

Schedulers on EX9200 Switches



Published: 2015-05-15

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

Juniper Networks, Junos, Steel-Belted Radius, NetScreen, and ScreenOS are registered trademarks of Juniper Networks, Inc. in the United States and other countries. The Juniper Networks Logo, the Junos logo, and JunosE are trademarks of Juniper Networks, Inc. All other trademarks, service marks, registered trademarks, or registered service marks are the property of their respective owners.

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

Schedulers on EX9200 Switches
Copyright © 2015, Juniper Networks, Inc.
All rights reserved.

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

YEAR 2000 NOTICE

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

END USER LICENSE AGREEMENT

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

Table of Contents

	About the Documentation	ix
	Documentation and Release Notes	ix
	Supported Platforms	ix
	Using the Examples in This Manual	ix
	Merging a Full Example	x
	Merging a Snippet	x
	Documentation Conventions	xi
	Documentation Feedback	xiii
	Requesting Technical Support	xiii
	Self-Help Online Tools and Resources	xiii
	Opening a Case with JTAC	xiv
Part 1	Overview	
Chapter 1	Schedulers	3
	Schedulers Overview	3
	Default Schedulers Overview	4
	Priority Scheduling Overview	5
	Applying Scheduler Maps Overview	6
	Applying a Shaping Rate to Physical Interfaces Overview	7
	Forwarding Classes and Fabric Priority Queues	8
	Default Fabric Priority Queuing	8
	Overriding Default Fabric Priority Queuing	8
Part 2	Configuration	
Chapter 2	Configuration Tasks for Schedulers	11
	Configuring Schedulers	11
	Managing Congestion on the Egress Interface by Configuring the Scheduler	
	Buffer Size	12
	Configuring Large Delay Buffers for Slower Interfaces	14
	Maximum Delay Buffer for NxDSO Interfaces	17
	Example: Configuring Large Delay Buffers for Slower Interfaces	19
	Enabling and Disabling the Memory Allocation Dynamic per Queue	21
	Determining Packet Drop Behavior by Configuring Drop Profile Maps for	
	Schedulers	23
	Configuring Scheduler Transmission Rate	24
	Example: Configuring Scheduler Transmission Rate	26
	Allocation of Leftover Bandwidth	26

	Configuring Schedulers for Priority Scheduling	27
	Example: Configuring Priority Scheduling	28
	Configuring Strict-High Priority on M Series and T Series Routers	28
	Configuring Per-Unit Schedulers for Channelized Interfaces	29
	Configuring the Shaping Rate for Physical Interfaces	32
	Oversubscribing Interface Bandwidth	33
	Verifying Configuration of Bandwidth Oversubscription	39
	Examples: Oversubscribing Interface Bandwidth	39
	Providing a Guaranteed Minimum Rate	42
	Verifying Configuration of Guaranteed Minimum Rate	44
	Example: Providing a Guaranteed Minimum Rate	45
	Associating Schedulers with Fabric Priorities	45
	Example: Associating a Scheduler with a Fabric Priority	46
Chapter 3	Configuration Tasks for Scheduler Maps	49
	Configuring Scheduler Maps	49
	Applying Scheduler Maps to Physical Interfaces	49
	Applying Scheduler Maps and Shaping Rate to DLCIs and VLANs	50
	Applying Scheduler Maps to Packet Forwarding Component Queues	57
	Applying Custom Schedulers to Packet Forwarding Component Queues	58
	Examples: Scheduling Packet Forwarding Component Queues	59
Chapter 4	Configuration Statements for Schedulers	63
	[edit class-of-service] Hierarchy Level	64
	buffer-size (Schedulers)	68
	delay-buffer-rate	69
	drop-profile-map (Schedulers)	70
	excess-priority	71
	excess-rate	72
	fabric (Class-of-Service)	73
	forwarding-class (Interfaces)	73
	guaranteed-rate	74
	interfaces (CoS)	75
	loss-priority (Scheduler Drop Profiles)	77
	output-traffic-control-profile	78
	priority (Fabric Queues, Schedulers)	79
	priority (Schedulers)	80
	protocol (Schedulers)	81
	scheduler (Fabric Queues)	82
	scheduler (Scheduler Map)	82
	scheduler-map (Fabric Queues)	83
	scheduler-map (Interfaces and Traffic-Control Profiles)	83
	scheduler-map-chassis	84
	scheduler-maps (For Most Interface Types)	85
	schedulers (CoS)	86
	shaping-rate (Applying to an Interface)	87
	shaping-rate (Oversubscribing an Interface)	89
	traffic-control-profiles	90
	transmit-rate (Schedulers)	91

unit	93
[edit interfaces] Hierarchy Level	93
schedulers (Interfaces)	104

List of Tables

	About the Documentation	ix
	Table 1: Notice Icons	xi
	Table 2: Text and Syntax Conventions	xi
Part 2	Configuration	
Chapter 2	Configuration Tasks for Schedulers	11
	Table 3: Buffer Size Temporal Value Ranges by Routing Device Type	13
	Table 4: Recommended Delay Buffer Sizes	14
	Table 5: Maximum Delay Buffer with q-pic-large-buffer Enabled by Interface . . .	15
	Table 6: Delay-Buffer Calculations	16
	Table 7: NxDSO Transmission Rates and Delay Buffers	17
	Table 8: Bandwidth and Delay Buffer Allocations by Configuration Scenario . . .	37
	Table 9: Bandwidth and Delay Buffer Allocations by Configuration Scenario . . .	44
Chapter 3	Configuration Tasks for Scheduler Maps	49
	Table 10: Fine-Grained Queuing and Scheduling Support by Interface or PIC Type	52
	Table 11: Fine-Grained Queuing and Scheduling Support by MIC or MPC Type . . .	54

About the Documentation

- Documentation and Release Notes on page ix
- Supported Platforms on page ix
- Using the Examples in This Manual on page ix
- Documentation Conventions on page xi
- Documentation Feedback on page xiii
- Requesting Technical Support on page xiii

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:

- EX 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.xml;
    }
  }
}
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.xml; }
```

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 xi 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.
	Tip	Indicates helpful information.
	Best practice	Alerts you to a recommended use or implementation.

Table 2 on page xi 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

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

Convention	Description	Examples
Fixed-width text like this	Represents output that appears on the terminal screen.	<code>user@host> show chassis alarms</code> <code>No alarms currently active</code>
<i>Italic text like this</i>	<ul style="list-style-type: none">Introduces or emphasizes important new terms.Identifies guide names.Identifies RFC and Internet draft titles.	<ul style="list-style-type: none">A policy <i>term</i> is a named structure that defines match conditions and actions.<i>Junos OS CLI User Guide</i>RFC 1997, <i>BGP Communities Attribute</i>
<i>Italic text like this</i>	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 <i>domain-name</i>
Text like this	Represents names of configuration statements, commands, files, and directories; configuration hierarchy levels; or labels on routing platform components.	<ul style="list-style-type: none">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)	Encloses 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)	Encloses a variable for which you can substitute one or more values.	community name members [<i>community-ids</i>]
Indentation and braces ({ })	Identifies 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		
Bold text like this	Represents graphical user interface (GUI) items you click or select.	<ul style="list-style-type: none">In the Logical Interfaces box, select All Interfaces.To cancel the configuration, click Cancel.

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

Convention	Description	Examples
> (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 provide feedback by using either of the following methods:

- Online feedback rating system—On any page at the Juniper Networks Technical Documentation site at <http://www.juniper.net/techpubs/index.html>, simply click the stars to rate the content, and use the pop-up form to provide us with information about your experience. Alternately, you can use the online feedback form at <https://www.juniper.net/cgi-bin/docbugreport/>.
- E-mail—Send your comments to techpubs-comments@juniper.net. Include the document or topic name, URL or page number, and software 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:
<http://kb.juniper.net/InfoCenter/>
- 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

- [Schedulers on page 3](#)

CHAPTER 1

Schedulers

- [Schedulers Overview on page 3](#)
- [Default Schedulers Overview on page 4](#)
- [Priority Scheduling Overview on page 5](#)
- [Applying Scheduler Maps Overview on page 6](#)
- [Applying a Shaping Rate to Physical Interfaces Overview on page 7](#)
- [Forwarding Classes and Fabric Priority Queues on page 8](#)

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 26](#).

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;
  }
}
```

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 24](#). 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 round robin 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 round-robin fashion.
2. The software locates all medium-high priority queues that are currently in profile. These queues are serviced second in a round-robin fashion.
3. The software locates all medium-low priority queues that are currently in profile. These queues are serviced third in a round-robin fashion.
4. The software locates all low-priority queues that are currently in profile. These queues are serviced fourth in a 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.

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 NxDS0 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 49](#)
- [Applying Scheduler Maps and Shaping Rate to Physical Interfaces on IQ PICs](#)
- [Applying Scheduler Maps and Shaping Rate to DLCIs and VLANs on page 50](#)
- [Oversubscribing Interface Bandwidth on page 33](#)
- [Providing a Guaranteed Minimum Rate on page 42](#)
- [Applying Scheduler Maps to Packet Forwarding Component Queues](#)
- [Forwarding Classes and Fabric Priority Queues on page 8](#)
- [Associating Schedulers with Fabric Priorities on page 45](#)

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 32](#)

Forwarding Classes and Fabric Priority Queues

This topic covers the following information:

- [Default Fabric Priority Queuing on page 8](#)
- [Overriding Default Fabric Priority Queuing on page 8](#)

Default Fabric Priority Queuing

On Juniper Networks EX Series switches, M320 Multiservice Edge Routers, and Juniper Networks T Series Core Routers, the default behavior is for fabric priority queuing on egress interfaces to match the scheduling priority you assign. High-priority egress traffic is automatically assigned to high-priority fabric queues. Likewise, low-priority egress traffic is automatically assigned to low-priority fabric queues.

Overriding Default Fabric Priority Queuing

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

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

Related Documentation

- [Associating Schedulers with Fabric Priorities on page 45](#)

PART 2

Configuration

- [Configuration Tasks for Schedulers on page 11](#)
- [Configuration Tasks for Scheduler Maps on page 49](#)
- [Configuration Statements for Schedulers on page 63](#)

CHAPTER 2

Configuration Tasks for Schedulers

- [Configuring Schedulers on page 11](#)
- [Managing Congestion on the Egress Interface by Configuring the Scheduler Buffer Size on page 12](#)
- [Determining Packet Drop Behavior by Configuring Drop Profile Maps for Schedulers on page 23](#)
- [Configuring Scheduler Transmission Rate on page 24](#)
- [Configuring Schedulers for Priority Scheduling on page 27](#)
- [Configuring Per-Unit Schedulers for Channelized Interfaces on page 29](#)
- [Configuring the Shaping Rate for Physical Interfaces on page 32](#)
- [Oversubscribing Interface Bandwidth on page 33](#)
- [Providing a Guaranteed Minimum Rate on page 42](#)
- [Associating Schedulers with Fabric Priorities on page 45](#)

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:

- [Managing Congestion on the Egress Interface by Configuring the Scheduler Buffer Size on page 12](#)
- [Determining Packet Drop Behavior by Configuring Drop Profile Maps for Schedulers on page 23](#)

- [Configuring Scheduler Transmission Rate on page 24](#)
- [Configuring Schedulers for Priority Scheduling on page 27](#)

Managing Congestion on the Egress Interface by 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 3 on page 13](#).

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 3 on page 13](#).
- 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 3 on page 13](#). The maximums apply to the logical interface, not each queue.



NOTE: In general, the default temporal buffer value is inversely related to the speed, or shaping rate, of the interface. As the speed of the interface increases, the interface needs less and less buffer to hold data, as it is possible for the interface to send more and more data.

For information about configuring large buffer sizes on IQ PICs, see [“Configuring Large Delay Buffers for Slower Interfaces” on page 14](#).

Table 3: 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
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 14](#)
- [Enabling and Disabling the Memory Allocation Dynamic per Queue on page 21](#)

Configuring Large Delay Buffers for Slower Interfaces

By default, T1, E1, and NxDS0 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 4 on page 14](#) shows some recommended buffer sizes needed to absorb typical burst sizes from various upstream interface types.

Table 4: 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 NxDS0 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 5 on page 15](#).

Table 5: Maximum Delay Buffer with q-pic-large-buffer Enabled by Interface

Platform, PIC, or Interface Type	Maximum Buffer Size
With Large Buffer Sizes Not Enabled	
M320 and T Series router FPCs, Type 1 and Type 2	80,000 microseconds
M320 and T Series router FPCs, Type 3	50,000 microseconds
Other M Series router FPCs	200,000 microseconds
IQ PICs on all routers	100,000 microseconds
With Large Buffer Sizes Enabled	
Channelized T3 and channelized OC3 DLCIs—Maximum sizes vary by shaping rate:	
With shaping rate from 64,000 through 255,999 bps	4,000,000 microseconds
With shaping rate from 256,000 through 511,999 bps	2,000,000 microseconds
With shaping rate from 512,000 through 1,023,999 bps	1,000,000 microseconds
With shaping rate from 1,024,000 through 2,048,000 bps	500,000 microseconds
With shaping rate from 2,048,001 bps through 10 Mbps	400,000 microseconds
With shaping rate from 10,000,001 bps through 20 Mbps	300,000 microseconds
With shaping rate from 20,000,001 bps through 30 Mbps	200,000 microseconds
With shaping rate from 30,000,001 bps through 40 Mbps	150,000 microseconds
With shaping rate from 40,000,001 bps and above	100,000 microseconds
NxDSO IQ Interfaces—Maximum sizes vary by channel size:	
1xDSO through 3xDSO	4,000,000 microseconds
4xDSO through 7xDSO	2,000,000 microseconds
8xDSO through 15xDSO	1,000,000 microseconds
16xDSO through 32xDSO	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 50](#).

For NxDSO interfaces, the larger buffer sizes can be up to 4,000,000 microseconds, depending on the number of DSO channels in the NxDSO interface. For slower NxDSO interfaces with fewer channels, the delay buffer can be relatively larger than for faster NxDSO interfaces with more channels. This is shown in [Table 7 on page 17](#). To calculate specific buffer sizes for various NxDSO interfaces, see [“Maximum Delay Buffer for NxDSO Interfaces” on page 17](#).

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 6 on page 16](#).

Table 6: Delay-Buffer Calculations

Delay Buffer Configuration	Formula	Example
Percentage	available interface bandwidth * configured percentage buffer-size * maximum buffer = queue buffer	<p>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:</p> <pre> sched-expedited { transmit-rate percent 30; buffer-size percent 30; } </pre> <p>1.5 Mbps * 0.3 * 500,000 microseconds = 225,000 bits = 28,125 bytes</p>
Temporal	available interface bandwidth * configured percentage transmit-rate * configured temporal buffer-size = queue buffer	<p>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:</p> <pre> sched-best { transmit-rate percent 20; buffer-size temporal 500000; } </pre> <p>1.5 Mbps * 0.2 * 500,000 microseconds = 150,000 bits = 18,750 bytes</p>

Table 6: Delay-Buffer Calculations (*continued*)

Delay Buffer Configuration	Formula	Example
Percentage, with buffer size larger than transmit rate		<p>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.</p> <pre> sched-extra-buffer { transmit-rate percent 10; buffer-size percent 20; } </pre>
FRF.16 LSQ bundles	<p>For total bundle bandwidth < T1 bandwidth, the delay-buffer rate is 1 second.</p> <p>For total bundle bandwidth >= T1 bandwidth, the delay-buffer rate is 200 milliseconds (ms).</p>	

For more information, see the following sections:

- [Maximum Delay Buffer for NxDSO Interfaces on page 17](#)
- [Example: Configuring Large Delay Buffers for Slower Interfaces on page 19](#)

Maximum Delay Buffer for NxDSO Interfaces

Because NxDSO interfaces carry less bandwidth than a T1 or E1 interface, the buffer size on an NxDSO interface can be relatively larger, depending on the number of DSO channels combined. The maximum delay buffer size is calculated with the following formula:

$$\text{Interface Speed} * \text{Maximum Delay Buffer Time} = \text{Delay Buffer Size}$$

For example, a 1xDSO 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):

$$64 \text{ Kbps} * 4,000,000 \text{ microseconds} = 32 \text{ KB}$$

[Table 7 on page 17](#) shows the delay-buffer calculations for 1xDSO through 32xDSO interfaces.

Table 7: NxDSO Transmission Rates and Delay Buffers

Interface Speed	Delay Buffer Size
1xDSO Through 4xDSO: Maximum Delay Buffer Time Is 4,000,000 Microseconds	
1xDSO: 64 Kbps	32 KB
2xDSO: 128 Kbps	64 KB
3xDSO: 192 Kbps	96 KB

Table 7: NxDSO Transmission Rates and Delay Buffers (*continued*)

Interface Speed	Delay Buffer Size
4xDSO Through 7xDSO: Maximum Delay Buffer Time Is 2,000,000 Microseconds	
4xDSO: 256 Kbps	64 KB
5xDSO: 320 Kbps	80 KB
6xDSO: 384 Kbps	96 KB
7xDSO: 448 Kbps	112 KB
8xDSO Through 15xDSO: Maximum Delay Buffer Time Is 1,000,000 Microseconds	
8xDSO: 512 Kbps	64 KB
9xDSO: 576 Kbps	72 KB
10xDSO: 640 Kbps	80 KB
11xDSO: 704 Kbps	88 KB
12xDSO: 768 Kbps	96 KB
13xDSO: 832 Kbps	104 KB
14xDSO: 896 Kbps	112 KB
15xDSO: 960 Kbps	120 KB
16xDSO Through 32xDSO: Maximum Delay Buffer Time Is 500,000 Microseconds	
16xDSO: 1024 Kbps	64 KB
17xDSO: 1088 Kbps	68 KB
18xDSO: 1152 Kbps	72 KB
19xDSO: 1216 Kbps	76 KB
20xDSO: 1280 Kbps	80 KB
21xDSO: 1344 Kbps	84 KB
22xDSO: 1408 Kbps	88 KB
23xDSO: 1472 Kbps	92 KB
24xDSO: 1536 Kbps	96 KB
25xDSO: 1600 Kbps	100 KB

Table 7: NxDSO Transmission Rates and Delay Buffers (*continued*)

Interface Speed	Delay Buffer Size
26xDSO: 1664 Kbps	104 KB
27xDSO: 1728 Kbps	108 KB
28xDSO: 1792 Kbps	112 KB
29xDSO: 1856 Kbps	116 KB
30xDSO: 1920 Kbps	120 KB
31xDSO: 1984 Kbps	124 KB
32xDSO: 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 6 on page 16](#).

```
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:

$$\text{available interface bandwidth} * \text{configured percentage buffer-size} * \text{maximum buffer} \\ = \text{queue buffer}$$

$$1.5 \text{ Mbps} * 0.2 * 500,000 \text{ microseconds} = 150,000 \text{ bits} = 18,750 \text{ bytes}$$

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 7 on page 17](#).

```
class-of-service {
  interfaces {
    t1-0/0/0:1 {
      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;
```

The minimum buffer allocated to any queue is 18,432 bytes. If a queue is configured to have a buffer size less than 18K, the queue retains a buffer size of 18,432 bytes.

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 68](#)
 - [schedulers \(CoS\) on page 86](#)
 - [q-pic-large-buffer](#)
 - [schedulers \(CoS\) on page 86](#)

Determining Packet Drop Behavior by Configuring Drop Profile Maps for Schedulers

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

Drop-profile maps associate RED drop profiles with a scheduler. The map examines the current loss priority setting of the packet (**low**, **medium-low**, **medium-high**, or **high**) 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 *Defining Packet Drop Behavior by Configuring RED Drop Profiles*.

By default, the drop profile is mapped to packets with low PLP and any protocol type. To configure how packet loss priorities 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;

```

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

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. In other words, the map sets the drop profile for each packet with a specific PLP and protocol type exiting the interface. For more information about how CoS maps work, see *Mapping CoS Component Inputs to Outputs*.



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.

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

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

To use this drop-profile map, you must configure the settings for the **dp** drop profile at the **[edit class-of-service drop-profiles dp]** hierarchy level. For more information, see *Managing Congestion Using RED Drop Profiles and Packet Loss Priorities*.

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

- *Defining Packet Drop Behavior by 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.

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 6,400,000,000,000 bps. On all other platforms, the rate can be from 3200 through 6,400,000,000,000 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 11](#).

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

- [Example: Configuring Scheduler Transmission Rate on page 26](#)
- [Allocation of Leftover Bandwidth on page 26](#)

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.

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, 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 27](#)
- [Schedulers Overview on page 3](#)
- [Configuring a Scheduler](#)
- [excess-rate on page 72](#)
- [schedulers on page 86](#)

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*.

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.

The following sections discuss priority scheduling:

- [Example: Configuring Priority Scheduling on page 28](#)
- [Configuring Strict-High Priority on M Series and T Series Routers on page 28](#)

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*

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 49](#).



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 cel;
}
cstm1-3/0/0 {
  no-partition 1 interface-type cau4;
}
cel-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 {
        unit 0 {
            scheduler-map schedule-mlppp;
        }
    }
    ds-5/3/0:1 {
        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 6,400,000,000,000 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 7](#)

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 data-link connection identifiers (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 50.

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 6,400,000,000,000 bps.

For all MX Series router and EX Series switch interfaces, the shaping rate can be from 65,535 through 6,400,000,000,000 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 42](#).

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 6,400,000,000,000 bps.

The actual delay buffer is based on the calculations described in [“Configuring Large Delay Buffers for Slower Interfaces” on page 14](#) and [“Maximum Delay Buffer for NxDSO Interfaces” on page 17](#). For an example showing how the delay-buffer rates are applied, see [“Examples: Oversubscribing Interface Bandwidth” on page 39](#).

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:

$$(\text{remaining delay-buffer rate} * \text{shaping rate}) / (\text{sum of shaping rates})$$

where the remaining delay-buffer rate is equal to:

$$(\text{interface speed}) - (\text{sum of configured delay-buffer rates})$$

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 11](#) and [“Configuring Scheduler Maps” on page 49](#).

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 14](#).

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 either the `scheduler-map` or `shaping-rate` statement is included in the logical interface configuration.

Table 8 on page 37 shows how the bandwidth and delay buffer are allocated in various configurations.

Table 8: 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 <code>[edit class-of-service interfaces <i>interface-name</i> unit <i>logical-unit-number</i>]</code> hierarchy level.	<p>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 <code>q-pic-large-buffer</code> statement. For more information, see “Configuring Large Delay Buffers for Slower Interfaces” on page 14.</p> <p>Unshaped logical interfaces receive the remaining bandwidth and a delay buffer in proportion to the remaining bandwidth.</p>
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.	<p>Logical interface receives a delay buffer based on the scaled shaping rate:</p> $\text{scaled shaping rate} = (\text{shaping-rate} * [\text{physical interface bandwidth}]) / \text{SUM}(\text{shaping-rates of all logical interfaces on the physical interface})$ <p>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.</p>

Table 8: Bandwidth and Delay Buffer Allocations by Configuration Scenario
Scenario (continued)

Configuration Scenario	Delay Buffer Allocation
You oversubscribe the interface. You configure a shaping rate. You configure a delay-buffer rate.	<p>Logical interface receives a delay buffer based on the delay-buffer rate. For example, on IQ and IQ2 interfaces:</p> <p>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</p> <p>On LSQ DLCIs, if total bundle bandwidth < T1 bandwidth:</p> <p>delay-buffer-rate = 1 second</p> <p>On LSQ DLCIs, if total bundle bandwidth >= T1 bandwidth:</p> <p>delay-buffer-rate = 200 ms</p> <p>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 14.</p> <p>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.</p>
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.	<p>Logical interface receives a delay buffer based on the guaranteed rate.</p> <p>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).</p>



NOTE: In Junos OS Release 13.3, IP packets with DLCI 0 or 1023 are identified as part of control traffic and routed to the high-priority queue. This oversubscribes the high-priority queue, which is reserved for frame relay control traffic. Oversubscribing the high-priority queue causes the frame relay Local Management Interface (LMI) packets to be dropped.

Verifying Configuration of Bandwidth Oversubscription

To verify your configuration, you can issue the 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 NxDSO Interfaces” on page 17](#).

```
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 6,400,000,000,000 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 6,400,000,000,000 bps.

The actual delay buffer is based on the calculations described in [“Configuring Large Delay Buffers for Slower Interfaces” on page 14](#) and [“Maximum Delay Buffer for NxDSO Interfaces” on page 17](#). For an example showing how the delay-buffer rates are applied, see [“Example: Providing a Guaranteed Minimum Rate” on page 45](#).

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 11](#) and [“Configuring Scheduler Maps” on page 49](#).

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 14](#).

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 9 on page 44](#) shows how the bandwidth and delay buffer are allocated in various configurations.

Table 9: 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 14 .
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 14 .

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:

$$\text{delay-buffer-rate} < [8 \times 64 \text{ Kbps}]: 2 \text{ seconds of delay-buffer-rate}$$

For more information about this calculation, see [“Maximum Delay Buffer for NxDSO Interfaces” on page 17](#).

```
chassis {
  fpc 3 {
    pic 0 {
      q-pic-large-buffer;
    }
  }
}
interfaces {
  tl-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 {
  tl-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 on page 8](#)

Configuration Tasks for Scheduler Maps

- [Configuring Scheduler Maps on page 49](#)
- [Applying Scheduler Maps to Physical Interfaces on page 49](#)
- [Applying Scheduler Maps and Shaping Rate to DLCIs and VLANs on page 50](#)
- [Applying Scheduler Maps to Packet Forwarding Component Queues on page 57](#)

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 49](#), 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 50](#).



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 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

You can configure logical interface scheduling on the following MICs and MPCs as well as any MPC that contains a queuing chip:

- 16x10GE MPC
- MPC3E:
 - 10x10GE MIC with SFP+
 - 2x40GE MIC with QSFP+
 - 1x100GE MIC with CXP
- MPC4E:
 - 32x10GE with SFPP
 - 2x100GE + 8x10GE with SFPP
- MPC6E:
 - 24x10GE MIC with SFPP
 - 24x10GE MIC with SFPP OTN
 - 2x100GE MIC with CFP2 OTN
 - 4x100GE MIC with CXP

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 33](#).

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 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 NxDS0 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 10 on page 52 shows the interfaces/PICs that support fine-grained queuing and scheduling.

Table 10: Fine-Grained Queuing and Scheduling Support by Interface or 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 10: Fine-Grained Queuing and Scheduling Support by Interface or 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 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 10: Fine-Grained Queuing and Scheduling Support by Interface or 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 11 on page 54 shows the MICs and MPCs that support fine-grained queuing and scheduling.

Table 11: Fine-Grained Queuing and Scheduling Support by MIC or MPC Type

MPC	MIC	Supported	Example Configuration
Fixed Configuration MPCs			
16x10GE MPC	No	Yes	[edit class-of-service interfaces ge-0/0/0 unit 1] scheduler-map map-1;
32x10GE MPC4E	No	Yes	[edit class-of-service interfaces ge-0/0/0 unit 1] scheduler-map map-1;
2x100GE + 8x10GE MPC4E	No	Yes	[edit class-of-service interfaces ge-0/0/0 unit 1] scheduler-map map-1;
6x40GE + 24x10GE MPC5E	No	No	No
6x40GE + 24x10GE MPC5EQ	No	Yes	[edit class-of-service interfaces ge-0/0/0 unit 1] scheduler-map map-1;
2x100GE + 4x10GE MPC5E	No	No	No
2x100GE + 4x10GE MPC5EQ	No	Yes	[edit class-of-service interfaces ge-0/0/0 unit 1] scheduler-map map-1;
MPCs			
MPC1	No	No	No
MPC1E	No	No	No
MPC1 Q	Any supported MIC	Yes	Example of supported configuration: [edit class-of-service interfaces ge-0/0/0 unit 1] scheduler-map map-1;

Table 11: Fine-Grained Queuing and Scheduling Support by MIC or MPC Type (*continued*)

MPC	MIC	Supported	Example Configuration
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;
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;
MPC6E	Any supported MIC	Yes	Example of supported configuration: [edit class-of-service interfaces et-0/0/0 unit 1] scheduler-map map-1;

To configure scheduling on logical interfaces:

1. Enable per-unit scheduling on the interface by including the **per-unit-scheduler** statement at the **[edit interfaces *interface-name*]** hierarchy level:

```
[edit interfaces interface-name]
per-unit-scheduler;
```

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

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

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

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

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

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

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

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

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

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



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

By default, the logical interface bandwidth is the average of unused bandwidth for the number of logical interfaces that require default bandwidth treatment. You can specify a peak bandwidth rate in bps, either as a complete decimal number or as a decimal number followed by the abbreviation **k** (1000), **m** (1,000,000), or **g** (1,000,000,000). The range is from 1000 through 6,400,000,000,000 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, MPC4E, and MPC6E, the minimum is 292,000 bps.

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

- Related Documentation**
- *per-unit-scheduler*
 - *Example: Applying Scheduling and Shaping to VLANs*
 - *Example: Applying Scheduler Maps and Shaping Rate to DLCIs*

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 [CLI Explorer](#).

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 49](#). For information about logical interface scheduling configuration, see [“Applying Scheduler Maps and Shaping Rate to DLCIs and VLANs” on page 50](#).

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, **ge-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 59](#). 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 58](#)
- [Examples: Scheduling Packet Forwarding Component Queues on page 59](#)

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 49](#).

Examples: Scheduling Packet Forwarding Component Queues

Applying a Chassis Scheduler Map to a 2-Port IQ PIC

Apply a chassis scheduler map to interfaces **ge-0/1/0** and **ge-0/1/1**.

According to customary wildcard rules, the **ge-0/1/0** configuration overrides the **ge-0/1/*** configuration, implying that the chassis scheduler map **MAP1** is not applied to **ge-0/1/0**. However, the wildcard rule is not obeyed in this case; the chassis scheduler map applies to both interfaces **ge-0/1/0** and **ge-0/1/1**.

```
[edit]
class-of-service {
  interfaces {
    ge-0/1/0 {
      unit 0 {
        classifiers {
          inet-precedence default;
        }
      }
    }
    ge-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 Two T3 Interfaces on a

```
[edit interfaces]
ct3-3/0/0 {
```

**Channelized DS3 IQ
PIC**

```

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 **ge-3/0/0.0**:

```

[edit class-of-service]
interfaces {
    ge-3/0/* {
        unit 0 {
            scheduler-map MY_SCHED_MAP;
        }
    }
    ge-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 {
    ge-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 4

Configuration Statements for Schedulers

- [\[edit class-of-service\] Hierarchy Level](#) on page 64
- [buffer-size \(Schedulers\)](#) on page 68
- [delay-buffer-rate](#) on page 69
- [drop-profile-map \(Schedulers\)](#) on page 70
- [excess-priority](#) on page 71
- [excess-rate](#) on page 72
- [fabric \(Class-of-Service\)](#) on page 73
- [forwarding-class \(Interfaces\)](#) on page 73
- [guaranteed-rate](#) on page 74
- [interfaces \(CoS\)](#) on page 75
- [loss-priority \(Scheduler Drop Profiles\)](#) on page 77
- [output-traffic-control-profile](#) on page 78
- [priority \(Fabric Queues, Schedulers\)](#) on page 79
- [priority \(Schedulers\)](#) on page 80
- [protocol \(Schedulers\)](#) on page 81
- [scheduler \(Fabric Queues\)](#) on page 82
- [scheduler \(Scheduler Map\)](#) on page 82
- [scheduler-map \(Fabric Queues\)](#) on page 83
- [scheduler-map \(Interfaces and Traffic-Control Profiles\)](#) on page 83
- [scheduler-map-chassis](#) on page 84
- [scheduler-maps \(For Most Interface Types\)](#) on page 85
- [schedulers \(CoS\)](#) on page 86
- [shaping-rate \(Applying to an Interface\)](#) on page 87
- [shaping-rate \(Oversubscribing an Interface\)](#) on page 89
- [traffic-control-profiles](#) on page 90
- [transmit-rate \(Schedulers\)](#) on page 91
- [unit](#) on page 93

- [\[edit interfaces\] Hierarchy Level](#) on page 93
- [schedulers \(Interfaces\)](#) on page 104

[\[edit class-of-service\] Hierarchy Level](#)

```

class-of-service {
  classifiers {
    type classifier-name {
      forwarding-class class-name {
        loss-priority (high | low | medium-high | medium-low) code-points [ aliases bits ];
      }
      import (classifier-name | default);
    }
  }
  code-point-aliases {
    (dscp | dscp-ipv6 | exp | ieee-802.1 | ieee-802.1ad | inet-precedence) {
      alias-name bits;
    }
  }
  drop-profiles {
    profile-name {
      fill-level percentage drop-probability percentage;
      interpolate {
        drop-probability value;
        fill-level value;
      }
    }
  }
  fabric {
    scheduler-map {
      priority (high | low) scheduler scheduler-name;
    }
  }
  forwarding-class-map {
    map-name {
      class class-name queue-num queue-number <restricted-queue queue-number>;
    }
  }
  forwarding-classes {
    class class-name policing-priority (normal | premium) queue-num queue-number
      priority (high | low);
    queue queue-number class-name policing-priority (normal | premium) priority (high |
      low);
  }
  forwarding-policy {
    class class-name {
      classification-override {
        forwarding-class class-name;
      }
    }
  }
  next-hop-map map-name {
    forwarding-class class-name {
      discard;
      lsp-next-hop [ lsp-regular-expressions ];
      next-hop [ next-hop-names ];
    }
  }
}

```

```

        non-lsp-next-hop;
    }
}
fragmentation-maps {
    map-name {
        forwarding-class class-name {
            drop-timeout milliseconds;
            fragment-threshold bytes;
            multilink-class number;
            no-fragmentation;
        }
    }
}
host-outbound-traffic {
    dscp-code-point value;
    forwarding-class class-name;
    ieee-802.1 {
        default value;
        rewrite-rules;
    }
    tcp {
        raise-internet-control-priority;
    }
}
interfaces {
    ... the interfaces subhierarchy appears after the main [edit class-of-service] hierarchy
    ...
}
restricted-queues {
    forwarding-class class-name queue-number;
}
rewrite-rules {
    (dscp | dscp-ipv6 | exp | frame-relay-de | ieee-802.1 | ieee-802.1ad | inet-precedence)
    rewrite-rule {
        forwarding-class class-name {
            loss-priority level code-point (alias | bits);
        }
        import (rewrite-rule | default);
    }
}
routing-instances routing-instance-name {
    classifiers {
        dscp (classifier-name | default);
        dscp-ipv6 (classifier-name | default);
        exp (classifier-name | default);
        ieee-208.1 (classifier-name | default | encapsulated | vlan-tag (inner | outer));
    }
}
scheduler-maps {
    map-name {
        forwarding-class class-name scheduler scheduler-name;
    }
}
schedulers {

```

```

scheduler-name {
    adjust-minimum value;
    adjust-percent value;
    buffer-size (exact | percent percentage | remainder);
    drop-profile-map loss-priority (any | high | low | medium-high | medium-low)
        protocol any;
    excess-priority (high | low | medium-high | medium-low);
    excess-rate (percent percentage | proportion proportion);
    priority (high | low | medium-high | medium-low | strict-high);
    shaping-rate (bps | percent percentage | burst-size size);
    transmit-rate (bps | percent percentage | remainder) <exact | rate-limit>;
}
}
traceoptions {
    file <files number> <match regular-expression> <size maximum-file-size>
        <world-readable | no-world-readable>;
    flag flag;
    no-remote-trace;
}
traffic-control-profiles {
    profile-name {
        adjust-minimum rate;
        delay-buffer-rate (bps | cps cps | percent percentage);
        excess-rate (percent percentage | proportion value);
        guaranteed-rate (bps | percent percentage) <burst-size bytes>;
        overhead-accounting (frame-mode | cell-mode) <bytes byte-value>;
        scheduler-map map-name;
        shaping-rate (bps | percent percentage) <burst-size bytes>;
    }
}
tri-color;
}

class-of-service {
    interfaces {
        interface-name {
            excess-bandwidth-share (equal | proportional value);
            input-excess-bandwidth-share (equal | proportional value);
            input-scheduler-map map-name;
            input-shaping-rate bps;
            input-traffic-control-profile profile-name;
            output-forwarding-class-map map-name;
            output-traffic-control-profile profile-name;
            scheduler-map map-name;
            scheduler-map-chassis (map-name | derived);
            shaping-rate bps;
            unit (logical-unit-number | *) {
                classifiers {
                    dscp (classifier-name | default) {
                        family [ inet mpls ];
                    }
                    dscp-ipv6 (classifier-name | default) {
                        family [ inet mpls ];
                    }
                    exp (classifier-name | default);
                    ieee-208.1 (classifier-name | default) <vlan-tag (inner | outer)>;
                }
            }
        }
    }
}

```



```

        ieee-208.1ad (classifier-name | default);
        inet-precedence (classifier-name | default);
    }
    forwarding-class class-name;
    input-scheduler-map map-name;
    input-shaping-rate bps;
    input-traffic-control-profile profile-name shared-instance instance-name;
    loss-priority-maps {
        (map-name | default);
    }
    loss-priority-rewrites {
        (map-name | default);
    }
    output-forwarding-class-map map-name;
    output-traffic-control-profile profile-name shared-instance instance-name;
    rewrite-rules {
        dscp (rule-name | default) <protocol mpls>;
        dscp-ipv6 (rule-name | default);
        exp (rule-name | default) <protocol [ mpls-any | mpls-inet-both |
            mpls-inet-both-non-vpn ]>;
        exp-push-push-push default;
        exp-swap-push-push default;
        ieee-802.1 (rewrite-name | default) <vlan-tag (outer | outer-and-inner)>;
        ieee-802.1ad (rewrite-name | default) <vlan-tag (outer | outer-and-inner)>;
        inet-precedence (rewrite-name | default) <protocol mpls>;
    }
    scheduler-map map-name;
    shaping-rate bps;
    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 (equal | proportional value);
    input-excess-bandwidth-share (equal | proportional value);
    input-traffic-control-profile profile-name;
    input-traffic-control-profile-remaining profile-name;
    internal-node;
    output-traffic-control-profile profile-name;
    output-traffic-control-profile-remaining profile-name;
}
}
}

```

Related Documentation

- *Notational Conventions Used in Junos OS Configuration Hierarchies*

buffer-size (Schedulers)

Syntax	buffer-size (percent <i>percentage</i> remainder temporal <i>microseconds</i>);
Hierarchy Level	[edit class-of-service schedulers <i>scheduler-name</i>]
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



NOTE: For the routers with channelized OC12/STM4 IQE PIC with SFP (PB-4CHOC12-STM4-IQE-SFP) and channelized OC48/STM16 IQE PIC with SFP (PB-1CHOC48-STM16-IQE-SFP), the minimum buffer allocated to any queue is 18,432 bytes. If a queue is configured to have a buffer size less than 18K, the queue retains a buffer size of 18,432 bytes.

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.

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

Related Documentation

- [Managing Congestion on the Egress Interface by Configuring the Scheduler Buffer Size on page 12](#)
- [Table 3 on page 13](#)

delay-buffer-rate

Syntax	<code>delay-buffer-rate (percent <i>percentage</i> <i>rate</i>);</code>
Hierarchy Level	[edit class-of-service traffic-control-profiles <i>profile-name</i>]
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 8 on page 37 .
Options	<p>percent <i>percentage</i>—For LSQ interfaces, delay-buffer rate as a percentage of the available interface bandwidth.</p> <p>Range: 1 through 100 percent</p> <p><i>rate</i>—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).</p> <p>Range: 1000 through 6,400,000,000,000 bps</p>
Required Privilege Level	<p>interface—To view this statement in the configuration.</p> <p>interface-control—To add this statement to the configuration.</p>
Related Documentation	<ul style="list-style-type: none"> • Oversubscribing Interface Bandwidth on page 33 • Providing a Guaranteed Minimum Rate on page 42 • Configuring Traffic Control Profiles for Shared Scheduling and Shaping • output-traffic-control-profile on page 78

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) <i>profile-name</i> ;
Hierarchy Level	[edit class-of-service schedulers <i>scheduler-name</i>]
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	<ul style="list-style-type: none">• Default Schedulers Overview on page 4• Determining Packet Drop Behavior by Configuring Drop Profile Maps for Schedulers on page 23

excess-priority


Syntax	<code>excess-priority [low medium-low medium-high high none];</code>
Hierarchy Level	[edit class-of-service schedulers <i>scheduler-name</i>]
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	<p>low—Excess traffic for this scheduler has low priority.</p> <p>medium-low—Excess traffic for this scheduler has medium-low priority.</p> <p>medium-high—Excess traffic for this scheduler has medium-high priority.</p> <p>high—Excess traffic for this scheduler has high priority.</p> <p>none—System does not demote the priority of guaranteed traffic when the bandwidth exceeds the shaping rate or the guaranteed rate.</p>
Required Privilege Level	<p>interface—To view this statement in the configuration.</p> <p>interface-control—To add this statement to the configuration.</p>
Related Documentation	<ul style="list-style-type: none"> • <i>Configuring Excess Bandwidth Sharing on IQE PICs</i> • <i>Bandwidth Sharing on Nonqueueing Packet Forwarding Engines Overview</i> • <i>Managing Excess Bandwidth Distribution on Static Interfaces on MICs and MPCs</i>

excess-rate

Syntax	<code>excess-rate (percent <i>percentage</i> proportion <i>value</i>);</code>
Hierarchy Level	[edit class-of-service schedulers <i>scheduler-name</i>], [edit class-of-service traffic-control-profiles <i>traffic-control-profile-name</i>]
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.
<div>  NOTE: The proportion option provides a greater range of values over the percent option and hence influences the priorities assigned to the queues. </div>	
Options	<p>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.</p> <p>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 <i>traffic-control-profile-name</i>] hierarchy level only. Range: 0 through 1000</p>
Required Privilege Level	<p>interface—To view this statement in the configuration.</p> <p>interface-control—To add this statement to the configuration.</p>
Related Documentation	<ul style="list-style-type: none"> • Configuring Scheduler Transmission Rate on page 24 • 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	<pre>fabric { scheduler-map { priority (high low) scheduler scheduler-name; } }</pre>
Hierarchy Level	[edit class-of-service]
Release Information	<p>Statement introduced before Junos OS Release 7.4.</p> <p>Statement introduced before Junos OS 11.4 for EX Series switches.</p>
Description	<p>Define CoS parameters of the switch fabric. For M320 and T Series routers only, associate a scheduler with a fabric priority.</p> <p>On EX Series switches, this statement is supported only on EX8200 standalone switches and EX8200 Virtual Chassis.</p> <p>The remaining statements are explained separately.</p>
Required Privilege Level	<p>interface—To view this statement in the configuration.</p> <p>interface-control—To add this statement to the configuration.</p>
Related Documentation	<ul style="list-style-type: none"> See Associating Schedulers with Fabric Priorities on page 45.

forwarding-class (Interfaces)

Syntax	forwarding-class <i>class-name</i> ;
Hierarchy Level	[edit class-of-service interfaces <i>interface-name</i> unit <i>logical-unit-number</i>]
Release Information	<p>Statement introduced before Junos OS Release 7.4.</p> <p>Statement introduced in Junos OS Release 12.2 for ACX Series routers.</p>
Description	Associate a forwarding class configuration or default mapping with a specific interface.
Options	<i>class-name</i> —Name of the forwarding class.
Required Privilege Level	<p>interface—To view this statement in the configuration.</p> <p>interface-control—To add this statement to the configuration.</p>
Related Documentation	<ul style="list-style-type: none"> <i>Applying Forwarding Classes to Interfaces</i>

guaranteed-rate

Syntax	<code>guaranteed-rate (percent <i>percentage</i> <i>rate</i>) <burst-size <i>bytes</i>>;</code>
Hierarchy Level	[edit class-of-service traffic-control-profiles <i>profile-name</i>]
Release Information	<p>Statement introduced in Junos OS Release 7.6.</p> <p>Option burst-size introduced for Enhanced Queuing (EQ) DPC interfaces in Junos OS Release 9.4.</p> <p>Option burst-size introduced for MIC and MPC interfaces in Junos OS Release 11.4.</p> <p>Option burst-size introduced for IQ2 and IQ2E interfaces in Junos OS Release 12.3</p>
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	<p>percent <i>percentage</i>—For LSQ interfaces, guaranteed rate as a percentage of the available interface bandwidth.</p> <p>Range: 1 through 100 percent</p> <p><i>rate</i>—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).</p> <p>Range: 1000 through 6,400,000,000,000 bps</p> <p>burst-size <i>bytes</i>—(Optional) Maximum burst size, in bytes.</p>
Required Privilege Level	<p>interface—To view this statement in the configuration.</p> <p>interface-control—To add this statement to the configuration.</p>
Related Documentation	<ul style="list-style-type: none">• Providing a Guaranteed Minimum Rate on page 42• Configuring Traffic Control Profiles for Shared Scheduling and Shaping• output-traffic-control-profile on page 78

interfaces (CoS)

```
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.

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

- *Understanding How Behavior Aggregate Classifiers Prioritize Trusted Traffic*
- *Configuring Rewrite Rules*

loss-priority (Scheduler Drop Profiles)

Syntax	loss-priority (any high low medium-high medium-low);
Hierarchy Level	[edit class-of-service schedulers <i>scheduler-name</i> 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	interface—To view this statement in the configuration.
Level	interface-control—To add this statement to the configuration.

Related Documentation	<ul style="list-style-type: none"> • Default Schedulers Overview on page 4 • Determining Packet Drop Behavior by Configuring Drop Profile Maps for Schedulers on page 23 • Configuring Schedulers for Priority Scheduling on page 27 • Configuring Tricolor Marking • protocol (Schedulers) on page 81
------------------------------	---

output-traffic-control-profile


Syntax	<code>output-traffic-control-profile <i>profile-name</i> shared-instance <i>instance-name</i>;</code>
Hierarchy Level	<code>[edit class-of-service interfaces <i>interface-name</i>],</code> <code>[edit class-of-service interfaces <i>interface-name</i> <i>unit</i> <i>logical-unit-number</i>],</code> <code>[edit class-of-service interfaces <i>interface-name</i> interface-set <i>interface-set-name</i>]</code>
Release Information	<p>Statement introduced in Junos OS Release 7.6.</p> <p>interface-set option added for Enhanced Queuing DPCs on MX Series routers in Junos OS Release 8.5.</p> <p>interface-set option added for MIC and MPC interfaces on MX Series routers in Junos OS Release 10.2.</p> <p>Support on GRE tunnel interfaces configured on physical and logical interfaces on MICs or MPCs in MX Series routers added in Junos OS Release 13.3.</p>
Description	<p>Apply the specified CoS traffic control profile (traffic scheduling and shaping configuration objects) to the output traffic at the physical interface, logical interface, or interface set.</p> <p>The statement is supported on the following interfaces:</p> <ul style="list-style-type: none"> • Channelized IQ PIC interfaces • Gigabit Ethernet IQ, Gigabit Ethernet IQ2, and IQ2E PIC interfaces • Link services IQ (LSQ) interfaces on AS PICs • Enhanced Queuing DPC, MIC, and MPC interfaces on MX Series routers • GRE tunnel interfaces configured on physical or logical interfaces hosted on MIC or MPC line cards in MX Series routers.
	<p> NOTE: Interface sets (sets of interfaces used to configure hierarchical CoS schedulers on supported Ethernet interfaces) are not supported on GRE tunnel interfaces.</p>
	<p>The shared-instance statement is supported on Gigabit Ethernet IQ2 PICs only.</p>
Options	<i>profile-name</i> —Name of the traffic-control profile to be applied to this interface
Required Privilege Level	<p>interface—To view this statement in the configuration.</p> <p>interface-control—To add this statement to the configuration.</p>
Related Documentation	<ul style="list-style-type: none"> • Oversubscribing Interface Bandwidth on page 33 • <i>Configuring Traffic Control Profiles for Shared Scheduling and Shaping</i> • <i>Configuring Hierarchical Schedulers for CoS</i> (Enhanced Queuing DPC, MIC, and MPC interfaces on MX Series routers)

- *Configuring Interface Sets* (Enhanced Queuing DPC, MIC, and MPC interfaces on MX Series routers)
- *output-traffic-control-profile-remaining*
- [traffic-control-profiles on page 90](#)

priority (Fabric Queues, Schedulers)

Syntax	<code>priority (high low)scheduler scheduler-name;</code>
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	<p>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.</p> <p>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.</p> <p>On EX Series switches, this statement is supported only on EX8200 standalone switches and EX8200 Virtual Chassis.</p>
Options	<p>high—Scheduler has high priority.</p> <p>low—Scheduler has low priority.</p> <p>The remaining statements are explained separately.</p>
Required Privilege Level	<p>interface—To view this statement in the configuration.</p> <p>interface-control—To add this statement to the configuration.</p>
Related Documentation	<ul style="list-style-type: none">• See Associating Schedulers with Fabric Priorities on page 45.

priority (Schedulers)

Syntax	<code>priority <i>priority-level</i>;</code>
Hierarchy Level	[edit class-of-service schedulers <i>scheduler-name</i>]
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	<p><i>priority-level</i> can be one of the following:</p> <ul style="list-style-type: none">• 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.
<div> 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.</div>	
Required Privilege Level	interface—To view this statement in the configuration. interface-control—To add this statement to the configuration.
Related Documentation	<ul style="list-style-type: none">• Configuring Schedulers for Priority Scheduling on page 27

protocol (Schedulers)

Syntax	protocol (any non-tcp tcp);
Hierarchy Level	[edit class-of-service schedulers <i>scheduler-name</i> 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	<ul style="list-style-type: none"> • Configuring Schedulers on page 11

scheduler (Fabric Queues)

Syntax	<code>scheduler <i>scheduler-name</i>;</code>
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 Release 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	<i>scheduler-name</i> —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	<ul style="list-style-type: none">See Associating Schedulers with Fabric Priorities on page 45.

scheduler (Scheduler Map)

Syntax	<code>scheduler <i>scheduler-name</i>;</code>
Hierarchy Level	[edit class-of-service scheduler-maps <i>map-name</i>]
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	<i>scheduler-name</i> —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	<ul style="list-style-type: none">Configuring Schedulers on page 11

scheduler-map (Fabric Queues)

Syntax	<code>scheduler-map priority (high low) scheduler <i>scheduler-name</i>;</code>
Hierarchy Level	<code>[edit class-of-service fabric]</code>
Release Information	Statement introduced before Junos OS Release 7.4. Statement introduced before Junos OS Release 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	<ul style="list-style-type: none"> See Associating Schedulers with Fabric Priorities on page 45.

scheduler-map (Interfaces and Traffic-Control Profiles)

Syntax	<code>scheduler-map <i>map-name</i>;</code>
Hierarchy Level	<code>[edit class-of-service interfaces <i>interface-name</i>],</code> <code>[edit class-of-service interfaces <i>interface-name</i> unit <i>logical-unit-number</i>],</code> <code>[edit class-of-service traffic-control-profiles]</code>
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	<i>map-name</i> —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	<ul style="list-style-type: none"> Configuring Schedulers on page 11 Oversubscribing Interface Bandwidth on page 33 output-traffic-control-profile on page 78

scheduler-map-chassis

Syntax	<code>scheduler-map-chassis (derived <i>map-name</i>);</code>
Hierarchy Level	[edit class-of-service interfaces <i>interface-type-fpc/pic/*</i>]
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. <i>map-name</i> —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	<ul style="list-style-type: none">• <i>Applying Scheduler Maps to Packet Forwarding Component Queues</i>• scheduler-map (Fabric Queues) on page 83


scheduler-maps (For Most Interface Types)

Syntax	<pre>scheduler-maps { map-name { forwarding-class class-name scheduler scheduler-name; } }</pre>
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	<p><i>map-name</i>—Name of the scheduler map.</p> <p>The remaining statements are explained separately.</p> <p>See “Configuring Schedulers” on page 11.</p>
Required Privilege Level	<p>interface—To view this statement in the configuration.</p> <p>interface-control—To add this statement to the configuration.</p>

schedulers (CoS)

Syntax	<pre>schedulers { scheduler-name { adjust-minimum <i>rate</i>; adjust-percent <i>percentage</i>; buffer-size (<i>seconds</i> percent <i>percentage</i> remainder temporal <i>microseconds</i>); drop-profile-map loss-priority (any low medium-low medium-high high) <i>protocol</i> (any non-tcp tcp) drop-profile <i>profile-name</i>; excess-priority [low medium-low medium-high high none]; excess-rate (percent <i>percentage</i> proportion <i>value</i>); priority <i>priority-level</i>; shaping-rate (percent <i>percentage</i> <i>rate</i>); transmit-rate (percent <i>percentage</i> <i>rate</i> remainder) <exact rate-limit>; } }</pre>
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	<p>scheduler-name—Name of the scheduler to be configured.</p> <p>The remaining statements are explained separately.</p>
Required Privilege Level	interface—To view this statement in the configuration. interface-control—To add this statement to the configuration.
Related Documentation	<ul style="list-style-type: none">• Schedulers Overview on page 3• Default Schedulers Overview on page 4• Configuring Schedulers on page 11• Configuring a Scheduler

shaping-rate (Applying to an Interface)

Syntax	<code>shaping-rate rate;</code>
Hierarchy Level	<code>[edit class-of-service interfaces <i>interface-name</i>],</code> <code>[edit class-of-service interfaces <i>interface-name</i> unit <i>logical-unit-number</i>]</code>
Release Information	Statement introduced before Junos OS Release 7.4. <code>[edit class-of-service interfaces <i>interface-name</i>]</code> hierarchy level added in Junos OS Release 7.5. Statement introduced in Junos OS Release 13.2 on PTX Series Packet Transport Routers.
Description	<p>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.</p> <p>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.</p> <p>Logical and physical interface traffic shaping rates are mutually exclusive. This means you can include the shaping-rate statement at the <code>[edit class-of-service interfaces <i>interface-name</i>]</code> hierarchy level or the <code>[edit class-of-service interfaces <i>interface-name</i> unit <i>logical-unit-number</i>]</code> hierarchy level, but not both.</p>
	<p> NOTE: For MX Series routers and for EX Series switches, the shaping rate value for the physical interface at the <code>[edit class-of-service interfaces <i>interface-name</i>]</code> 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.</p> <p>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.</p>
	<p>Alternatively, you can configure a shaping rate for a logical interface and oversubscribe the physical interface by including the shaping-rate statement at the <code>[edit class-of-service traffic-control-profiles]</code> hierarchy level. With this configuration approach, you can independently control the delay-buffer rate, as described in “Oversubscribing Interface Bandwidth” on page 33.</p> <p>For FRF.15 and FRF.16 bundles on link services interfaces, only shaping rates based on percentage are supported.</p>
Default	If you do not include this statement at the <code>[edit class-of-service interfaces <i>interface-name</i> unit <i>logical-unit-number</i>]</code> 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 <code>[edit class-of-service interfaces <i>interface-name</i>]</code> 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: 1000 through 6,400,000,000,000 bps.



NOTE: For all MX Series and EX series interfaces, the rate can be from 65,535 through 6,400,000,000,000 bps.

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

Related Documentation

- [Applying Scheduler Maps Overview on page 6](#)
- *Configuring Virtual LAN Queuing and Shaping on PTX Series Routers*

shaping-rate (Oversubscribing an Interface)

Syntax	<code>shaping-rate (percent <i>percentage</i> <i>rate</i>) <burst-size <i>bytes</i>>;</code>
Hierarchy Level	[edit class-of-service traffic-control-profiles <i>profile-name</i>]
Release Information	<p>Statement introduced in Junos OS Release 7.6.</p> <p>Option burst-size introduced for Enhanced Queuing (EQ) DPC interfaces on MX Series routers in Junos OS Release 9.4.</p> <p>Option burst-size option introduced for MIC and MPC interfaces on MX Series routers in Junos OS Release 11.4.</p> <p>Option burst-size introduced for IQ2 and IQ2E interfaces in Junos OS Release 12.3.</p>
Description	<p>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.</p> <p>For physical interfaces on T4000 router interfaces on Type 5 FPCs, configure traffic shaping rate.</p> <p>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).</p>
Default	The default behavior depends on various factors. For more information, see Table 8 on page 37 .
Options	<p>percent <i>percentage</i>—For LSQ interfaces, shaping rate as a percentage of the available interface bandwidth.</p> <p>Range: 1 through 100 percent</p> <p><i>rate</i>—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).</p> <p>Range: IQ and IQ2 interfaces—1000 through 6,400,000,000,000 bps</p> <p>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.</p> <p>burst-size <i>bytes</i>—(Optional) Maximum burst size, in bytes.</p> <p>Range: 0 through 1,000,000,000</p>
Required Privilege Level	<p>interface—To view this statement in the configuration.</p> <p>interface-control—To add this statement to the configuration.</p>

- Related Documentation**
- [Configuring Traffic Control Profiles for Shared Scheduling and Shaping](#)
 - [Oversubscribing Interface Bandwidth on page 33](#)
 - [output-traffic-control-profile on page 78](#)

traffic-control-profiles

Syntax	<pre>traffic-control-profiles <i>profile-name</i> { adjust-minimum <i>rate</i>; atm-service (cbr rtvbr nrtvbr); delay-buffer-rate (percent <i>percentage</i> <i>rate</i>); excess-rate (percent <i>percentage</i> proportion <i>value</i>); excess-rate-high (percent <i>percentage</i> proportion <i>value</i>); excess-rate-low (percent <i>percentage</i> proportion <i>value</i>); guaranteed-rate (percent <i>percentage</i> <i>rate</i>) <burst-size <i>bytes</i>>; max-burst-size <i>cells</i>; overhead-accounting (frame-mode cell-mode frame-mode-bytes cell-mode-bytes) <bytes (<i>byte-value</i>)>; peak-rate <i>rate</i>; scheduler-map <i>map-name</i>; shaping-rate (percent <i>percentage</i> <i>rate</i>) <burst-size <i>bytes</i>>; shaping-rate-excess-high <i>rate</i> [burst-size <i>bytes</i>]; shaping-rate-excess-low <i>rate</i> [burst-size <i>bytes</i>]; shaping-rate-priority-high <i>rate</i> [burst-size <i>bytes</i>]; shaping-rate-priority-low <i>rate</i> [burst-size <i>bytes</i>]; shaping-rate-priority-medium <i>rate</i> [burst-size <i>bytes</i>]; strict-priority-scheduler; sustained-rate <i>rate</i>; }</pre>
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, Enhanced Queuing (EQ) DPCs, and PTX Series routers only, configure traffic shaping and scheduling profiles. For Enhanced EQ PICs, EQ DPCs, and PTX Series routers only, you can include the excess-rate statement.
Options	<p><i>profile-name</i>—Name of the traffic-control profile.</p> <p>The remaining statements are explained separately.</p>
Required Privilege Level	<p>interface—To view this statement in the configuration.</p> <p>interface-control—To add this statement to the configuration.</p>
Related Documentation	<ul style="list-style-type: none"> • Oversubscribing Interface Bandwidth on page 33 • Understanding Scheduling on PTX Series Routers • output-traffic-control-profile on page 78

transmit-rate (Schedulers)

Syntax	<code>transmit-rate (rate percent <i>percentage</i> remainder) <exact rate-limit>;</code>
Hierarchy Level	[edit class-of-service schedulers <i>scheduler-name</i>]
Release Information	<p>Statement introduced before Junos OS Release 7.4.</p> <p>rate-limit option introduced in Junos OS Release 8.3. Applied to the Multiservices PICs in Junos OS Release 9.4.</p> <p>Statement introduced in Junos OS Release 12.1X48 for PTX Series Packet Transport Routers.</p> <p>Statement introduced in Junos OS Release 12.2 for ACX Series Routers.</p>
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	<p>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.</p> <p>percent <i>percentage</i>—Percentage of transmission capacity. A percentage of zero drops all packets in the queue.</p> <p>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</p>



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 `schedulers` *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 6,400,000,000,000 bps



NOTE: For all MX Series interfaces, the rate can be from 65,535 through 6,400,000,000,000 bps.

rate-limit—(Optional) Limit the transmission rate to the rate-controlled amount by applying a policing action to the queue. Packets are hard-dropped when traffic exceeds the specified maximum transmission rate.



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. The **rate-limit** option cannot rate limit the queue if strict-priority scheduling is configured with the *strict-priority-scheduler* statement.



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	<ul style="list-style-type: none">• Configuring Schedulers on page 11• Configuring Scheduler Transmission Rate on page 24• Understanding Scheduling on PTX Series Routers
------------------------------	---

unit

Syntax	<pre> 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; } </pre>
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	<p>logical-unit-number—Number of the logical unit.</p> <p>Range: 0 through 16,384</p> <p>The remaining statements are explained separately.</p>
Required Privilege Level	<p>interface—To view this statement in the configuration.</p> <p>interface-control—To add this statement to the configuration.</p>
Related Documentation	<ul style="list-style-type: none"> Understanding How Behavior Aggregate Classifiers Prioritize Trusted Traffic Configuring Rewrite Rules

[edit interfaces] Hierarchy Level

The following statement hierarchy can also be included at the **[edit logical-systems logical-system-name]** hierarchy level.

```

interfaces {
  interface-name {
    ... the "interface-name" subhierarchy appears after the main [edit interfaces] hierarchy level ...
  }
}

```

```

}
interface-set interface-set-name {
  interface interface-name {
    (unit unit-number | vlan-tags-outer vlan-tag);
  }
}
}
irb {
  accounting-profile name;
  description text;

  (gratuitous-arp-reply | no-gratuitous-arp-reply);
  hold-time up milliseconds down milliseconds;
  mtu bytes;
  no-gratuitous-arp-request;

  traceoptions {
    flag flag;
  }
  (traps | no-traps);
  unit logical-unit-number {
    accounting-profile name;
    bandwidth rate;
    description text;
    disable;
    encapsulation type;
    family inet {
      accounting {
        destination-class-usage;
        source-class-usage {
          input;
          output;
        }
      }
    }
  }
  address ipv4-address {
    arp ip-address (mac | multicast-mac) mac-address <publish>;
    broadcast address;
    preferred;
    primary;
    vrrp-group group-id {
      (accept-data | no-accept-data);
      advertise-interval seconds;
      advertisements-threshold number;
      authentication-key key;
      authentication-type authentication;
      fast-interval milliseconds;
      (preempt | no-preempt) {
        hold-time seconds;
      }
    }
    priority number;
    track {
      interface interface-name {
        bandwidth-threshold bits-per-second priority-cost priority;
        priority-cost priority;
      }
    }
    priority-hold-time seconds;
  }
}

```

```

        route prefix/prefix-length routing-instance instance-name priority-cost priority;
    }
    virtual-address [ addresses ];
    vrrp-inherit-from vrrp-group;
}
}
filter {
    input filter-name;
    output filter-name;
}
mtu bytes;
no-neighbor-learn;
no-redirects;
primary;
rpf-check {
    fail-filter filter-name;
    mode {
        loose;
    }
}
targeted-broadcast {
    forward-and-send-to-re;
    forward-only;
}
}
family inet6 {
    accounting {
        destination-class-usage;
        source-class-usage {
            input;
            output;
        }
    }
}
address address {
    eui-64;
    ndp ip-address (mac | multicast-mac) mac-address <publish>;
    preferred;
    primary;
    vrrp-inet6-group group-id {
        accept-data | no-accept-data;
        advertisements-threshold number;
        authentication-key key;
        authentication-type authentication;
        fast-interval milliseconds;
        inet6-advertise-interval milliseconds;
        preempt | no-preempt {
            hold-time seconds;
        }
    }
    priority number;
    track {
        interface interface-name {
            bandwidth-threshold bandwidth priority-cost number;
            priority-cost number;
        }
    }
    priority-hold-time seconds;
    route ip-address/mask routing-instance instance-name priority-cost cost;
}

```

```

    }
    virtual-inet6-address [addresses];
    virtual-link-local-address ipv6-address;
    vrrp-inherit-from {
        active-group group-number;
        active-interface interface-name;
    }
}
(dad-disable | no-dad-disable);
filter {
    input filter-name;
    output filter-name;
}
mtu bytes;
nd6-stale-time seconds;
no-neighbor-learn;
no-redirects;
policer {
    input policer-name;
    output policer-name;
}
rpf-check {
    fail-filter filter-name;
    mode {
        loose;
    }
}
}
family iso {
    address interface-address;
    mtu bytes;
}
family mpls {
    filter {
        input filter-name;
        output filter-name;
    }
    mtu bytes;
    policer {
        input policer-name;
        output policer-name;
    }
}
native-inner-vlan-id vlan-id;
proxy-arp (restricted | unrestricted);
(traps | no-traps);
vlan-id-list [vlan-id's];
vlan-id-range [vlan-id-range];
}
}
traceoptions {
    file <filename> <files number> <match regular-expression> <size maximum-file-size>
        <world-readable | no-world-readable>;
    flag flag <disable>;
    no-remote-trace;
}

```

```

    }
  }

interfaces {
  interface-name {
    disable;
    accounting-profile name;
    aggregated-ether-options {
      ethernet-switch-profile {
        tag-protocol-id [ hexadecimal-identifiers ];
      }
      (flow-control | no-flow-control);
      lacp {
        (active | passive);
        admin-key key;
        fast-failover;
        link-protection {
          disable;
          (revertive | non-revertive);
        }
        periodic (fast | slow);
        system-id mac-address;
        system-priority priority;
      }
      (link-protection | no-link-protection);
      link-speed (100m | 1g | 8g | 10g | 40g | 50g | 80g | 100g | oc192);
      logical-interface-fpc-redundancy;
      (loopback | no-loopback);
      mc-ae {
        chassis-id chassis-id;
        events {
          iccp-peer-down {
            force-icl-down;
            prefer-status-control-active;
          }
        }
        mc-ae-id mc-ae-id;
        mode (active-active | active-standby);
        redundancy-group group-id;
        status-control (active | standby);
      }
      minimum-links number;
      rebalance-periodic {
        start-time time;
        interval number;
      }
      source-address-filter {
        mac-address;
      }
      (source-filtering | no-source-filtering);
    }
  }
  auto-configure {
    remove-when-no-subscribers;
    stacked-vlan-ranges {
      access-profile profile-name;
      authentication {

```

```

    password password-string;
    username-include {
        circuit-type;
        delimiter delimiter-character;
        domain-name domain-name-string;
        interface-name;
        mac-address;
        option-82 ( circuit-id | remote-id);
        radius-realm radius-realm-string;
        user-prefix user-prefix-string;
    }
}
dynamic-profile profile-name {
    accept (any | dhcp-v4 | dhcp-v6 | inet | inet6);
    ranges (any | low-tag-high-tag), (any | low-tag-high-tag);
}
}
vlan-ranges {
    access-profile profile-name;
    authentication {
        password password-string;
        username-include {
            circuit-type;
            delimiter delimiter-character;
            domain-name domain-name-string;
            interface-name;
            mac-address;
            option-82;
            radius-realm radius-realm-string;
            user-prefix user-prefix-string;
        }
    }
}
dynamic-profile profile-name {
    accept (any | dhcp-v4 | dhcp-v6 | inet | inet6);
    ranges (any | low-tag)—(any | high-tag);
}
}
override tag vlan-tag dynamic-profile profile name;
}
encapsulation (ethernet-bridge | ethernet-vpls | extended-vlan-bridge |
    extended-vlan-vpls | flexible-ethernet-services | vlan-vpls);
ether-options {
    802.3ad {
        aex;
        (backup | primary);
        lacp {
            force-up;
            port-priority
        }
    }
}
asynchronous-notification;
(auto-negotiation | no-auto-negotiation);
ethernet-switch-profile {
    ethernet-policer-profile {
        input-priority-map {
            ieee802.1p premium [ values ];

```



```

    }
    output-priority-map {
        classifier {
            premium {
                forwarding-class class-name {
                    loss-priority (high | low);
                }
            }
        }
    }
    policer cos-policer-name {
        aggregate {
            bandwidth-limit bps;
            burst-size-limit bytes;
        }
        premium {
            bandwidth-limit bps;
            burst-size-limit bytes;
        }
    }
    tag-protocol-id;
}
(mac-learn-enable | no-mac-learn-enable);
}
(flow-control | no-flow-control);
ignore-l3-incompletes;
link-mode (automatic | full-duplex | half-duplex);
(lloopback | no-loopback);
keepalives <interval seconds> <down-count number> <up-count number>;
speed (1g | 10m | 100m | 10m-100m | auto-negotiation);
source-address-filter {
    mac-address;
}
source-filtering | no-source-filtering;
}
flexible-vlan-tagging;
(gratuitous-arp-reply | no-gratuitous-arp-reply);
hold-time (up milliseconds | down milliseconds);
interface-transmit-statistics;
(keepalives <down-count number> <interval seconds> <up-count number> |
no-keepalives);
layer2-policer {
    apply-groups [ group-names ];
    apply-groups-except [ group-names ];
}
link-mode (automatic | full-duplex);
mac mac-address;
mtu bytes;
multi-chassis-protection peer-ip-address {
    interface interface-name;
}
native-vlan-id number;
no-gratuitous-arp-request;
optics-options {
    alarm low-light-alarm {
        (link-down | syslog);
    }
}

```

```

    }
    warning low-light-warning {
        (link-down | syslog);
    }
    wavelength nm;
}
passive-monitor-mode;
per-unit-scheduler;
speed (10m | 100m | 1g | auto | oc3 | oc12 | oc48);
stacked-vlan-tagging;
traceoptions {
    flag flag;
}
transmit-bucket {
    overflow discard;
    rate percentage;
    threshold bytes;
}
(traps | no-traps);
unidirectional;
vlan-tagging;
}

```

```

interface-name {
    unit logical-unit-number {
        disable;
        accept-source-mac {
            mac-address mac-address {
                policer {
                    input policer-name;
                    output policer-name;
                }
            }
        }
        accounting-profile name;
        advisory-options {
            downstream-rate rate;
            upstream-rate rate;
        }
        arp-resp (restricted|unrestricted);
        bandwidth rate;
        clear-dont-fragment-bit;
        copy-tos-to-outer-ip-header;
        demux-destination family;
        encapsulation (vlan-bridge | vlan-vpls);
        epd-threshold cells plp1 cells;
        filter filter-name;
        inner-vlan-id-range start start-id end end-id;
        input-vlan-map {
            (pop | pop-pop | pop-swap | push | push-push | swap | swap-push | swap-swap);
            inner-tag-protocol-id tpid;
            inner-vlan-id number;
            tag-protocol-id tpid;
            vlan-id number;
        }
    }
}

```

```

interface-shared-with psdnumerical-index;
layer2-policer {
    input-hierarchical-policer policer-name;
    input-policer policer-name;
    input-three-color policer-name;
    output-policer policer-name;
    output-three-color policer-name;
}
multi-chassis-protection peer-ip-address {
    interface interface-name;
}
native-inner-vlan-id number;
output-vlan-map {
    (pop | pop-pop | pop-swap | push | push-push | swap | swap-push | swap-swap);
    inner-tag-protocol-id tpid;
    inner-vlan-id number;
    tag-protocol-id tpid;
    vlan-id number;
}
peer-interface interface-name;
peer-unit unit-number;
plp-to-clp;
proxy-arp <restricted | unrestricted>;
rpm {
    (client | server);
    twamp-server;
}
swap-by-poppush;
vlan-id number;
vlan-id-list [ vlan-id vlan-id-vlan-id ];
vlan-id-range number-number;
vlan-tags (inner <tpid.>vlan-id | inner-list [ vlan-id vlan-id-vlan-id ] |
    inner-range <tpid.>vlan-id-vlan-id) outer <tpid.>vlan-id;
}

unit logical-unit-number {
    family ethernet-switching {
        filter {
            group filter-group-number;
            (input filter-name | input-list [ filter-names ]);
            (output filter-name | output-list [ filter-names ]);
            (inner-vlan-id-list [ vlan-ids ] | vlan-id number | vlan-id-list [ number
                number-number ]);
            interface-mode (access | trunk);
            policer {
                input policer-name;
                output policer-name;
            }
            vlan-rewrite {
                translate old-vlan-id new-vlan-id;
            }
            vlan {
                members [ all vlan-identifiers ];
            }
        }
    }
    family inet {

```

```

filter {
    group filter-group-number;
    (input filter-name | input-list [ filter-names ]);
    (output filter-name | output-list [ filter-names ]);
}
input-hierarchical-policer policer-name;
mac-validate (loose | strict);
mtu bytes;
no-neighbor-learn;
no-redirects;
policer {
    arp policer-template-name;
    input policer-name;
    output policer-name;
}
primary;
receive-options-packets;
receive-ttl-exceeded;
rpf-check {
    fail-filter filter-name;
    mode loose;
}
sampling {
    (input | output | input output);
}
simple-filter {
    input filter-name;
}
targeted-broadcast {
    forward-and-send-to-re;
    forward-only;
}
unnumbered-address interface-name <destination address>
    <destination-profile profile-name> <preferred-source-address address>;
}

family inet6 {
    address ipv6-address {
        destination destination-address;
        eui-64;
        ndp ipv6-address <l2-interface interface-name> <(mac mac-address |
            multicast-mac multicast-mac-address) <publish>>;
        preferred;
        primary;
        vrrp-inet6-group group-number {
            (accept-data | no-accept-data);
            fast-interval milliseconds;
            inet6-advertise-interval seconds;
            (no-preempt; | ... the following preempt statement ...)
            preempt {
                hold-time seconds;
            }
            priority number;
            track {
                interface interface-name {

```

```
        bandwidth-threshold bits-per-second priority-cost priority;  
        priority-cost priority;  
    }  
    priority-hold-time seconds;  
    route ip-address-prefix/prefix-length routing-instance instance-name  
        priority-cost priority;  
    }  
    virtual-inet6-address [ addresses ];  
    virtual-link-local-address ipv6-address;  
    vrrp-inherit-from {  
        active-group group-number;  
        active-interface interface-name;  
    }  
    }  
    }  
    (dad-disable | no-dad-disable);  
    filter {  
        group filter-group-number;  
        (input filter-name | input-list [ filter-names ] );  
        (output filter-name | output-list [ filter-names ] );  
    }  
    input-hierarchical-policer policer-name;  
    mtu bytes;  
    nd6-stale-time seconds;  
    no-neighbor-learn;  
    policer {  
        input policer-name;  
        output policer-name;  
    }  
    rpf-check {  
        fail-filter filter-name;  
        mode loose;  
    }  
    sampling {  
        (input | output | input output);  
    }  
    unnumbered-address interface-name preferred-source-address address;  
    }  
  
    family iso {  
        address iso-address;  
        mtu bytes;  
    }  
  
    family mlfr-end-to-end {  
        bundle logical-interface-name;  
    }  
  
    family mpls {  
        filter {  
            group filter-group-number;  
            (input filter-name | input-list [ filter-names ] );  
            (output filter-name | output-list [ filter-names ] );  
        }  
    }  
}
```

```

    }
    input-hierarchical-policer policer-name;
    maximum-labels maximum-labels;
    mtu bytes;
    policer {
        input policer-name;
        output policer-name;
    }
}

family vpls {
    core-facing;
    filter {
        group filter-group-number;
        (input filter-name | input-list [ filter-names ]);
        (output filter-name | output-list [ filter-names ]);
    }
    policer {
        input policer-name;
        output policer-name;
    }
}
}
}
}

```

Related Documentation • *Notational Conventions Used in Junos OS Configuration Hierarchies*

schedulers (Interfaces)

Syntax	<code>schedulers <i>number</i>;</code>
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	<i>number</i> —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	• <i>Configuring the Number of Schedulers for Ethernet IQ2 PICs</i>