmission Rate on page 53
- Allocation of Leftover Bandwidth on page 54

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 55.)

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, the M320 Enhanced Type 4 FPCs, and T4000 routers with Type 5 FPCs and EX Series switches, 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, and EX Series switches, share excess bandwidth in the ratio of the transmit rates, but you can adjust this distribution.

Related Documentation

- Configuring Schedulers for Priority Scheduling on page 55
- Schedulers Overview on page 3
- *Configuring a Scheduler*
- **excess-rate on page 146**
- **schedulers on page 171**

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 10.

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 on page 55
- Configuring Strict-High Priority on M Series and T Series Routers on page 56

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.

Related Documentation

- Schedulers Overview on page 3
- Platform Support for Priority Scheduling on page 10

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 89.

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;
    }
}
```

Configuring the Shaping Rate for Physical Interfaces

To configure the shaping rate on the physical interface, either include the **shaping-rate** statement at the **[edit class-of-service interfaces *interface-name*]** hierarchy level or include the **output-traffic-control-profile** statement at the **[edit class-of-service interfaces *interface-name*]** hierarchy level.

You can specify a peak bandwidth rate in bps, either as a complete decimal number or as a decimal number followed by the abbreviation **k** (1000), **m** (1,000,000), or **g**

(1,000,000,000). For physical interfaces, the range is from 1000 through 160,000,000,000 bps.

For physical interfaces on T4000 routers with Type 5 FPCs, the shaping rate value for the physical interface must be a minimum of 292 Kbps. The maximum value of **shaping-rate** is limited by the maximum transmission rate of the interface.

The following are two example configurations for applying a shaping rate of 5 Gbps on a T4000 12x10 Gbps physical interface (xe-4/0/0):

Applying a shaping rate at the [edit class-of-service interfaces *interface-name*] hierarchy:

```
[edit class-of-service]
interfaces {
   xe-4/0/0 {
      shaping-rate 5g;
   }
}
```

Applying a shaping rate using traffic-control-profiles:

```
[edit class-of-service]
traffic-control-profiles {
   output {
      shaping-rate 5g;
   }
}
interfaces {
   xe-4/0/0 {
      output-traffic-control-profile output;
   }
}
```

To view the results of your configuration, issue the following **show** commands:

- **show class-of-service interface *interface-name***
- **show interfaces *interface-name* extensive**

Related Documentation

- Applying a Shaping Rate to Physical Interfaces Overview on page 13

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.15 and FRF.16 link services IQ (LSQ) interfaces on AS PICs, Multiservices PICs, and Multiservices DPCs, you can oversubscribe interface bandwidth. This means that the logical interfaces (and DLCIs within an FRF.15 or FRF.16 bundle) can be oversubscribed when there is leftover bandwidth. In the case of FRF.16 bundle interfaces, the physical interface can be oversubscribed. The oversubscription is capped to the configured PIR. Any unused bandwidth is distributed equally among oversubscribed logical interfaces or DLCIs, or physical interfaces.

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 103.

When configuring oversubscription for FRF.16 bundle interfaces, you can assign traffic control profiles that apply on a physical interface basis. When you apply traffic control profiles to FRF.16 bundles at the *logical* interface level, member link interface bandwidth is underutilized when there is a small proportion of traffic or no traffic at all on an individual DLCI. Support for traffic control features on the FRF.16 bundle physical interface level addresses this limitation.

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*);

   NOTE: When configuring oversubscription for FRF.16 bundle interfaces on a physical interface basis, you *must* specify shaping-rate as a percentage.

   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 router and EX Series switch 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 69.

For more information about Gigabit Ethernet IQ2 PICs, see *CoS on Enhanced IQ2 PICs Overview*.

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:

   NOTE: When configuring oversubscription for FRF.16 bundle interfaces on a physical interface basis, you *must* specify **delay-buffer-rate** as a percentage.

   ```
   [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 41 and "Maximum Delay Buffer for NxDS0 Interfaces" on page 45. For an example showing how the delay-buffer rates are applied, see "Examples: Oversubscribing Interface Bandwidth" on page 66.

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 39 and "Configuring Scheduler Maps" on page 89.

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 41.

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 enable scheduling for FRF.16 bundles physical interfaces, include the **no-per-unit-scheduler** statement at the **[edit interfaces *interface-name*]** hierarchy level:

   [edit interfaces *interface-name*]
   no-per-unit-scheduler;

7. To apply the traffic-scheduling profile , 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 20 on page 64 shows how the bandwidth and delay buffer are allocated in various configurations.

Table 20: 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 41. Unshaped logical interfaces receive the remaining bandwidth and a delay buffer in proportion to the remaining bandwidth.
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.

Table 20: 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 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 41. 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.
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 45.

```
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 with Scheduling Based on the Logical 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;
      }
    }
  }
}
```

Oversubscribing an LSQ Interface with Scheduling Based on the Physical Interface

Apply a traffic-control profile to the physical interface representing an FRF.16 bundle:

```
interfaces {
  lsq-0/2/0:0 {
    no-per-unit-scheduler;
    encapsulation multilink-frame-relay-uni-nni;
    unit 0 {
      dlci 100;
      family inet {
        address 18.18.18.2/24;
      }
    }
  }
class-of-service {
  traffic-control-profiles {
    rlsq_tc {
      scheduler-map rlsq;
```

```
            shaping-rate percent 60;
            delay-buffer-rate percent 10;
        }
    }
    interfaces {
        lsq-0/2/0:0 {
            output-traffic-control-profile rlsq_tc;
        }
    }
}
scheduler-maps {
    rlsq {
        forwarding-class best-effort scheduler rlsq_scheduler;
        forwarding-class expedited-forwarding scheduler rlsq_scheduler1;
    }
}
schedulers {
    rlsq_scheduler {
        transmit-rate percent 20;
        priority low;
    }
    rlsq_scheduler1 {
        transmit-rate percent 40;
        priority high;
    }
}
```

On an FRF.15 bundle, apply the following configuration:

```
class-of-service {
    traffic-control-profiles {
        rlsq {
            scheduler-map sched_0;
            shaping-rate percent 40;
            delay-buffer-rate percent 50;
        }
    }
    interfaces lsq-2/0/0 {
        unit 0 {
            output-traffic-control-profile rlsq;
        }
    }
}
interfaces lsq-2/0/0 {
    per-unit-scheduler;
    unit 0 {
        encapsulation multilink-frame-relay-end-to-end;
        family inet {
            address 10.1.1.2/32;
        }
    }
}
```


Providing a Guaranteed Minimum Rate

On Gigabit Ethernet IQ PIC, EQ DPC, MIC, MPC, and Channelized IQ PIC interfaces, and on 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*) <burst-size *bytes*>;

   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*.

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 41 and "Maximum Delay Buffer for NxDS0 Interfaces" on page 45. For an example showing how the delay-buffer rates are applied, see "Example: Providing a Guaranteed Minimum Rate" on page 72.

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 39 and "Configuring Scheduler Maps" on page 89.

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 41.

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 21 on page 71 shows how the bandwidth and delay buffer are allocated in various configurations.

Table 21: 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 41.
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 41.

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 45.

```
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;
    }
  }
}
```

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.

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;
  }
}
```


Related Documentation

- *Forwarding Classes and Fabric Priority Queues*

CHAPTER 7

Configuration for Queue-Level Bandwidth Sharing

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.

NOTE: Rate limiting is implemented differently on Enhanced Queuing DPCs and non-queuing Packet Forwarding Engines. On Enhanced Queuing DPCs, rate-limiting is implemented using a single rate two color policer. On non-queuing Packet Forwarding Engines, rate-limiting is achieved by shaping the queue to the transmit rate and keeping the queue delay buffers small to prevent too many packets from being queued once the shaping rate is reached.

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 22 on page 77. 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 22: 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 23 on page 77. 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 23: 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
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 24 on page 77. 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 24: 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 25 on page 78. 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 25: 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 26 on page 78. 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 26: 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 27 on page 78. 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 27: 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

Table 27: Sharing with the Same Priority Levels *(continued)*

Queue	Guaranteed (Transmit) Rate	Excess Rate	Guaranteed Priority	Excess Priority	Offered Load	Expected Throughput
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 28 on page 79. 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 28: 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
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 29 on page 79. 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 29: 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 30 on page 80. 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 30: 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 8

Configuration for Schedulers on Specific Interface Types

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 map 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 31 on page 84.

Table 31: 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

Table 31: Scheduler Allocation for an Ethernet IQ2 PIC *(continued)*

Ethernet IQ2 PIC Port	Number of Allocated Schedulers
3	128
4	128
5	128
6	128
7	152 (128 plus the 24 remaining that cannot be meaningfully allocated to other ports)

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*.

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]

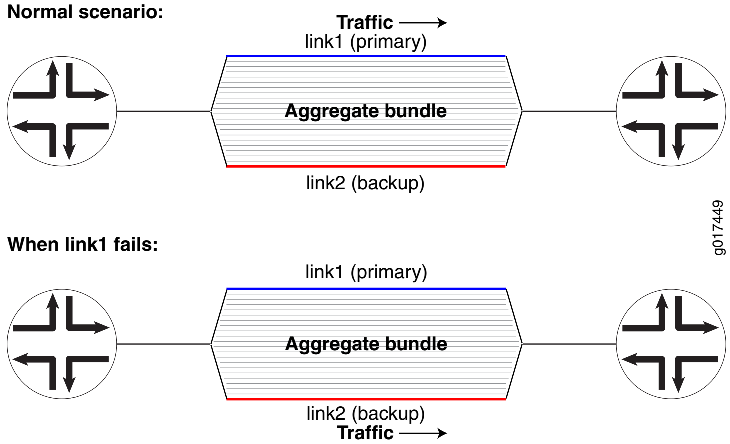

```
scheduler0 {
  transmit-rate percent 10 rate-limit;
  priority strict-high;
  excess-rate percent 30;
}
scheduler1 {
  transmit-rate percent 1m rate-limit;
  priority high;
  excess-rate percent 70;
}

[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;
}
```

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 7 on page 87 shows how the flow of traffic changes from primary to backup when the primary link in an aggregate bundle fails.

Figure 7: 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;
    }
  }
}
```

CHAPTER 9

Configuration for Scheduler Maps

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 to Physical Interfaces

After you have defined a scheduler map, as described in "Configuring Scheduler Maps" on page 89, 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 routing device 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 103.

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

When you apply a scheduler map to a physical interface, or when you modify the configuration of a scheduler map that is already applied to a physical interface, packets already in the output queues of the interface might get dropped. The amount of packet loss is not deterministic and depends on the offered traffic load at the time you apply or modify the scheduler map.

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 Operational Mode Commands*.

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 89. For information about logical interface scheduling configuration, see "Applying Scheduler Maps and Shaping Rate to DLCIs and VLANs" on page 103.

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 92. For general information about wildcards, see the *Junos OS Administration Library for Routing Devices*.

NOTE: **The interface applies wildcard configuration only if you do not add any specific configuration. If you add the specific interface configuration, then the interface deletes the applied wildcard configuration and applies the specified configuration.**

For more information, see the following sections:

- Applying Custom Schedulers to Packet Forwarding Component Queues on page 91
- Examples: Scheduling Packet Forwarding Component Queues on page 92

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;
```

When you apply a custom scheduler map to packet forwarding component queues, or when you modify the configuration of a custom scheduler map that is already applied to packet forwarding component queues, packets already in the chassis queues might be dropped. The amount of packet loss is not deterministic and depends on the offered traffic load at the time you apply or modify the custom scheduler map.

For information about defining the scheduler map referenced by ***map-name***, see "Configuring Scheduler Maps" on page 89.

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*.

```
[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.

CHAPTER 10

Configuration for Scheduler Maps on Specific Interface Types

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 103. 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*.

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

For shaping rate and WRR, the information included in the calculations varies by PIC type, as shown in Table 32 on page 99.

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*.

Table 32: 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;
    }
}
```

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 Packet Forwarding Engine and will not show any difference.

Logical interface scheduling (also called *per-unit scheduling*) allows you to enable multiple output queues on a logical interface and associate customized output scheduling and shaping for each queue.

NOTE: Ingress scheduling does not support logical interface scheduling.

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

You can configure logical interface scheduling on the following MICs and MPCs:

- 16x10GE MPC
- MPC3E:
  - 10x10GE MIC with SFP+
  - 2x40GE MIC with QSFP+
  - 1x100GE MIC with CXP

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 60.

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*.

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 33 on page 105 shows the interfaces/PICs that support fine-grained queuing and scheduling.

Table 33: Fine-Grained Queuing and Scheduling Support by Interfaces/PIC Type

Interface Type	PIC Type	Supported	Example Configuration
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 33: Fine-Grained Queuing and Scheduling Support by Interfaces/PIC Type *(continued)*

Interface Type	PIC Type	Supported	Example Configuration
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	No
	ATM2 IQ PIC with LLC/SNAP encapsulation	No	No
	Channelized OC12 IQ PIC with PPP encapsulation	No	No
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;
Channelized interfaces (except the Channelized OC12 PIC)	Channelized STM1	No	No

Table 33: Fine-Grained Queuing and Scheduling Support by Interfaces/PIC Type *(continued)*

Interface Type	PIC Type	Supported	Example Configuration
Logical interfaces	Fast Ethernet	No	No
	Gigabit Ethernet	No	No
	ATM1	No	No
	Channelized OC12	No	No

Table 34 on page 107 shows the MICs and MPCs that support fine-grained queuing and scheduling.

Table 34: Fine-Grained Queuing and Scheduling Support by MICs or MPCs Type

MPC	MIC	Supported	Example Configuration
Fixed Configuration MPCs			
16x10GE MPC	No	Yes	Example of supported configuration: [edit class-of-service interfaces ge-0/0/0 unit 1] scheduler-map map-1;
MPCs			
MPC1	No	No	No
MPC1E	No	No	No
MPC1 Q	Any supported MIC	Yes	Example of supported configuration: [edit class-of-service interfaces ge-0/0/0 unit 1] scheduler-map map-1;
MPC1E Q	Any supported MIC	Yes	Example of supported configuration: [edit class-of-service interfaces ge-0/0/0 unit 1] scheduler-map map-1;
MPC2	No	No	No
MPC2E	No	No	No
MPC2 Q	Any supported MIC	Yes	Example of supported configuration: [edit class-of-service interfaces ge-0/0/0 unit 1] scheduler-map map-1;
MPC2E Q	Any supported MIC	Yes	Example of supported configuration: [edit class-of-service interfaces ge-0/0/0 unit 1] scheduler-map map-1;

Table 34: Fine-Grained Queuing and Scheduling Support by MICs or MPCs Type *(continued)*

MPC	MIC	Supported	Example Configuration
MPC2 EQ	Any supported MIC	Yes	Example of supported configuration: [edit class-of-service interfaces ge-0/0/0 unit 1] scheduler-map map-1;
MPC2E EQ	Any supported MIC	Yes	Example of supported configuration: [edit class-of-service interfaces ge-0/0/0 unit 1] scheduler-map map-1;
MPC2E P	No	No	No
MPC3E	10-Gigabit Ethernet MIC with SFP+	Yes	Example of supported configuration: [edit class-of-service interfaces xe-0/0/0 unit 1] scheduler-map map-1;
	40-Gigabit Ethernet MIC with QSFP+	Yes	Example of supported configuration: [edit class-of-service interfaces et-0/0/0 unit 1] scheduler-map map-1;
	100-Gigabit Ethernet MIC with CXP	Yes	Example of supported configuration: [edit class-of-service interfaces et-0/0/0 unit 1] scheduler-map map-1;

To configure scheduling on logical interfaces:

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 including the **per-unit-scheduler** statement, you must also include the **vlan-tagging** statement or the **flexible-vlan-tagging** statement (to apply scheduling to VLANs) or the **encapsulation frame-relay** statement (to apply scheduling to DLCIs) at the **[edit interfaces *interface-name*]** hierarchy level.

   When you include this statement, the maximum number of VLANs supported is 768 on a single-port Gigabit Ethernet IQ PIC. On a dual-port Gigabit Ethernet IQ PIC, the maximum number is 384.

   See "Scaling of Per-VLAN Queuing on Non-Queuing MPCs" on page 118 for scaling information on non-queuing MPCs.

2. Associate a scheduler with the interface by including the **scheduler-map** statement at the **[edit class-of-service interfaces *interface-name* unit *logical-unit-number*]** hierarchy level:

   [edit class-of-service interfaces *interface-name* unit *logical-unit-number*]
   scheduler-map *map-name*;

Alternatively, associate a scheduler with the interface by including the **scheduler-map** statement at the **[edit class-of-service traffic-control-profiles *traffic control profile name*]** hierarchy level and then include the **output-traffic-control-profile** statement at the **[edit class-of-service interfaces *interface name* unit *logical unit number*]** hierarchy level.

[edit class-of-service traffic-control-profiles *traffic control profile name*]
scheduler-map *map-name*;

[edit class-of-service interfaces *interface-name* unit *logical-unit-number*]
output-traffic-control-profile *traffic-control-profile-name*;

3. Configure shaping on the interface by including the **shaping-rate** statement at the **[edit class-of-service interfaces *interface-name* unit *logical-unit-number*]** hierarchy level:

   [edit class-of-service interfaces *interface-name* unit *logical-unit-number*]
   shaping-rate *rate*;

   NOTE: You can also apply the shaping rate to the traffic control profile.

   By default, the logical interface bandwidth is the average of unused bandwidth for the number of logical interfaces that require default bandwidth treatment. You can specify a peak bandwidth rate in bps, either as a complete decimal number or as a decimal number followed by the abbreviation **k** (1000), **m** (1,000,000), or **g** (1,000,000,000). 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 the 16x10GE MPC, the minimum is 250,000 bps, and for the MPC3E, the minimum is 292,000 bps.

   For FRF.16 bundles on link services interfaces, only shaping rates based on percentage are supported.

Example: Applying Scheduler Maps and Shaping Rate to DLCIs

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 “Applying Scheduling and Shaping to VLANs” on page 111.

```
[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;
```

```
    }
  }
```

Related Documentation

- per-unit-scheduler on page 163
- Applying Scheduling and Shaping to VLANs on page 111

Applying Scheduling and Shaping to VLANs

This example shows how to apply schedulers to individual logical interfaces.

- Requirements on page 111
- Overview on page 111
- Configuration on page 111

Requirements

This example uses the following hardware and software components:

- Junos OS Release 7.4 or later running on router line cards that support Intelligent Queuing (IQ).
- Junos OS Release 13.2 or later running on MX Series routers containing 16x10GE MPC or MPC3E line cards.

Overview

By default, output scheduling is not enabled on logical interfaces. Logical interfaces without shaping configured share a default scheduler. *Logical interface scheduling* (also called *per-unit scheduling*) allows you to enable multiple output queues on a logical interface and associate customized scheduling and shaping for each queue.

To enable per-unit scheduling, include the **per-unit-scheduler** statement at the **[edit interfaces *interface name*]** hierarchy level. When per-unit schedulers are enabled, you can define dedicated schedulers for logical interfaces by including the **scheduler-map** statement at the **[edit class-of-service interfaces *interface name* unit *logical unit number*]** hierarchy level. Alternatively, you can include the **scheduler-map** statement at the **[edit class-of-service traffic-control-profiles *traffic control profile name*]** hierarchy level and then include the **output-traffic-control-profile** statement at the **[edit class-of-service interfaces *interface name* unit *logical unit number*]** hierarchy level.

This example shows how to define schedulers for logical interfaces through the use of traffic control profiles.

Configuration

CLI Quick Configuration

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

```
set interfaces xe-9/0/3 per-unit-scheduler
```



```
set interfaces xe-9/0/3 vlan-tagging
set interfaces xe-9/0/3 unit 1 vlan-id 101
set interfaces xe-9/0/3 unit 1 family inet address 1.1.1.1/24
set interfaces xe-9/0/3 unit 2 vlan-id 102
set interfaces xe-9/0/3 unit 2 family inet address 2.1.1.1/24
set class-of-service classifiers inet-precedence c8 forwarding-class be loss-priority low
  code-points 000
set class-of-service classifiers inet-precedence c8 forwarding-class ef loss-priority low
  code-points 001
set class-of-service classifiers inet-precedence c8 forwarding-class af loss-priority low
  code-points 010
set class-of-service classifiers inet-precedence c8 forwarding-class nc loss-priority low
  code-points 011
set class-of-service classifiers inet-precedence c8 forwarding-class be1 loss-priority low
  code-points 100
set class-of-service classifiers inet-precedence c8 forwarding-class ef1 loss-priority low
  code-points 101
set class-of-service classifiers inet-precedence c8 forwarding-class af1 loss-priority low
  code-points 110
set class-of-service classifiers inet-precedence c8 forwarding-class nc1 loss-priority low
  code-points 111
set class-of-service forwarding-classes queue 0 be
set class-of-service forwarding-classes queue 1 ef
set class-of-service forwarding-classes queue 2 af
set class-of-service forwarding-classes queue 3 nc
set class-of-service forwarding-classes queue 4 be1
set class-of-service forwarding-classes queue 5 ef1
set class-of-service forwarding-classes queue 6 af1
set class-of-service forwarding-classes queue 7 nc1
set class-of-service traffic-control-profiles tcp_ifd shaping-rate 2500000000
set class-of-service traffic-control-profiles tcp_ifd overhead-accounting bytes -20
set class-of-service traffic-control-profiles tcp_gold scheduler-map gold
set class-of-service traffic-control-profiles tcp_gold shaping-rate 2500000000
set class-of-service traffic-control-profiles tcp_gold overhead-accounting bytes -20
set class-of-service traffic-control-profiles tcp_gold guaranteed-rate 1g
set class-of-service traffic-control-profiles tcp_silver scheduler-map silver
set class-of-service traffic-control-profiles tcp_silver shaping-rate 1g
set class-of-service traffic-control-profiles tcp_silver overhead-accounting bytes -20
set class-of-service traffic-control-profiles tcp_silver guaranteed-rate 500m
set class-of-service interfaces xe-9/0/3 output-traffic-control-profile tcp_ifd
set class-of-service interfaces xe-9/0/3 unit 1 output-traffic-control-profile tcp_gold
set class-of-service interfaces xe-9/0/3 unit 2 output-traffic-control-profile tcp_silver
set class-of-service scheduler-maps gold forwarding-class be scheduler gold_internet
set class-of-service scheduler-maps gold forwarding-class ef scheduler gold_video
set class-of-service scheduler-maps gold forwarding-class af scheduler gold_voice
set class-of-service scheduler-maps gold forwarding-class nc scheduler gold_reserved
set class-of-service scheduler-maps silver forwarding-class be scheduler silver_internet
set class-of-service scheduler-maps silver forwarding-class ef scheduler silver_video
set class-of-service scheduler-maps silver forwarding-class af scheduler silver_voice
set class-of-service scheduler-maps silver forwarding-class nc scheduler silver_reserved
set class-of-service schedulers gold_internet excess-rate percent 40
set class-of-service schedulers gold_internet buffer-size percent 20
set class-of-service schedulers gold_internet priority low
set class-of-service schedulers gold_video transmit-rate percent 50
set class-of-service schedulers gold_video buffer-size percent 50
set class-of-service schedulers gold_voice shaping-rate percent 10
```



```
set class-of-service schedulers gold_voice buffer-size percent 10
set class-of-service schedulers gold_voice priority strict-high
set class-of-service schedulers gold_reserved excess-rate percent 20
set class-of-service schedulers gold_reserved buffer-size percent 10
set class-of-service schedulers gold_reserved priority low
set class-of-service schedulers silver_internet excess-rate percent 40
set class-of-service schedulers silver_internet buffer-size percent 20
set class-of-service schedulers silver_internet priority low
set class-of-service schedulers silver_video transmit-rate percent 50
set class-of-service schedulers silver_video buffer-size percent 50
set class-of-service schedulers silver_voice shaping-rate percent 10
set class-of-service schedulers silver_voice buffer-size percent 10
set class-of-service schedulers silver_voice priority strict-high
set class-of-service schedulers silver_reserved excess-rate percent 20
set class-of-service schedulers silver_reserved buffer-size percent 10
set class-of-service schedulers silver_reserved priority low
```

Step-by-Step Procedure

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

1. Configure the device interfaces.

```
[edit interfaces]
user@PE1# set xe-9/0/3 per-unit-scheduler
user@PE1# set xe-9/0/3 vlan-tagging
user@PE1# set xe-9/0/3 unit 1 vlan-id 101
user@PE1# set xe-9/0/3 unit 1 family inet address 1.1.1.1/24
user@PE1# set xe-9/0/3 unit 2 vlan-id 102
user@PE1# set xe-9/0/3 unit 2 family inet address 2.1.1.1/24
```

2. Configure the classifiers.

```
[edit class-of-service]
user@PE1# set classifiers inet-precedence c8 forwarding-class be loss-priority low
  code-points 000
user@PE1# set classifiers inet-precedence c8 forwarding-class ef loss-priority low
  code-points 001
user@PE1# set classifiers inet-precedence c8 forwarding-class af loss-priority low
  code-points 010
user@PE1# set classifiers inet-precedence c8 forwarding-class nc loss-priority low
  code-points 011
user@PE1# set classifiers inet-precedence c8 forwarding-class be1 loss-priority low
  code-points 100
user@PE1# set classifiers inet-precedence c8 forwarding-class ef1 loss-priority low
  code-points 101
user@PE1# set classifiers inet-precedence c8 forwarding-class af1 loss-priority low
  code-points 110
user@PE1# set classifiers inet-precedence c8 forwarding-class nc1 loss-priority low
  code-points 111
```

3. Configure the forwarding classes.

```
[edit class-of-service]
user@PE1# set forwarding-classes queue 0 be
user@PE1# set forwarding-classes queue 1 ef
user@PE1# set forwarding-classes queue 2 af
user@PE1# set forwarding-classes queue 3 nc
```

```
user@PE1# set forwarding-classes queue 4 be1
user@PE1# set forwarding-classes queue 5 ef1
user@PE1# set forwarding-classes queue 6 af1
user@PE1# set forwarding-classes queue 7 nc1
```

4. Configure the traffic control profiles.

```
[edit class-of-service]
user@PE1# set traffic-control-profiles tcp_ifd shaping-rate 2500000000
user@PE1# set traffic-control-profiles tcp_ifd overhead-accounting bytes -20
user@PE1# set traffic-control-profiles tcp_gold scheduler-map gold
user@PE1# set traffic-control-profiles tcp_gold shaping-rate 2500000000
user@PE1# set traffic-control-profiles tcp_gold overhead-accounting bytes -20
user@PE1# set traffic-control-profiles tcp_gold guaranteed-rate 1g
user@PE1# set traffic-control-profiles tcp_silver scheduler-map silver
user@PE1# set traffic-control-profiles tcp_silver shaping-rate 1g
user@PE1# set traffic-control-profiles tcp_silver overhead-accounting bytes -20
user@PE1# set traffic-control-profiles tcp_silver guaranteed-rate 500m
```

5. Map the traffic control profiles to their respective physical or logical interface.

```
[edit class-of-service]
user@PE1# set interfaces xe-9/0/3 output-traffic-control-profile tcp_ifd
user@PE1# set interfaces xe-9/0/3 unit 1 output-traffic-control-profile tcp_gold
user@PE1# set interfaces xe-9/0/3 unit 2 output-traffic-control-profile tcp_silver
```

6. Configure the scheduler maps.

```
[edit class-of-service]
user@PE1# set scheduler-maps gold forwarding-class be scheduler gold_internet
user@PE1# set scheduler-maps gold forwarding-class ef scheduler gold_video
user@PE1# set scheduler-maps gold forwarding-class af scheduler gold_voice
user@PE1# set scheduler-maps gold forwarding-class nc scheduler gold_reserved
user@PE1# set scheduler-maps silver forwarding-class be scheduler silver_internet
user@PE1# set scheduler-maps silver forwarding-class ef scheduler silver_video
user@PE1# set scheduler-maps silver forwarding-class af scheduler silver_voice
user@PE1# set scheduler-maps silver forwarding-class nc scheduler silver_reserved
```

7. Configure the schedulers.

```
[edit class-of-service]
user@PE1# set schedulers gold_internet excess-rate percent 40
user@PE1# set schedulers gold_internet buffer-size percent 20
user@PE1# set schedulers gold_internet priority low
user@PE1# set schedulers gold_video transmit-rate percent 50
user@PE1# set schedulers gold_video buffer-size percent 50
user@PE1# set schedulers gold_voice shaping-rate percent 10
user@PE1# set schedulers gold_voice buffer-size percent 10
user@PE1# set schedulers gold_voice priority strict-high
user@PE1# set schedulers gold_reserved excess-rate percent 20
user@PE1# set schedulers gold_reserved buffer-size percent 10
user@PE1# set schedulers gold_reserved priority low
user@PE1# set schedulers silver_internet excess-rate percent 40
user@PE1# set schedulers silver_internet buffer-size percent 20
user@PE1# set schedulers silver_internet priority low
user@PE1# set schedulers silver_video transmit-rate percent 50
user@PE1# set schedulers silver_video buffer-size percent 50
user@PE1# set schedulers silver_voice shaping-rate percent 10
user@PE1# set schedulers silver_voice buffer-size percent 10
```



```
user@PE1# set schedulers silver_voice priority strict-high
user@PE1# set schedulers silver_reserved excess-rate percent 20
user@PE1# set schedulers silver_reserved buffer-size percent 10
user@PE1# set schedulers silver_reserved priority low
```

Results

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

```
user@PE1# show interfaces
  interfaces {
    xe-9/0/3 {
      per-unit-scheduler;
      vlan-tagging;
      unit 1 {
        vlan-id 101;
        family inet {
          address 1.1.1.1/24;
        }
      }
      unit 2 {
        vlan-id 102;
        family inet {
          address 2.1.1.1/24;
        }
      }
    }
  }

user@PE1# show class-of-service
  class-of-service {
    classifiers {
      inet-precedence c8 {
        forwarding-class be {
          loss-priority low code-points 000;
        }
        forwarding-class ef {
          loss-priority low code-points 001;
        }
        forwarding-class af {
          loss-priority low code-points 010;
        }
        forwarding-class nc {
          loss-priority low code-points 011;
        }
        forwarding-class be1 {
          loss-priority low code-points 100;
        }
        forwarding-class ef1 {
          loss-priority low code-points 101;
        }
        forwarding-class af1 {
          loss-priority low code-points 110;
```

```
    }
    forwarding-class nc1 {
      loss-priority low code-points 111;
    }
  }
}
forwarding-classes {
  queue 0 be;
  queue 1 ef;
  queue 2 af;
  queue 3 nc;
  queue 4 be1;
  queue 5 ef1;
  queue 6 af1;
  queue 7 nc1;
}
traffic-control-profiles {
  tcp_ifd {
    shaping-rate 2500000000;
    overhead-accounting bytes -20;
  }
  tcp_gold {
    scheduler-map gold;
    shaping-rate 2500000000;
    overhead-accounting bytes -20;
    guaranteed-rate 1g;
  }
  tcp_silver {
    scheduler-map silver;
    shaping-rate 1g;
    overhead-accounting bytes -20;
    guaranteed-rate 500m;
  }
}
interfaces {
  xe-9/0/3 {
    output-traffic-control-profile tcp_ifd;
    unit 1 {
      output-traffic-control-profile tcp_gold;
    }
    unit 2 {
      output-traffic-control-profile tcp_silver;
    }
  }
}
scheduler-maps {
  gold {
    forwarding-class be scheduler gold_internet;
    forwarding-class ef scheduler gold_video;
    forwarding-class af scheduler gold_voice;
    forwarding-class nc scheduler gold_reserved;
  }
  silver {
    forwarding-class be scheduler silver_internet;
    forwarding-class ef scheduler silver_video;
    forwarding-class af scheduler silver_voice;
```

```
            forwarding-class nc scheduler silver_reserved;
        }
    }
    schedulers {
        gold_internet {
            excess-rate percent 40;
            buffer-size percent 20;
            priority low;
        }
        gold_video {
            transmit-rate percent 50;
            buffer-size percent 50;
        }
        gold_voice {
            shaping-rate percent 10;
            buffer-size percent 10;
            priority strict-high;
        }
        gold_reserved {
            excess-rate percent 20;
            buffer-size percent 10;
            priority low;
        }
        silver_internet {
            excess-rate percent 40;
            buffer-size percent 20;
            priority low;
        }
        silver_video {
            transmit-rate percent 50;
            buffer-size percent 50;
        }
        silver_voice {
            shaping-rate percent 10;
            buffer-size percent 10;
            priority strict-high;
        }
        silver_reserved {
            excess-rate percent 20;
            buffer-size percent 10;
            priority low;
        }
    }
}
```

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

Related Documentation

- Applying Scheduler Maps and Shaping Rate to DLCIs and VLANs on page 103
- per-unit-scheduler on page 163

Scaling of Per-VLAN Queuing on Non-Queuing MPCs

Support for per-VLAN (logical interface) queuing is introduced in Junos OS Release 13.2 on 16x10GE MPC and MPC3E line cards.

You can configure logical interface scheduling on the following MICs and MPCs:

- 16x10GE MPC
- MPC3E:
  - 10x10GE MIC with SFP+
  - 2x40GE MIC with QSFP+
  - 1x100GE MIC with CXP

To enable logical interface scheduling, you include the **per-unit-scheduler** statement at the **[edit interfaces *interface name*]** hierarchy level. When per-unit schedulers are enabled, you can define dedicated schedulers for logical interfaces by including the **scheduler-map** statement at the **[edit class-of-service interfaces *interface name* unit *logical unit number*]** hierarchy level. Alternatively, you can include the **scheduler-map** statement at the **[edit class-of-service traffic-control-profiles *traffic control profile name*]** hierarchy level and then include the **output-traffic-control-profile** statement at the **[edit class-of-service interfaces *interface name* unit *logical unit number*]** hierarchy level.

The following table shows the number of VLANs per port available in both 8-queue and 4-queue mode.

Table 35: Number of VLANs Per Port Available on Non-Queuing MPCs

MPC	MIC	VLANs/Port – 8-Queue Mode	VLANs/Port – 4-Queue Mode
16X10GE	No	21	44
MPC3E	10-Gigabit Ethernet MIC with SFP+ (10X10GE)	2	6
MPC3E	40-Gigabit Ethernet MIC with QSFP+ (2X40GE)	20	42
MPC3E	100-Gigabit Ethernet MIC with CXP (1X100GE)	20	42

Enabling per-unit schedulers on these interfaces adds additional output to the **show interfaces *interface name* [detail | extensive]** command. This additional output lists the maximum resources available and the number of configured resources for schedulers. Following is sample output showing the CoS scheduler resource information on a non-queuing line card:

`root@R1#` **run show interfaces et-2/2/0 detail**

```
Physical interface: et-2/2/0, Enabled, Physical link is Down
  Interface index: 165, SNMP ifIndex: 550, Generation: 168
  Link-level type: Ethernet, MTU: 1522, Speed: 100Gbps, BPDU Error: None, Loopback:
```

```
Disabled, Source filtering: Disabled,
 Flow control: Enabled
 Device flags   : Present Running Down
 Interface flags: Hardware-Down SNMP-Traps Internal: 0x4000
 Link flags     : Scheduler
 CoS queues     : 8 supported, 8 maximum usable queues
 Schedulers     : 0
 Hold-times     : Up 0 ms, Down 0 ms
 Current address: 80:71:1f:10:e6:b4, Hardware address: 80:71:1f:10:e6:b4
 Last flapped   : 2013-05-07 16:17:01 PDT (03:16:41 ago)
 Statistics last cleared: Never
 Traffic statistics:
  Input  bytes  :                  0                    0 bps
  Output bytes  :                  0                    0 bps
  Input  packets:                  0                    0 pps
  Output packets:                  0                    0 pps
  IPv6 transit statistics:
   Input  bytes  :                 0
   Output bytes  :                 0
   Input  packets:                 0
   Output packets:                 0
 Egress queues: 8 supported, 4 in use
 CoS scheduler resource information:
   Maximum units supported per MIC/PIC: 20
   Configured units per MIC/PIC: 1
   Maximum units allowed per port: 20
   Configured units on this port: 1
 Queue counters:       Queued packets  Transmitted packets      Dropped packets

   0 best-effort                    0                    0                    0

   1 expedited-fo                   0                    0                    0

   2 assured-forw                   0                    0                    0

   3 network-cont                   0                    0                    0

 Queue number:         Mapped forwarding classes
   0                   best-effort
```

If you enable more VLANs than the previously mentioned MPC/MIC combinations support, VLANs up to the supported numbers receive dedicated queuing resources. The additional VLANs share port queues. Scheduling for port queues cannot be controlled. However, port queues are guaranteed 1 percent of the physical interface bandwidth to avoid queue starving and buffer holdup.

NOTE: The persistency of queuing resources for a set of VLANs cannot be maintained through the following events:

- System reboot
- MPC or MIC restart
- A Routing Engine switch without graceful Routing Engine switchover (GRES)
- Disabling then re-enabling **per-unit scheduler**
- Deactivating then reactivating interfaces
- Deactivating then reactivating class-of-service (CoS) interfaces

Related Documentation

- per-unit-scheduler on page 163
- Applying Scheduler Maps and Shaping Rate to DLCIs and VLANs on page 103

CHAPTER 11

Configuration for Hierarchical Scheduling

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 3D Universal Edge 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.

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, you must enable hierarchical scheduling at the physical interface.

Related Documentation

- **hierarchical-scheduler on page 149**
- Hierarchical Schedulers Terminology on page 15
- *Scheduler Node Scaling on MIC and MPC Interfaces Overview*
- *CoS on Enhanced IQ2 PICs Overview*
- Understanding Two-Level and Three-level Hierarchical CoS for Subscriber Interfaces on MX Series Routers on page 17

Configuring Hierarchical Schedulers on Aggregated Ethernet Interfaces

The following example shows the creation of an interface set for aggregated Ethernet interfaces in a static Ethernet configuration.

To configure interface sets for aggregated Ethernet (AE) interfaces created under static configurations:

1. Create the AE interfaces.

   ```
   [edit]
   user@host# show chassis | display set
   set chassis aggregated-devices ethernet device-count 10
   ```

2. Configure the AE physical interfaces and member links.

   user@host# show interfaces | display set

   ```
   set interfaces ge-5/2/0 gigether-options 802.3ad ae0
   set interfaces ge-5/2/1 gigether-options 802.3ad ae0
   set interfaces ae0 hierarchical-scheduler maximum-hierarchy-levels 2
   set interfaces ae0 flexible-vlan-tagging
   set interfaces ae0 unit 0 vlan-id 100
   set interfaces ae0 unit 1 vlan-id 101
   set interfaces ae0 unit 2 vlan-id 102
   set interfaces ae0 unit 3 vlan-id 103
   set interfaces ae0 unit 4 vlan-id 104
   ```

3. Configure the interface set.

   ```
   set interfaces interface-set ifset1-ae0 interface ae0 unit 0
   set interfaces interface-set ifset1-ae0 interface ae0 unit 1
   ```

4. Configure class-of-service parameters for the interface sets.

set class-of-service interfaces interface-set ifset1-ae0 output-traffic-control-profile tcp

5. Configure scheduler mode.

   set class-of-service interfaces ae0 member-link-scheduler scale

Related Documentation

- Hierarchical Schedulers on Aggregated Ethernet Interfaces Overview on page 25

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 or aggregated Ethernet interfaces (**unit 100**, **unit 200**, and so on)
- At the stacked VLAN level using a list of outer VLAN IDs (**vlan-tags-outer 210**, **vlan-tags-outer 220**, and so on).

  The **svlan** ***number*** listing option with a single outer VLAN tag is a convenient way to specify 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, aggregated Ethernet interfaces, 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.

NOTE: 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 interface naming option lists aggregated Ethernet interfaces:

```
[edit interfaces]
  interface-set demuxset1 {
    interface demux0 {
      unit 1;
      ..
    }
  }
  demux0 {
    unit 1 {
      demux-options {
        underlying-interface ae0.1;
      }
      family inet {
        demux-source {
          100.1.1.1/24;
        }
        address 100.1.1.1/24;
      }
    }
    ..
    ae0 {
      unit 1 {
      }
      ..
    }
  }
  class-of-service {
    interface-set demuxset1 {
      output-traffic-control-profile tcp2;
```



```
        }
    }
}
```

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.

Related Documentation

- Interface Set Caveats on page 22

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;
}
```

Related Documentation

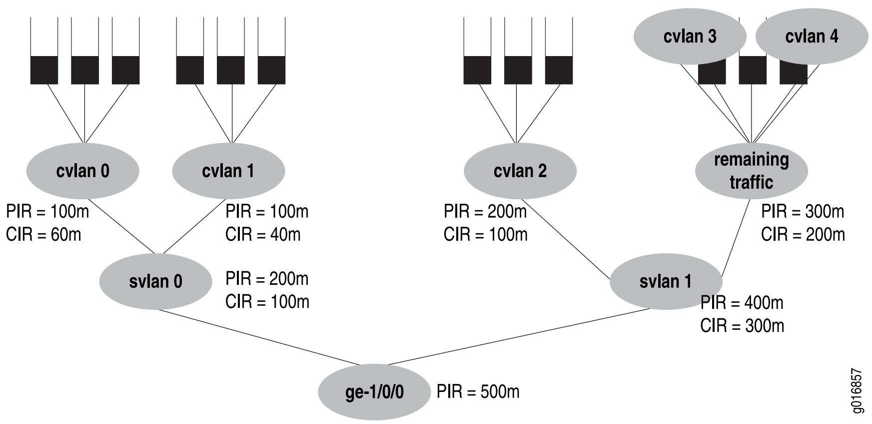
- output-traffic-control-profile on page 160

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 8 on page 126. 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 8: Handling Remaining Traffic



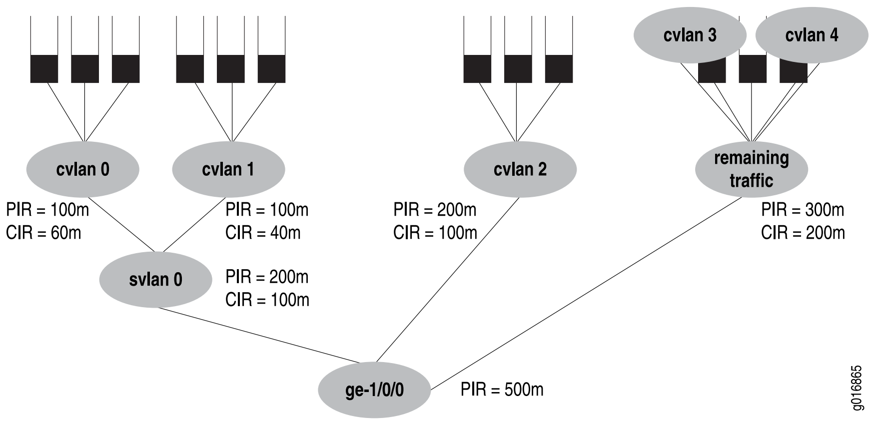
This example considers the case where customer VLANs 3 and 4 have no explicit traffic control profile, yet need to establish a shaping and guaranteed transmit rate for traffic heading for those customer VLANs. The solution is to add a traffic control profile to the **svlan1** interface set. This example 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 9 on page 127.

Figure 9: 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;
  }
}
```

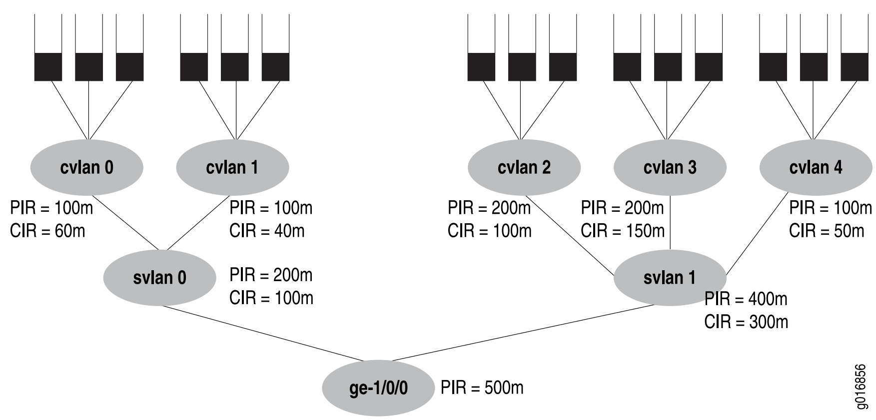
If an interface set has logical interfaces configured with a traffic control profile, then the use of the **internal-node** statement has no effect.

Internal nodes can specify a **traffic-control-profile-remaining** statement.

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 10 on page 129. The queues are shown at the top of the figure with the other three levels of the hierarchy below.

Figure 10: 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

```
[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*.

```
  [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 131.

```
  [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.

Example: Reducing Jitter in Hierarchical CoS Queues

This example shows how to reduce jitter in the output queues for VLAN ports hosted on a hierarchical queuing MPC.

Requirements

This example uses the following Juniper Networks hardware and Junos OS software:

- MX960 router in an IPv4 network and running Junos OS Release 13.2 or later.
- Available Gigabit Ethernet port hosted on FPC slot 2, PIC slot 0, port 0.
- Available Gigabit Ethernet port hosted on port 0 of a Gigabit Ethernet Modular Interface Card (MIC) in PIC slot 0 of an MPC2 Q Modular Port Concentrator (MPC) in FPC slot 5.

Before you begin configuring this example, make sure that the maximum number of queues allowed for the hierarchical queuing MPC in slot 5 has not yet been configured. When you enter the **show chassis fpc 5** command from configuration mode, the **max-queues** statement should not display.

Overview

In this example you configure hierarchical scheduling on a VLAN port hosted on a hierarchical queuing MPC. To reduce jitter in the queues for all egress ports hosted on the MPC, reduce the maximum number of queues allowed for MPC.

Configuration

CLI Quick Configuration

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

```
set interfaces xe-2/0/0 per-unit-scheduler
set interfaces xe-2/0/0 flexible-vlan-tagging
set interfaces xe-2/0/0 unit 0 vlan-id 1
set interfaces xe-2/0/0 unit 0 family inet address 1.1.1.1/24
set interfaces xe-2/0/0 unit * classifiers ieee-802.1 ieee_jitter
set interfaces xe-5/0/0 per-unit-scheduler
set interfaces xe-5/0/0 flexible-vlan-tagging
set interfaces xe-5/0/0 unit 0 vlan-id 1
set interfaces xe-5/0/0 unit 0 family inet address 2.1.1.1/24
set class-of-service-interfaces xe-5/0/0 unit * output-traffic-control-profile tcp
set class-of-service forwarding-classes queue 0 be
set class-of-service forwarding-classes queue 1 ef
set class-of-service forwarding-classes queue 2 af
set class-of-service forwarding-classes queue 3 nc
set class-of-service schedulers be_sch priority low
```

```
set class-of-service schedulers ef_sch priority low
set class-of-service schedulers af_sch priority strict-high
set class-of-service schedulers nc_sch priority low
set class-of-service classifiers ieee_jitter forwarding-class be loss-priority low code-points
  000
set class-of-service classifiers ieee_jitter forwarding-class ef loss-priority low code-points
  001
set class-of-service classifiers ieee_jitter forwarding-class af loss-priority low code-points
  010
set class-of-service classifiers ieee_jitter forwarding-class nc loss-priority low code-points
  011
set class-of-service scheduler-maps smap_jitter forwarding-class be scheduler be_sch
set class-of-service scheduler-maps smap_jitter forwarding-class ef scheduler ef_sch
set class-of-service scheduler-maps smap_jitter forwarding-class af scheduler af_sch
set class-of-service scheduler-maps smap_jitter forwarding-class nc scheduler nc_sch
set class-of-service traffic-control-profiles tcp scheduler-map smap_jitter
set class-of-service traffic-control-profiles tcp shaping-rate 6g
```

Baseline Configuration

Step-by-Step Procedure

Configure hierarchical scheduling at **xe-5.0.0**.

1. To configure the VLAN 1 input and output at **xe-2/0/0.0** and **xe-5/0/0.0**:

```
[edit]
user@host# set interfaces xe-2/0/0 per-unit-scheduler
user@host# set interfaces xe-2/0/0 flexible-vlan-tagging
user@host# set interfaces xe-2/0/0 unit 0 vlan-id 1
user@host# set interfaces xe-2/0/0 unit 0 family inet address 1.1.1.1/24

user@host# set interfaces xe-5/0/0 per-unit-scheduler
user@host# set interfaces xe-5/0/0 flexible-vlan-tagging
user@host# set interfaces xe-5/0/0 unit 0 vlan-id 1
user@host# set interfaces xe-5/0/0 unit 0 family inet address 2.1.1.1/24
```

2. Map each of four queues to a forwarding class.

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

3. Assign a packet-scheduling priority value to each forwarding class.

```
[edit]
user@host# set class-of-service schedulers be_sch priority low
user@host# set class-of-service schedulers ef_sch priority low
user@host# set class-of-service schedulers af_sch priority strict-high
user@host# set class-of-service schedulers ef_sch priority low
```

4. Customize the default IEEE 802.1p classifier (BA classifier based on Layer 2 header) by defining different values for iEEE 802.1p code points.

```
[edit]
user@host# set class-of-service classifiers ieee_jitter forwarding-class be
  loss-priority low code-points 000
```

```
user@host# set class-of-service classifiers ieee_jitter forwarding-class ef loss-priority
  low code-points 001
user@host# set class-of-service classifiers ieee_jitter forwarding-class af loss-priority
  low code-points 010
user@host# set class-of-service classifiers ieee_jitter forwarding-class nc
  loss-priority low code-points 011
```

5. Apply the BA classifier to the input of the logical units on **xe-2/0/0**.

```
[edit]
user@host# set interfaces xe-2/0/0 unit * classifiers ieee-802.1 ieee_jitter
```

6. Configure the scheduler map **smap_jitter** to map the forwarding classes to the schedulers.

```
[edit]
user@host# set class-of-service scheduler-maps smap_jitter forwarding-class be
  scheduler be_sch
user@host# set class-of-service scheduler-maps smap_jitter forwarding-class ef
  scheduler ef_sch
user@host# set class-of-service scheduler-maps smap_jitter forwarding-class af
  scheduler af_sch
user@host# set class-of-service scheduler-maps smap_jitter forwarding-class nc
  scheduler nc_sch
```

7. Configure the traffic control profile **tcp** to combine the scheduler map **smap_jitter** (that maps the forwarding classes to the schedulers for port-based scheduling) with a shaping rate (for hierarchical scheduling).

```
[edit]
user@host# set class-of-service traffic-control-profiles tcp scheduler-map
  smap_jitter
user@host# set class-of-service traffic-control-profiles tcp shaping-rate 6g
```

8. Apply the traffic control profile to the router output at **xe-5/0/0**.

```
[edit]
user@host# set class-of-service-interfaces xe-5/0/0 unit *
  output-traffic-control-profile tcp
```

9. If you are done configuring the device, commit the configuration.

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

Results Confirm your configuration by entering **show interfaces** and **show cloass-of-service** commands from configuration mode. If the output does not display the intended configuration, repeat the instructions in this example to correct the configuration.

```
[edit]
user@host# show interfaces
xe-2/0/0 {
  per-unit-scheduler;
  flexible-vlan-tagging;
  unit 0 {
    vlan-id 1;
    family inet {
      address 1.1.1.1/24;
```



```
        }
    }
}
xe-5/0/0 {
    per-unit-scheduler;
    flexible-vlan-tagging;
    unit 0 {
        vlan-id 1;
        family inet {
            address 2.1.1.1/24;
        }
    }
}
```

```
[edit]
user@host# show class-of-service
classifiers {
    ieee-802.1 ieee_jitter {
        forwarding-class be {
            loss-priority low code-points 000;
        }
        forwarding-class ef {
            loss-priority low code-points 001;
        }
        forwarding-class af {
            loss-priority low code-points 010;
        }
        forwarding-class nc {
            loss-priority low code-points 011;
        }
    }
}
forwarding-classes {
    queue 0 be;
    queue 1 ef;
    queue 2 af;
    queue 3 nc;
}
traffic-control-profiles {
    tcp {
        scheduler-map smap_jitter;
        shaping-rate 6g;
    }
}
interfaces {
    xe-2/0/0 {
        unit * {
            classifiers {
                ieee-802.1 ieee_jitter;
            }
        }
    }
    xe-5/0/0 {
        unit * {
            output-traffic-control-profile tcp;
        }
```

```
    }
  }
  scheduler-maps {
    smap_jitter {
      forwarding-class be scheduler be_sch;
      forwarding-class ef scheduler ef_sch;
      forwarding-class af scheduler af_sch;
      forwarding-class nc scheduler nc_sch;
    }
  }
  schedulers {
    be_sch {
      priority low;
    }
    ef_sch {
      priority low;
    }
    af_sch {
      priority strict-high;
    }
    nc_sch {
      priority low;
    }
  }
```

Verification

Confirm that the configuration is working properly

Measuring End-to-End Jitter to Establish the Baseline

Purpose Establish a baseline measurement by noting the amount of jitter that occurs when the hierarchical queuing line card hosting the egress port is configured with the default maximum number of queues.

Action To measure jitter:

1. Pass traffic through the VLAN.
2. Measure the variation in packet delay for selected packets in the data flow.

Configuring Jitter Reduction

Purpose Reduce jitter in the VLAN port output queues.

Action

1. Configure a reduced maximum number of queues for egress ports on the hierarchical queuing MPC in slot 5, thereby reducing the jitter in the port queues.

```
[edit]
user@host# set chassis fpc 5 max-queue 64k
```


2. If you are done configuring the device, commit the configuration.

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

Measuring End-to-End Jitter to Verify Jitter Reduction

Purpose Measure the amount of jitter that occurs when the hierarchical queuing line card hosting the egress port is configured with a reduced maximum number of queues.

Action To measure jitter:

1. Pass traffic through the VLAN.
2. Measure the variation in packet delay for selected packets in the data flow.

Related Documentation

- Jitter Reduction in Hierarchical CoS Queues on page 33
- **max-queues on page 158**

CHAPTER 12

Configuration Statements

buffer-size (Schedulers)

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.
Statement introduced in Junos OS Release 12.1X48 for PTX Series Packet Transport Routers.
Statement introduced in Junos OS Release 12.2 for ACX Series Routers.

Description Specify buffer size.

NOTE: **On PTX Series Packet Transport Routers, buffer-size cannot be configured on rate-limited queues.**

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, respectively.

Options **percent *percentage***—Buffer size as a percentage of the total buffer.

Range: 0 through 100

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. See Table 15 on page 40.

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 40
- *Example: Configuring CoS for a PBB Network on MX Series Routers*

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 FRF.15 and 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 20 on page 64.

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

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 60
- Providing a Guaranteed Minimum Rate on page 69

- *Configuring Traffic Control Profiles for Shared Scheduling and Shaping*

- **output-traffic-control-profile on page 160**

drop-profile-map (Schedulers)

Syntax drop-profile-map loss-priority (any | low | medium-low | medium-high | high) protocol(any | non-tcp | tcp) drop-profile (Schedulers) *profile-name*;

Hierarchy Level [edit class-of-service schedulers *scheduler-name*]

Release Information Statement introduced before Junos OS Release 7.4.
Statement introduced in Junos OS Release 12.1X48 for PTX Series Packet Transport Routers.
Statement introduced in Junos OS Release 12.2 for ACX Series Routers.

Description Define the loss-priority value for a 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 Overview on page 4
- Configuring Drop Profile Maps for Schedulers on page 51

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*
- Configuring Hierarchical Schedulers for CoS on page 121
- Configuring Interface Sets on page 123

excess-priority

Syntax excess-priority [low | medium-low | medium-high | high | none];

Hierarchy Level [edit class-of-service schedulers *scheduler-name*]

Release Information Statement introduced in Junos OS Release 9.3.
Option **none** introduced in Junos OS Release 11.4.

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.

none—System does not demote the priority of guaranteed traffic when the bandwidth exceeds the shaping rate or the guaranteed rate.

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*
- Bandwidth Sharing on Nonqueuing Packet Forwarding Engines Overview on page 6
- *Managing Excess Bandwidth Distribution on Static Interfaces on MICs and MPCs*

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 MIC and MPC interfaces added in Junos OS Release 10.1.
Statement introduced in Junos OS Release 12.1X48R2 for PTX Series Packet Transport Routers.

Description For an Enhanced IQ PIC interfaces, Multiservices PIC interfaces, or MX Series router interfaces on MPCs or MICs, and T4000 router interfaces on Type 5 FPCs and EX Series switches, determine the percentage or proportion of excess bandwidth traffic to share.

NOTE: **The proportion option provides a greater range of values over the percent option and hence influences the priorities assigned to the queues.**

Options ***percentage***—Percentage of the excess bandwidth to share.

Range: 0 through 100 percent

Default: Excess bandwidth is shared in proportion to the configured transmit rate of each queue.

value—(M Series, MX Series, T Series routers and EX Series switches only) 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 Scheduler Transmission Rate on page 51
- *Configuring Excess Bandwidth Sharing on IQE PICs*
- *Allocating Excess Bandwidth Among Frame Relay DLCIs on Multiservices PICs*
- *Managing Excess Bandwidth Distribution on Static Interfaces on MICs and MPCs*

fabric (Class-of-Service)

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.
Statement introduced before Junos OS 11.4 for EX Series switches.

Description Define CoS parameters of the switch fabric. For M320 and T Series routers only, associate a scheduler with a fabric priority.

On EX Series switches, this statement is supported only on EX8200 standalone switches and EX8200 Virtual Chassis.

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

- See Associating Schedulers with Fabric Priorities on page 72.

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.
Statement introduced in Junos OS Release 12.2 for ACX Series routers.

Description Associate a forwarding class configuration or default mapping with a specific interface.

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

- *Applying Forwarding Classes to Interfaces*

guaranteed-rate

Syntax guaranteed-rate (percent *percentage* | *rate*) <burst-size *bytes*>;

Hierarchy Level [edit class-of-service traffic-control-profiles *profile-name*]

Release Information Statement introduced in Junos OS Release 7.6.
Option **burst-size** introduced for Enhanced Queuing (EQ) DPC interfaces in Junos OS Release 9.4.
Option **burst-size** introduced for MIC and MPC interfaces in Junos OS Release 11.4.
Option **burst-size** introduced for IQ2 and IQ2E interfaces in Junos OS Release 12.3

Description For Gigabit Ethernet IQ, Channelized IQ PICs, AS PIC FRF.16 LSQ interfaces, and EQ DPCs only, configure a guaranteed minimum rate. You can also configure an optional burst size for a logical interface on EQ DPCs and on IQ2 and IQ2E PICs. This can help to ensure that higher priority services do not starve lower priority services.

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

burst-size *bytes*—(Optional) Maximum burst size, in bytes.

Required Privilege Level interface—To view this statement in the configuration.
interface-control—To add this statement to the configuration.

Related Documentation

- Providing a Guaranteed Minimum Rate on page 69
- *Configuring Traffic Control Profiles for Shared Scheduling and Shaping*
- **output-traffic-control-profile on page 160**

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.

NOTE: To enable hierarchical scheduling on MX80 routers, configure the **hierarchical-scheduler** statement at each member physical interface level of a particular aggregated Ethernet interface as well as at that aggregated Ethernet interface level. On other routing platforms, it is enough if you include this statement at the aggregated Ethernet interface level.

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 121
- Understanding Two-Level and Three-level Hierarchical CoS for Subscriber Interfaces on MX Series Routers on page 17
- **hierarchical-scheduler (Subscriber Interfaces on MX Series Routers) on page 150**

hierarchical-scheduler (Subscriber Interfaces on MX Series Routers)

Syntax

```
hierarchical-scheduler {
  implicit-hierarchy;
  maximum–hierarchy–levels number;
}
```

Hierarchy Level [edit interfaces *interface-name*]

Release Information Statement introduced in Junos OS Release 10.1.
implicit-hierarchy option introduced in Junos OS Release 13.1.

Description On MX Series routers with MPC/MICs, configure the parameters for hierarchical scheduling on the interface.

NOTE: To enable hierarchical scheduling on MX Series routers, configure the **hierarchical-scheduler** statement at each member physical interface level of a particular aggregated Ethernet interface as well as at that aggregated Ethernet interface level. On other routing platforms, it is enough if you include this statement at the aggregated Ethernet interface level.

Options **implicit-hierarchy**—Configure three-level hierarchical scheduling. When you include the **implicit-hierarchy** option, a hierarchical relationship is formed between the CoS scheduler nodes at level 1, level 2, and level 3. The **implicit-hierarchy** option is supported only on MPC/MIC subscriber interfaces and interface sets running over aggregated Ethernet on MX Series routers.

maximum-hierarchy-levels ***number***—Configure two-level hierarchical scheduling. Specify the maximum number of hierarchical scheduling levels allowed for node scaling. The only supported value is 2. The **maximum-hierarchy-levels** option is supported on MPC/MIC or EQ DPC subscriber interfaces and interface sets running over aggregated Ethernet on MX Series routers.

- If you include the **maximum-hierarchy-levels** option, interface sets are allowed only at level 3; they are not allowed at level 2. In this case, if you configure a level 2 interface set, you generate Packet Forwarding Engine errors.
- If you do not include the **maximum-hierarchy-levels** option, interface sets can be at either level 2 or level 3, depending on whether the member logical interfaces within the interface set have a traffic control profile. If any member logical interface has a traffic control profile, then the interface set is a level 2 CoS scheduler node. If no member logical interface has a traffic control profile, the interface set is at level 3.

Required Privilege Level view-level—To view this statement in the configuration.
control-level—To add this statement to the configuration.

Related Documentation

- Understanding Two-Level and Three-level Hierarchical CoS for Subscriber Interfaces on MX Series Routers on page 17
- *Configuring Hierarchical CoS for a Subscriber Interface of Aggregated Ethernet Links*
- *Junos OS Class of Service Library for Routing Devices*
- Configuring Hierarchical Schedulers for CoS on page 121
- *Hierarchical CoS on MPLS Pseudowire Subscriber Interfaces Overview*

input-traffic-control-profile

Syntax

input-traffic-control-profile *profile-name* shared-instance *instance-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

For Gigabit Ethernet IQ2 and IQ2E PIC, Enhanced Queuing DPC , MIC, and MPC interfaces, apply an input traffic scheduling and shaping profile to the logical interface.

NOTE: **The shared-instance statement applies only to Gigabit Ethernet IQ2 and IQ2E PICs.**

Options

profile-name—Name of the traffic-control profile to be applied to this interface.

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 Traffic Control Profiles for Shared Scheduling and Shaping*
- *Configuring Ingress Hierarchical CoS on Enhanced Queuing DPCs*
- *input-shaping-rate (Logical Interface)*
- **traffic-control-profiles on page 176**

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 DPC, MICs, or MPC interfaces on MX Series routers, or for IQ2E PICs interfaces on M Series and T Series router, apply an input traffic scheduling and shaping profile for the remaining traffic to the logical interface or interface set.

Options ***profile-name***—Name of the traffic-control profile for the 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*
- input-traffic-control-profile on page 151

interfaces

Syntax

```
interfaces {
  interface-name {
    classifiers{
      dscp(classifier-name | default) {
    }
      ieee-802.1 (classifier-name | default)  vlan-tag (inner | outer | classifier-name);
      inet-precedence (rewrite-name | default);
    }
    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);
    rewrite-rules {
      dscp (rewrite-name | default);
      ieee-802.1 (rewrite-name | default) vlan-tag (outer);
      inet-precedence (rewrite-name | default);
      }
    }
    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 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);
        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-bandwidth-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.

NOTE: The dscp-ipv6 and ieee-802.1ad classifier types are not supported on ACX Series routers. For further information about support on ACX Series routers, see *Understanding CoS CLI Configuration Statements on ACX Series Universal Access Routers*.

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*
- *Configuring Rewrite Rules*
- *Understanding CoS CLI Configuration Statements on ACX Series Universal Access Routers*

interface-set (Ethernet Interfaces)

Syntax

```
interface-set interface-set-name {
  interface ethernet-interface-name {
    (unit unit-number | vlan-tags-outer vlan-tag);
  }
}
```

Hierarchy Level [edit interfaces]

Release Information Statement introduced in Junos OS Release 8.5.

Description The set of interfaces used to configure hierarchical CoS schedulers on Ethernet interfaces on the MX Series router and IQ2E PIC on M Series and T Series routers.

The remaining statements are described 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 E-LINE and E-LAN Services for a PBB Network on MX Series Routers*
- *Junos OS Class of Service Library for Routing Devices*

interface-set (Hierarchical Schedulers)

Syntax

```
interface-set interface-set-name {
  excess-bandwidth-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 Enhanced Queuing DPC, MIC, or MPC interfaces on MX Series routers, or for IQ2E PIC interfaces on M Series routers, configure hierarchical schedulers for an interface set.

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 Interface Sets on page 123
- Configuring Hierarchical Schedulers for CoS on page 121

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 128

loss-priority (Scheduler Drop Profiles)

Syntax loss-priority (any | high | low | medium-high | medium-low);

Hierarchy Level [edit class-of-service schedulers *scheduler-name* drop-profile-map]

Release Information Statement introduced before Junos OS Release 7.4.
Statement introduced in Junos OS Release 12.1X48 for PTX Series Packet Transport Routers.
Statement introduced in Junos OS Release 12.2 for ACX Series Routers.

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.

NOTE: On ACX Series Routers, only the **any** option is supported when you configure the **non-tcp** option for protocol.

high—The drop profile applies to packets with high PLP.

low—The drop profile applies to packets with low PLP.

medium-high—The drop profile applies to packets with medium-high PLP.

medium-low—The drop profile applies to packets with medium-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 Overview on page 4
- Configuring Drop Profile Maps for Schedulers on page 51
- Configuring Schedulers for Priority Scheduling on page 55
- *Configuring Tricolor Marking*
- protocol (Schedulers) on page 166

max-queues

Syntax max-queues *queues-per-line-card*;

Hierarchy Level [edit chassis fpc *slot-number*]

Release Information Statement introduced in Junos OS Release 13.2.

Description Configure the maximum number of queues allowed per MPC1 Q, MPC2 Q, and MPC2 EQ line card in an MX240, MX480, MX960, MX2010, or MX2020 router or per 2-port or 4-port 10-Gigabit Ethernet fixed MIC with XFP in an MX5, MX10, MX40, or MX80 modular chassis router.

Reducing the number of queues allowed in a hierarchical scheduling environment in turn reduces the degree of jitter in the queues.

Options ***queues-per-line-card***—Maximum number of queues allowed for the line card. Only the following keywords are valid: **8k**, **16k**, **32k**, **64k**, **128k**, or **256k**.

- Built-in 10-Gigabit Ethernet MICs and MPC1 Q line cards support up to 128 K queues. You can use this statement to configure the single Packet Forwarding Engine to support a lower maximum number of queues.
- MPC2 Q and MPC2 EQ line cards support up to 256 K queues. You can use this statement to configure the two Packet Forwarding Engines to support a lower maximum number of queues.

If you configure a keyword for a value that exceeds the number of queues supported by the line card hardware, the system uses the maximum number of queues supported by the line card.

Required Privilege Level interface—To view this statement in the configuration.
interface-control—To add this statement to the configuration.

Related Documentation

- Jitter Reduction in Hierarchical CoS Queues on page 33
- *MPC1 Q*
- *MPC2 Q*
- *MPC2 EQ*
- *MIC/MPC Compatibility*
- *10-Gigabit Ethernet MICs with XFP*

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.

Required Privilege Level view-level—To view this statement in the configuration.
control-level—To add this statement to the configuration.

Related Documentation

- Configuring Hierarchical Schedulers for CoS on page 121

output-traffic-control-profile

Syntax
output-traffic-control-profile *profile-name* shared-instance *instance-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 interfaces *interface-name* interface-set *interface-set-name*]

Release Information
Statement introduced in Junos OS Release 7.6.
interface-set option added for Enhanced Queuing DPCs on MX Series routers in Junos OS Release 8.5.
interface-set option added for MIC and MPC interfaces on MX Series routers in Junos OS Release 10.2.

Description
For Channelized IQ PIC interfaces, for Gigabit Ethernet IQ, Gigabit Ethernet IQ2, and IQ2E PIC interfaces, for link services IQ (LSQ) interfaces on AS PICs, and for Enhanced Queuing DPC, MIC, and MPC interfaces on MX Series routers and on EX Series switches, apply an output traffic scheduling and shaping profile to the 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 60
- *Configuring Traffic Control Profiles for Shared Scheduling and Shaping*
- *Example: Configuring CoS for a PBB Network on MX Series Routers*
- Configuring Hierarchical Schedulers for CoS on page 121 (Enhanced Queuing DPC, MIC, and MPC interfaces on MX Series routers)
- Configuring Interface Sets on page 123 (Enhanced Queuing DPC, MIC, and MPC interfaces on MX Series routers)
- **output-traffic-control-profile-remaining on page 161**
- **traffic-control-profiles on page 176**

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 DPC, MICs, and MPC interfaces on MX Series routers, and for IQ2E PIC interfaces 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. The remaining traffic is transmitted by the default interface or interface set.

You can map the TCP to the interface or interface set by using the **output-traffic-control-profile-remaining** statement to explicitly configure the queues of the default interface or interface set scheduler that transmits the remaining traffic.

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 121
- *Configuring Remaining Common Queues on MIC and MPC Interfaces*
- output-traffic-control-profile on page 160

overhead-accounting

Syntax

```
overhead-accounting {
  bytes bytes;
  cell-mode cell-mode-bytes cell-mode-bytes;
  frame-mode frame-mode-bytes frame-mode-bytes;
}
```

Hierarchy Level [edit class-of-service traffic-control-profiles *profile-name*] and [edit dynamic-profiles *profile-name* 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 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 Static Shaping Parameters to Account for Overhead in Downstream Traffic Rates*
- *Bandwidth Management for Downstream Traffic in Edge Networks Overview*
- *CoS Adjustment Control Profiles Overview*
- *Configuring CoS Adjustment Control Profiles*
- *adjustment-control-profiles*
- *Configuring Dynamic Shaping Parameters to Account for Overhead in Downstream Traffic Rates*
- *Bandwidth Management for Downstream Traffic in Edge Networks Overview*
- *egress-shaping-overhead*
- *bytes*
- *cell-mode*
- *frame-mode*

per-unit-scheduler

Syntax per-unit-scheduler;

Hierarchy Level [edit interfaces *interface-name*]

Release Information Statement introduced before Junos OS Release 7.4.
Statement introduced in Junos OS Release 13.2 on 16x10GE MPC and MPC3E line cards.
Statement introduced in Junos OS Release 13.2 on PTX Series Packet Transport Routers.

Description For Channelized OC3 IQ, Channelized OC12 IQ, Channelized STM1 IQ, Channelized T3 IQ, Channelized E1 IQ, E3 IQ, link services IQ interfaces (lsq-), link services (ls-) on J Series routers, Gigabit Ethernet IQ, Gigabit Ethernet IQ2 and IQ2-E, and 10-, 40-, and 100-Gigabit Ethernet interfaces (including the 16x10GE MPC), enable the association of scheduler map names with logical interfaces.

NOTE: Per-unit scheduling is not supported on T1 interfaces configured on the Channelized OC12 IQ PIC.

NOTE: On Gigabit Ethernet IQ2 and IQ2-E PICs without the **per-unit-scheduler** statement, the entire PIC supports 4071 VLANs and the user can configure all the VLANs on the same port.

On Gigabit Ethernet IQ2 and IQ2-E PICs with the **per-unit-scheduler** statement, the entire PIC supports 1024 – 2 * number of ports (1024 minus two times the number of ports), because each port is allocated two default schedulers.

When including the **per-unit-scheduler** statement, you must also include the **vlan-tagging** statement or the **flexible-vlan-tagging** statement (to apply scheduling to VLANs) or the **encapsulation frame-relay** statement (to apply scheduling to DLCIs) at the **[edit interfaces *interface-name*]** hierarchy level.

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 and Shaping Rate to DLCIs and VLANs on page 103
- Applying Scheduling and Shaping to VLANs on page 111
- *Configuring Virtual LAN Queuing and Shaping on PTX Series Packet Transport Routers*
- Scaling of Per-VLAN Queuing on Non-Queuing MPCs on page 118
- *flexible-vlan-tagging*
- *vlan-tagging*

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.
Statement introduced before Junos OS 11.4 for EX Series switches.

Description Define Fabric traffic priority. For M320, MX Series, T Series routers and EX Series switches 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.

On EX Series switches, this statement is supported only on EX8200 standalone switches and EX8200 Virtual Chassis.

Options **high**—Scheduler has high priority.

low—Scheduler has low priority.

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

- See Associating Schedulers with Fabric Priorities on page 72.
- *Understanding Junos OS CoS Components for EX Series Switches*

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.
Statement introduced in Junos OS Release 12.1X48 for PTX Series Packet Transport Routers.
Statement introduced in Junos OS Release 12.2 for ACX Series Routers.

Description Specify the 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.

NOTE: The **strict-high** priority level is the only priority level supported on ACX Series Routers. However, multiple strict-high priority queues can be configured per interface on ACX Series Routers.

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 55

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.
Statement introduced in Junos OS Release 12.1X48 for PTX Series Packet Transport Routers.
Statement introduced in Junos OS Release 12.2 for ACX Series Routers.

Description Specify the protocol type for the specified scheduler.

Options **any**—Accept any protocol type.

non-tcp—(ACX Series Routers, M Series and T Series (except T4000) routers only) Accept any protocol type other than TCP/IP.

> NOTE: On ACX Series Routers, when you configure the **non-tcp** option, only the **any** option is supported for loss-priority.

tcp—(ACX Series Routers, M Series and T Series (except T4000) routers only) Accept TCP/IP protocol type.

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 39

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.
Statement introduced before Junos OS 11.4 for EX Series switches.

Description Define scheduler name. For M320, MX Series, T Series routers and for EX Series switches 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.

Required Privilege Level interface—To view this statement in the configuration.
interface-control—To add this statement to the configuration.

Related Documentation

- See Associating Schedulers with Fabric Priorities on page 72.
- *Understanding Junos OS CoS Components for EX Series Switches*

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.
Statement introduced in Junos OS Release 12.2 for ACX Series Routers.

Description Associate a scheduler with a scheduler map.

Options ***scheduler-name***—Name of the scheduler configuration block.

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 39
- *Example: Configuring CoS for a PBB Network on MX Series Routers*

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.
Statement introduced before Junos OS 11.4 for EX Series switches.

Description Mapping of fabric traffic to packet schedulers. For M320, MX Series, T Series routers, and for EX Series switches only, associate a scheduler with a fabric priority.

On EX Series switches, this statement is supported only on EX8200 standalone switches and EX8200 Virtual Chassis.

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

- See Associating Schedulers with Fabric Priorities on page 72.
- *Understanding Junos OS CoS Components for EX Series Switches*

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 FRF.15 and 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.

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 39
- Oversubscribing Interface Bandwidth on page 60
- **output-traffic-control-profile on page 160**

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 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.

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.

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 Packet Forwarding Component Queues on page 90
- **scheduler-map (Fabric Queues) on page 168**

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.

See “Configuring Schedulers” on page 39 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 (Class of Service)

Syntax

```
schedulers {
  scheduler-name {
    adjust-minimum rate;
    adjust-percent percentage;
    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-priority [ low | medium-low | medium-high | high | none];
    excess-rate (percent percentage | proportion value);
    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.
Statement introduced in Junos OS Release 12.1X48 for PTX Series routers.

Description Specify the 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

- Schedulers Overview on page 3
- Default Schedulers Overview on page 4
- Configuring Schedulers on page 39
- *Configuring a Scheduler*
- *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

Required Privilege Level interface—To view this statement in the configuration.
interface-control—To add this statement to the configuration.

Related Documentation

- Configuring the Number of Schedulers for Ethernet IQ2 PICs on page 83

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-name*] hierarchy level added in Junos OS Release 7.5.
Statement introduced in Junos OS Release 13.2 on PTX Series Packet Transport Routers.

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 and T4000 routers with Type 5 FPCs only, configure traffic shaping based on the rate-limited bandwidth of the total interface bandwidth.

Logical and physical interface traffic shaping rates are mutually exclusive. This means 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.

NOTE: For MX Series routers and for EX Series switches, 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.

For T4000 routers with Type 5 FPCs, the shaping rate value for the physical interface must be a minimum of 292 Kbps. The maximum value of **shaping-rate** is limited by the maximum transmission rate of the 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 60.

For FRF.15 and 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-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-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 and EX series interfaces, the rate can be from 65,535 through 160,000,000,000 bps.

NOTE: For T4000 physical interfaces, the rate can be from 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

- Applying Scheduler Maps Overview on page 5
- *Configuring Virtual LAN Queuing and Shaping on PTX Series Packet Transport Routers*

shaping-rate (Oversubscribing an Interface)

Syntax shaping-rate (percent *percentage* | *rate*) <burst-size *bytes*>;

Hierarchy Level [edit class-of-service traffic-control-profiles *profile-name*]

Release Information Statement introduced in Junos OS Release 7.6.
Option **burst-size** introduced for Enhanced Queuing (EQ) DPC interfaces on MX Series routers in Junos OS Release 9.4.
Option **burst-size** option introduced for MIC and MPC interfaces on MX Series routers in Junos OS Release 11.4.
Option **burst-size** introduced for IQ2 and IQ2E interfaces in Junos OS Release 12.3.

Description For Gigabit Ethernet IQ, Channelized IQ PIC, FRF.15 and FRF.16 LSQ interfaces, and for EQ DPC, MIC, and MPC interfaces on MX Series routers, configure a shaping rate for a logical interface. You can also configure an optional burst size for a logical interface on EQ DPC interfaces and on IQ2 and IQ2E PIC interfaces. This can help to ensure that higher-priority services do not starve lower-priority services.

For physical interfaces on T4000 router interfaces on Type 5 FPCs, configure traffic shaping rate.

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 20 on page 64.

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, and T4000 routers with Type 5 FPCs, 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: IQ and IQ2 interfaces—1000 through 160,000,000,000 bps

Range: T4000 routers with Type 5 FPCs—the shaping rate value for the physical interface must be a minimum of 292 Kbps. The maximum value of **shaping-rate** is limited by the maximum transmission rate of the interface.

burst-size ***bytes***—(Optional) 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

- *Configuring Traffic Control Profiles for Shared Scheduling and Shaping*
- Oversubscribing Interface Bandwidth on page 60
- **output-traffic-control-profile on page 160**

traffic-control-profiles

Syntax

```
traffic-control-profiles profile-name {
  adjust-minimum rate;
  atm-service (cbr | rtvbr | nrtvbr);
  delay-buffer-rate (percent percentage | rate);
  excess-rate (percent percentage  | proportion value );
  excess-rate-high (percent percentage | proportion value);
  excess-rate-low (percent percentage | proportion value);
  guaranteed-rate (percent percentage | rate) <burst-size bytes>;
  max-burst-size cells;
  overhead-accounting (frame-mode | cell-mode | frame-mode-bytes | cell-mode-bytes)
    <bytes (byte-value)>;
  peak-rate rate;
  scheduler-map map-name;
  shaping-rate (percent percentage | rate) <burst-size bytes>;
  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 ];
  sustained-rate rate;
}
```

Hierarchy Level [edit class-of-service]

Release Information Statement introduced in Junos OS Release 7.6.

Description For Gigabit Ethernet IQ, Channelized IQ PICs, FRF.15 and FRF.16 LSQ interfaces, and Enhanced Queuing (EQ) DPCs only, configure traffic shaping and scheduling profiles. For Enhanced EQ PICs and EQ DPCs 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 60
- *Example: Configuring CoS for a PBB Network on MX Series Routers*
- **output-traffic-control-profile on page 160**

transmit-rate (Schedulers)

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.
Statement introduced in Junos OS Release 12.1X48 for PTX Series Packet Transport Routers.
Statement introduced in Junos OS Release 12.2 for ACX Series Routers.

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, respectively.

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. For PTX Series Packet Transport Routers, this option is allowed only on the non-strict-high (high, medium-high, medium-low, or low) queues.

percent *percentage*—Percentage of transmission capacity. A percentage of zero drops all packets in the queue.

Range: 0 through 100 percent for M, MX and T Series routers and EX Series switches; 1 through 100 percent for PTX Series Packet Transport Routers; 0 through 200 percent for the SONET/SDH OC48/STM16 IQE PIC

NOTE:

- On M Series Multiservice Edge Routers, for interfaces configured on 4-port E1 and 4-port T1 PICs only, you can configure a *percentage* value only from 11 through 100. These two PICs do not support transmission rates less than 11 percent.
- The configuration of the **transmit-rate percent 0 exact** statement at the **[edit class-of-service schedules *scheduler-name*]** hierarchy is ineffective on T4000 routers with Type 5 FPC.
- On MIC and MPC interfaces on MX Series routers, when the transmit rate is configured as a percentage and **exact** or **rate-limit** is enabled on a queue, the shaping rate of the parent node is used to compute the transmit rate. If **exact** or **rate-limit** is not configured, the guaranteed rate of the parent node is used to compute the transmit rate.

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.

NOTE: **For PTX Series Packet Transport Routers, this option is allowed only on the strict-high queue. We recommend that you configure rate limit on strict-high queues because the other queues may not meet their guaranteed bandwidths.**

NOTE: **The configuration of the rate-limit statement is supported on T4000 routers only with a Type 5 FPC.**

remainder—Use the 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 39
- Configuring Scheduler Transmission Rate on page 51
- *Example: Configuring CoS for a PBB Network on MX Series Routers*

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 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);
    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*
- *Configuring Rewrite Rules*

PART 3

Index

Index

Symbols

B

C