



---

Junos<sup>®</sup> OS

## Class of Service Overview

Release

13.1



---

Published: 2013-02-11

Juniper Networks, Inc.  
1194 North Mathilda Avenue  
Sunnyvale, California 94089  
USA  
408-745-2000  
[www.juniper.net](http://www.juniper.net)

This product includes the Envoy SNMP Engine, developed by Epilogue Technology, an Integrated Systems Company. Copyright © 1986-1997, Epilogue Technology Corporation. All rights reserved. This program and its documentation were developed at private expense, and no part of them is in the public domain.

This product includes memory allocation software developed by Mark Moraes, copyright © 1988, 1989, 1993, University of Toronto.

This product includes FreeBSD software developed by the University of California, Berkeley, and its contributors. All of the documentation and software included in the 4.4BSD and 4.4BSD-Lite Releases is copyrighted by the Regents of the University of California. Copyright © 1979, 1980, 1983, 1986, 1988, 1989, 1991, 1992, 1993, 1994. The Regents of the University of California. All rights reserved.

GateD software copyright © 1995, the Regents of the University. All rights reserved. Gate Daemon was originated and developed through release 3.0 by Cornell University and its collaborators. Gated is based on Kirton's EGP, UC Berkeley's routing daemon (routed), and DCN's HELLO routing protocol. Development of Gated has been supported in part by the National Science Foundation. Portions of the GateD software copyright © 1988, Regents of the University of California. All rights reserved. Portions of the GateD software copyright © 1991, D. L. S. Associates.

This product includes software developed by Maker Communications, Inc., copyright © 1996, 1997, Maker Communications, Inc.

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.

Products made or sold by Juniper Networks or components thereof might be covered by one or more of the following patents that are owned by or licensed to Juniper Networks: U.S. Patent Nos. 5,473,599, 5,905,725, 5,909,440, 6,192,051, 6,333,650, 6,359,479, 6,406,312, 6,429,706, 6,459,579, 6,493,347, 6,538,518, 6,538,899, 6,552,918, 6,567,902, 6,578,186, and 6,590,785.

#### *Junos® OS Class of Service Overview*

13.1

Copyright © 2013, 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       |
| <b>Part 1</b>    | <b>Overview</b>  |           |
| <b>Chapter 1</b> | <b>CoS Overview . . . . .</b>  | <b>3</b>  |
|                  | CoS Overview . . . . .   | 3         |
|                  | CoS Standards . . . . .  | 4         |
|                  | Understanding Packet Flow Across a Network . . . . .                         | 4         |
|                  | Junos CoS Components . . . . .   | 5         |
|                  | Default CoS Settings . . . . .   | 7         |
|                  | CoS Applications Overview . . . . .  | 9         |
|                  | Interface Types That Do Not Support CoS . . . . .                            | 10        |
|                  | VPLS and Default CoS Classification . . . . .                                | 11        |
| <b>Chapter 2</b> | <b>CoS Input and Output Configuration . . . . .</b>                          | <b>13</b> |
|                  | CoS Inputs and Outputs Overview . . . . .                                    | 13        |
| <b>Chapter 3</b> | <b>Packet Flow Within Routers . . . . .</b>                                  | <b>15</b> |
|                  | Packet Flow Within Routers Overview . . . . .                                | 15        |
|                  | Packet Flow on Juniper Networks J Series Services Routers . . . . .          | 16        |
|                  | Packet Flow on Juniper Networks M Series Multiservice Edge Routers . . . . . | 16        |
|                  | Incoming I/O Manager ASIC . . . . .  | 17        |
|                  | Internet Processor ASIC . . . . .  | 17        |
|                  | Outgoing I/O Manager ASIC . . . . .  | 17        |
|                  | Enhanced CFEB and CoS on M7i and M10i Multiservice Edge Routers . . . . .    | 17        |
|                  | Packet Flow on MX Series 3D Universal Edge Routers . . . . .                 | 18        |
|                  | Packet Flow on Juniper Networks T Series Core Routers . . . . .              | 20        |
|                  | Incoming Switch Interface ASICs . . . . .                                    | 21        |
|                  | T Series Routers Internet Processor ASIC . . . . .                           | 21        |
|                  | Queuing and Memory Interface ASICs . . . . .                                 | 22        |
|                  | Outgoing Switch Interface ASICs . . . . .                                    | 22        |

|                  |   |           |
|------------------|---|-----------|
| <b>Chapter 4</b> | <b>Packet Flow Through the CoS Process . . . . .</b>                        | <b>23</b> |
|                  | Packet Flow Through the CoS Process Overview . . . . .                      | 23        |
| <b>Part 2</b>    | <b>Configuration</b>  |           |
| <b>Chapter 5</b> | <b>Examples . . . . .</b>   | <b>29</b> |
|                  | CoS Inputs and Outputs Examples . . . . .                                   | 29        |
|                  | Example of Packet Flow on MX Series 3D Universal Edge Routers . . . . .     | 30        |
|                  | Packet Flow Through the CoS Process Configuration Example . . . . .         | 31        |
|                  | Example: Configuring Classifiers, Rewrite Markers, and Schedulers . . . . . | 33        |
|                  | Example: Configuring a CoS Policy for IPv6 Packets . . . . .                | 38        |
| <b>Chapter 6</b> | <b>Configuration Statement Hierarchies . . . . .</b>                        | <b>39</b> |
|                  | [edit chassis] Hierarchy Level . . . . .                                    | 39        |
|                  | [edit class-of-service] Hierarchy Level . . . . .                           | 40        |
|                  | [edit firewall] Hierarchy Level . . . . .                                   | 44        |
|                  | [edit interfaces] Hierarchy Level . . . . .                                 | 45        |
|                  | [edit services cos] Hierarchy Level . . . . .                               | 47        |
| <b>Part 3</b>    | <b>Index</b>  |           |
|                  | Index . . . . .   | 51        |

# List of Figures

|                  |   |           |
|------------------|---|-----------|
| <b>Part 1</b>    | <b>Overview</b>   |           |
| <b>Chapter 1</b> | <b>CoS Overview</b> .....   | <b>3</b>  |
|                  | Figure 1: Packet Flow Across the Network .....  | 5         |
| <b>Chapter 3</b> | <b>Packet Flow Within Routers</b> .....   | <b>15</b> |
|                  | Figure 2: M Series Routers Packet Forwarding Engine Components and Data<br>Flow ..... | 16        |
|                  | Figure 3: MX Series Router Packet Forwarding and Data Flow .....                      | 18        |
|                  | Figure 4: Packet Handling on the M Series and T Series Routers .....                  | 20        |
|                  | Figure 5: Packet Handling on the MX Series Routers .....                              | 20        |
|                  | Figure 6: T Series Router Packet Forwarding Engine Components and Data<br>Flow .....  | 21        |
| <b>Chapter 4</b> | <b>Packet Flow Through the CoS Process</b> .....                                      | <b>23</b> |
|                  | Figure 7: CoS Classifier, Queues, and Scheduler .....                                 | 24        |
|                  | Figure 8: Packet Flow Through CoS Configurable Components .....                       | 24        |



# List of Tables

|                  |   |           |
|------------------|---|-----------|
|                  | <b>About the Documentation . . . . .</b>            | <b>ix</b> |
|                  | Table 1: Notice Icons . . . . .                     | xi        |
|                  | Table 2: Text and Syntax Conventions . . . . .      | xi        |
| <b>Part 1</b>    | <b>Overview</b>                                     |           |
| <b>Chapter 1</b> | <b>CoS Overview . . . . .</b>                       | <b>3</b>  |
|                  | Table 3: Default VPLS Classifiers . . . . .         | 12        |
| <b>Chapter 2</b> | <b>CoS Input and Output Configuration . . . . .</b> | <b>13</b> |
|                  | Table 4: CoS Mappings—Inputs and Outputs . . . . .  | 13        |





# 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:

- [M Series](#)
- [MX Series](#)
- [T Series](#)
- [PTX Series](#)
- [J Series](#)

## Using the Examples in This Manual

---

If you want to use the examples in this manual, you can use the **load merge** or the **load merge relative** command. These commands cause the software to merge the incoming

configuration into the current candidate configuration. The example does not become active until you commit the candidate configuration.

If the example configuration contains the top level of the hierarchy (or multiple hierarchies), the example is a *full example*. In this case, use the **load merge** command.

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

## Merging a Full Example

To merge a full example, follow these steps:

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

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

```
system {
  scripts {
    commit {
      file ex-script.xsl;
    }
  }
}
interfaces {
  fxp0 {
    disable;
    unit 0 {
      family inet {
        address 10.0.0.1/24;
      }
    }
  }
}
```

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

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

## Merging a Snippet

To merge a snippet, follow these steps:

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

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

```
commit {
  file ex-script-snippet.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.                     |

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

Table 2: Text and Syntax Conventions

| Convention                 | Description  | Examples   |
|----------------------------|--|--|
| <b>Bold text like this</b> | Represents text that you type.                         | To enter configuration mode, type the <b>configure</b> command:<br><br>user@host> <b>configure</b> |
| Fixed-width text like this | Represents output that appears on the terminal screen. | user@host> <b>show chassis alarms</b><br>No alarms currently active                                |

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

| Convention                     | Description  | Examples  |
|--------------------------------|--|---|
| <i>Italic text like this</i>   | <ul style="list-style-type: none"> <li>Introduces or emphasizes important new terms.</li> <li>Identifies book names.</li> <li>Identifies RFC and Internet draft titles.</li> </ul> | <ul style="list-style-type: none"> <li>A policy <i>term</i> is a named structure that defines match conditions and actions.</li> <li><i>Junos OS System Basics Configuration Guide</i></li> <li>RFC 1997, <i>BGP Communities Attribute</i></li> </ul> |
| <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:<br><br>[edit]<br>root@# <b>set system domain-name</b> <i>domain-name</i>   |
| <b>Text like this</b>          | Represents names of configuration statements, commands, files, and directories; configuration hierarchy levels; or labels on routing platform components.                          | <ul style="list-style-type: none"> <li>To configure a stub area, include the <b>stub</b> statement at the [edit protocols ospf area area-id] hierarchy level.</li> <li>The console port is labeled <b>CONSOLE</b>.</li> </ul>                         |
| < > (angle brackets)           | Enclose optional keywords or variables.  | <b>stub &lt;default-metric metric&gt;;</b>  |
| (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.             | <b>broadcast   multicast</b><br><br><i>(string1   string2   string3)</i>  |
| # (pound sign)                 | Indicates a comment specified on the same line as the configuration statement to which it applies.   | <b>rsvp { # Required for dynamic MPLS only</b>  |
| [ ] (square brackets)          | Enclose a variable for which you can substitute one or more values.  | <b>community name members [ community-ids ]</b>   |
| Indentation and braces ( { } ) | Identify a level in the configuration hierarchy.   | [edit]<br>routing-options {<br>static {<br>route default {<br>nexthop <i>address</i> ;<br>retain;<br>}<br>}<br>}  |
| ;(semicolon)                   | Identifies a leaf statement at a configuration hierarchy level.  |   |
| <b>J-Web GUI Conventions</b>   |  |   |
| <b>Bold text like this</b>     | Represents J-Web graphical user interface (GUI) items you click or select.   | <ul style="list-style-type: none"> <li>In the Logical Interfaces box, select <b>All Interfaces</b>.</li> <li>To cancel the configuration, click <b>Cancel</b>.</li> </ul>   |
| > (bold right angle bracket)   | Separates levels in a hierarchy of J-Web selections.   | In the configuration editor hierarchy, select <b>Protocols&gt;Ospf</b> .  |

## Documentation Feedback

---

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

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

## Requesting Technical Support

---

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

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

## Self-Help Online Tools and Resources

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

- Find CSC offerings: <http://www.juniper.net/customers/support/>
- Search for known bugs: <http://www2.juniper.net/kb/>
- Find product documentation: <http://www.juniper.net/techpubs/>
- Find solutions and answer questions using our Knowledge Base: <http://kb.juniper.net/>
- Download the latest versions of software and review release notes: <http://www.juniper.net/customers/csc/software/>
- Search technical bulletins for relevant hardware and software notifications: <https://www.juniper.net/alerts/>

- Join and participate in the Juniper Networks Community Forum:  
<http://www.juniper.net/company/communities/>
- Open a case online in the CSC Case Management tool: <http://www.juniper.net/cm/>

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

## Opening a Case with JTAC

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

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

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

## PART 1

# Overview

- [CoS Overview on page 3](#)
- [CoS Input and Output Configuration on page 13](#)
- [Packet Flow Within Routers on page 15](#)
- [Packet Flow Through the CoS Process on page 23](#)





## CHAPTER 1

# CoS Overview

- [CoS Overview on page 3](#)
- [CoS Standards on page 4](#)
- [Understanding Packet Flow Across a Network on page 4](#)
- [Junos CoS Components on page 5](#)
- [Default CoS Settings on page 7](#)
- [CoS Applications Overview on page 9](#)
- [Interface Types That Do Not Support CoS on page 10](#)
- [VPLS and Default CoS Classification on page 11](#)

## CoS Overview

---

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

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

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

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

**Related  
Documentation**

- Hardware Capabilities and Limitations

## CoS Standards

---

The standards for Juniper Networks® Junos® operating system (Junos OS) class of service (CoS) capabilities are defined in the following RFCs:

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

## Understanding Packet Flow Across a Network

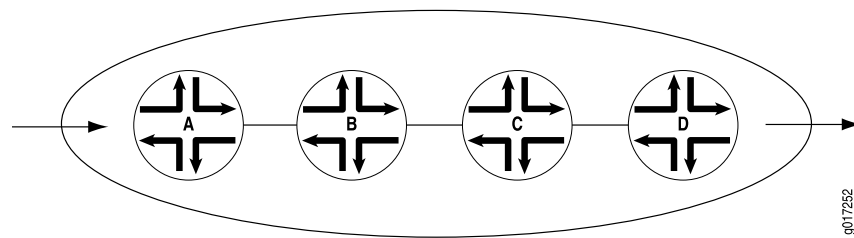
---

CoS works by examining traffic entering at the edge of your network. The edge routers classify traffic into defined service groups, to provide the special treatment of traffic across the network. For example, voice traffic can be sent across certain links, and data traffic can use other links. In addition, the data traffic streams can be serviced differently along the network path to ensure that higher-paying customers receive better service. As the traffic leaves the network at the far edge, you can reclassify the traffic.

To support CoS, you must configure each router in the network. Generally, each router examines the packets that enter it to determine their CoS settings. These settings then dictate which packets are first transmitted to the next downstream router. In addition, the routers at the edges of the network might be required to alter the CoS settings of the packets that enter the network from the customer or peer networks.

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

Figure 1: Packet Flow Across the Network



## Junos CoS Components

The Juniper Networks® Junos® operating system (Junos OS) CoS consists of many components that you can combine and tune to provide the level of services required by customers.

The Junos OS CoS components include:

- **Code-point aliases**—A *code-point alias* assigns a name to a pattern of code-point bits. You can use this name instead of the bit pattern when you configure other CoS components, such as classifiers, drop-profile maps, and rewrite rules.
- **Classifiers**—*Packet classification* refers to the examination of an incoming packet. This function associates the packet with a particular CoS servicing level. In the Junos OS, classifiers associate incoming packets with a forwarding class and loss priority and, based on the associated forwarding class, assign packets to output queues. Two general types of classifiers are supported:
  - **Behavior aggregate or CoS value traffic classifiers**—A *behavior aggregate* (BA) is a method of classification that operates on a packet as it enters the router. The CoS value in the packet header is examined, and this single field determines the CoS settings applied to the packet. BA classifiers allow you to set the forwarding class and loss priority of a packet based on the Differentiated Services code point (DSCP) value, DSCP IPv6 value, IP precedence value, MPLS EXP bits, and IEEE 802.1p value. The default classifier is based on the IP precedence value.
  - **Multifield traffic classifiers**—A *multifield* classifier is a second method for classifying traffic flows. Unlike a behavior aggregate, a multifield classifier can examine multiple fields in the packet. Examples of some fields that a multifield classifier can examine include the source and destination address of the packet as well as the source and destination port numbers of the packet. With multifield classifiers, you set the forwarding class and loss priority of a packet based on firewall filter rules.
- **Forwarding classes**—The *forwarding classes* affect the forwarding, scheduling, and marking policies applied to packets as they transit a router. The forwarding class plus the loss priority define the per-hop behavior. Four categories of forwarding classes are supported: best effort, assured forwarding, expedited forwarding, and network control. For Juniper Networks M Series Multiservice Edge Routers, four forwarding classes are supported. You can configure up to one each of the four types of forwarding classes. For M120 and M320 Multiservice Edge Routers, Juniper Networks MX Series 3D Universal Edge Routers, and Juniper Networks T Series Core Routers, 16 forwarding classes are

supported, so you can classify packets more granularly. For example, you can configure multiple classes of expedited forwarding (EF) traffic: EF, EF1, and EF2.

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

## Default CoS Settings

---

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

You can display default CoS settings by issuing the **show class-of-service** operational mode command. This section includes sample output displaying the default CoS settings. The sample output is truncated for brevity.

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

#### Default Forwarding Classes

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

#### Default Code-Point Aliases

Code point type: dscp

| Alias | Bit pattern |
|-------|-------------|
| af11  | 001010      |
| af12  | 001100      |

...

Code point type: dscp-ipv6

...

Code point type: exp

...

Code point type: ieee-802.1

...

Code point type: inet-precedence

...

#### Default Classifiers

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

...

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

...

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

...

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

...

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

...

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

...

#### Default Frame Relay Loss Priority Map

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

| Code point | Loss priority |
|------------|---------------|
| 0          | low           |
| 1          | high          |

#### Default Rewrite Rules

Rewrite rule: dscp-default, Code point type: dscp, Index: 24

| Forwarding class | Loss priority | Code point |
|------------------|---------------|------------|
| best-effort      | low           | 000000     |
| best-effort      | high          | 000000     |

...

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

...

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

...

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

...

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

...

**Default Drop Profile**

Drop profile: &lt;default-drop-profile&gt;, Type: discrete, Index: 1

| Fill level | Drop probability |
|------------|------------------|
| 100        | 100              |

**Default Schedulers**

Scheduler map: &lt;default&gt;, Index: 2

Scheduler: &lt;default-be&gt;, Forwarding class: best-effort, Index: 17

Transmit rate: 95 percent, Rate Limit: none, Buffer size: 95 percent, Priority: low

Drop profiles:

| Loss priority | Protocol | Index | Name                   |
|---------------|----------|-------|------------------------|
| Low           | Any      | 1     | <default-drop-profile> |
| High          | Any      | 1     | <default-drop-profile> |

...

**Related Documentation**

- Default Forwarding Classes
- Default Behavior Aggregate Classification Overview
- Default Drop Profile
- Default Schedulers Overview
- Forwarding Classes and Fabric Priority Queues

**CoS Applications Overview**

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

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

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

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

The Junos OS CoS applications support the following range of mechanisms:

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

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

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

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

---

## Interface Types That Do Not Support CoS

---

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

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



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

---

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

- **cau4**—Channelized STM1 IQ interface (configured on the Channelized STM1 IQ PIC).
- **coc1**—Channelized OC1 IQ interface (configured on the Channelized OC12 IQ PIC).
- **coc12**—Channelized OC12 IQ interface (configured on the Channelized OC12 IQ PIC).
- **cstm-1**—Channelized STM1 IQ interface (configured on the Channelized STM1 IQ PIC).



- **ct1**—Channelized T1 IQ interface (configured on the Channelized DS3 IQ PIC or Channelized OC12 IQ PIC).
- **ct3**—Channelized T3 IQ interface (configured on the Channelized DS3 IQ PIC or Channelized OC12 IQ PIC).
- **ce1**—Channelized E1 IQ interface (configured on the Channelized E1 IQ PIC or Channelized STM1 IQ PIC).
- **dsc**—Discard interface.
- **fxp**—Management and internal Ethernet interfaces.
- **lo**—Loopback interface. This interface is internally generated.
- **pe**—Encapsulates packets destined for the rendezvous point router. This interface is present on the first-hop router.
- **pd**—De-encapsulates packets at the rendezvous point. This interface is present on the rendezvous point.
- **vt**—Virtual loopback tunnel interface.



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

**Related Documentation**

- CoS on ATM Interfaces Overview

## VPLS and Default CoS Classification

A VPLS routing instance with the **no-tunnel-services** option configured has a default classifier applied to the label-switched interface for all VPLS packets coming from the remote VPLS PE. This default classifier is modifiable only on MX Series routers. On T Series, when **no-tunnel-services** option is configured, the custom classifier for VPLS instances is not supported.



**NOTE:** With **no-tunnel-services** configured, custom classifier for VPLS routing instances on T Series and LMNR based FPC for M320 is not supported. When a wild card configuration or an explicit routing instances are configured for VPLS on CoS CLI, the custom classifier binding results in default classifier binding on Packet Forwarding Engine (PFE).

For example, on routers with eight queues (Juniper Networks M120 and M320 Multiservice Edge Routers, MX Series 3D Universal Edge Routers, and T Series Core Routers), the default classification applied to **no-tunnel-services** VPLS packets are shown in [Table 3 on page 12](#).

Table 3: Default VPLS Classifiers

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



**NOTE:** Forwarding class to queue number mapping is not always one-to-one. Forwarding classes and queues are only the same when default forwarding-class-to-queue mapping is in effect. For more information about configuring forwarding class and queues, see [Configuring Forwarding Classes](#).

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



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

CHAPTER 2

# CoS Input and Output Configuration

- [CoS Inputs and Outputs Overview on page 13](#)

## CoS Inputs and Outputs Overview

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

Some CoS components map one set of values to another set of values. Each mapping contains one or more inputs and one or more outputs. When you configure a mapping, you set the outputs for a given set of inputs, as shown in [Table 4 on page 13](#).

Table 4: CoS Mappings—Inputs and Outputs

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

- Related Documentation
- Default Behavior Aggregate Classification Overview
  - Configuring Drop Profile Maps for Schedulers
  - Applying Default Rewrite Rules
  - [CoS Inputs and Outputs Examples on page 29](#)



## CHAPTER 3

# Packet Flow Within Routers

- [Packet Flow Within Routers Overview on page 15](#)
- [Packet Flow on Juniper Networks J Series Services Routers on page 16](#)
- [Packet Flow on Juniper Networks M Series Multiservice Edge Routers on page 16](#)
- [Packet Flow on MX Series 3D Universal Edge Routers on page 18](#)
- [Packet Flow on Juniper Networks T Series Core Routers on page 20](#)

### Packet Flow Within Routers Overview

---

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

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

#### Related Documentation

- [Packet Flow on Juniper Networks J Series Services Routers on page 16](#)
- [Packet Flow on Juniper Networks M Series Multiservice Edge Routers on page 16](#)
- [Packet Flow on MX Series 3D Universal Edge Routers on page 18](#)
- [Packet Flow on Juniper Networks T Series Core Routers on page 20](#)

## Packet Flow on Juniper Networks J Series Services Routers

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

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

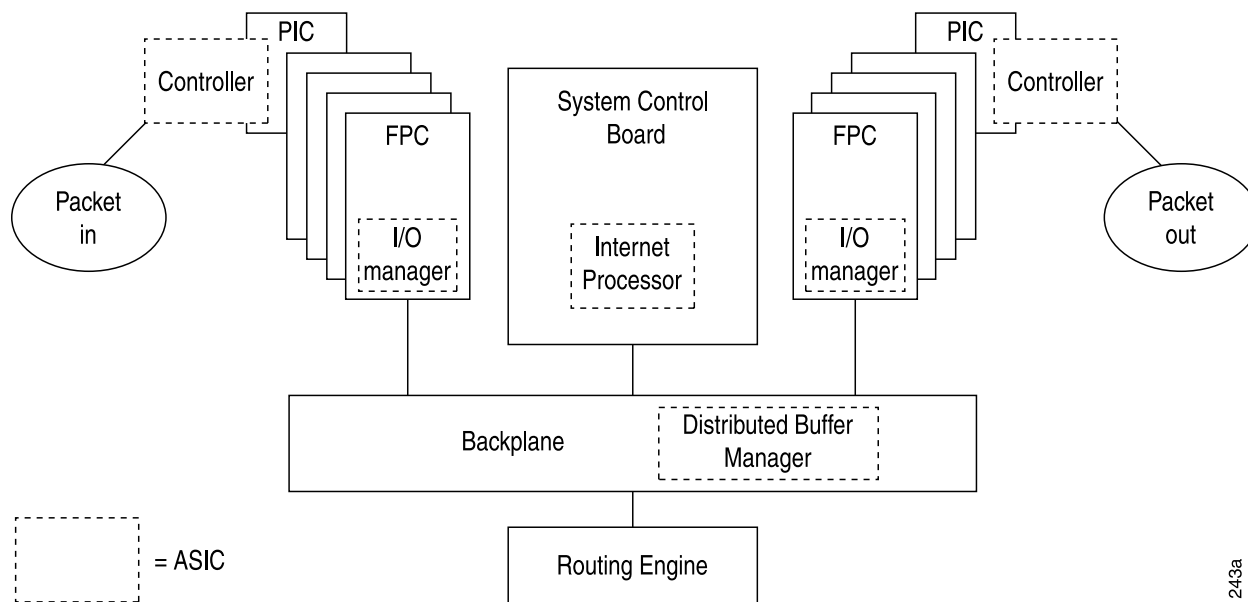
### Related Documentation

- [Packet Flow Within Routers Overview on page 15](#)
- [Packet Flow on Juniper Networks M Series Multiservice Edge Routers on page 16](#)
- [Packet Flow on MX Series 3D Universal Edge Routers on page 18](#)
- [Packet Flow on Juniper Networks T Series Core Routers on page 20](#)

## Packet Flow on Juniper Networks M Series Multiservice Edge Routers

On M Series Multiservice Edge Routers, CoS actions are performed in several locations in a Juniper Networks router: the incoming I/O Manager ASIC, the Internet Processor II ASIC, and the outgoing I/O Manager ASIC. These locations are shown in [Figure 2 on page 16](#).

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



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

- [Incoming I/O Manager ASIC on page 17](#)
- [Internet Processor ASIC on page 17](#)

- [Outgoing I/O Manager ASIC on page 17](#)
- [Enhanced CFEB and CoS on M7i and M10i Multiservice Edge Routers on page 17](#)

## Incoming I/O Manager ASIC

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

## Internet Processor ASIC

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

## Outgoing I/O Manager ASIC

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

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

## Enhanced CFEB and CoS on M7i and M10i Multiservice Edge Routers

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

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

- Support for 16 forwarding classes and 8 queues

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

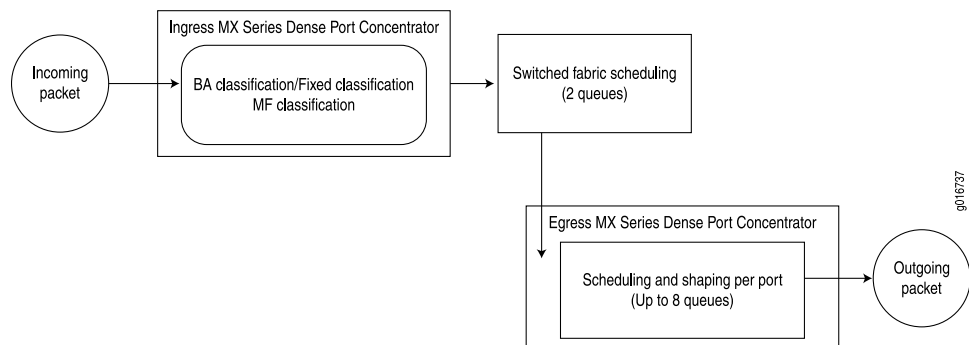
**Related Documentation**

- [Packet Flow Within Routers Overview on page 15](#)
- [Packet Flow on Juniper Networks J Series Services Routers on page 16](#)
- [Packet Flow on MX Series 3D Universal Edge Routers on page 18](#)
- [Packet Flow on Juniper Networks T Series Core Routers on page 20](#)

## Packet Flow on MX Series 3D Universal Edge Routers

The CoS architecture for MX Series 3D Universal Edge Routers, such as the MX960 router, is in concept similar to, but in particulars different from, other routers. The general architecture for MX Series routers is shown in [Figure 3 on page 18](#). [Figure 3 on page 18](#) illustrates packet flow through a Dense Port Concentrator (DPC).

**Figure 3: MX Series Router Packet Forwarding and Data Flow**



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

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





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

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

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

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

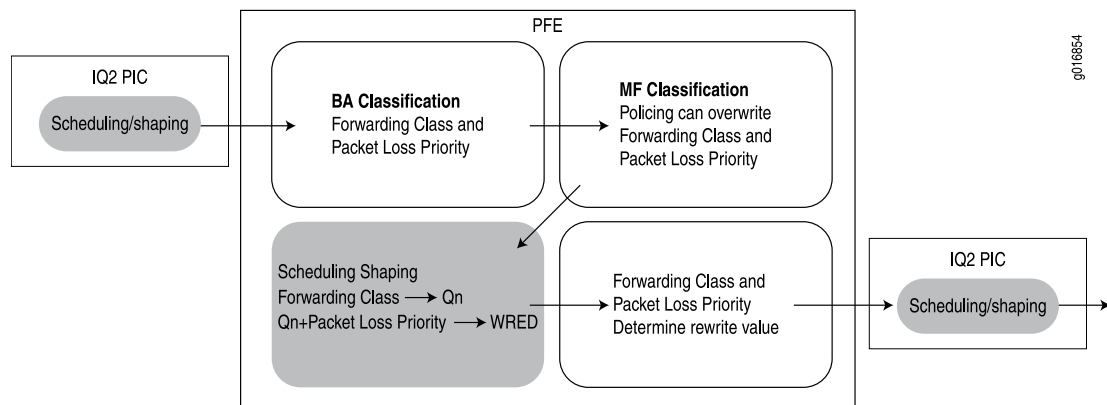
Egress port scheduling supports up to eight queues per port using a form of round-robin queue servicing. The supported priority levels are strict-high, high, medium-high, medium-low, and low. The MX Series router architecture supports both early discard and tail drop on the queues.

All CoS features are supported at line rate.

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

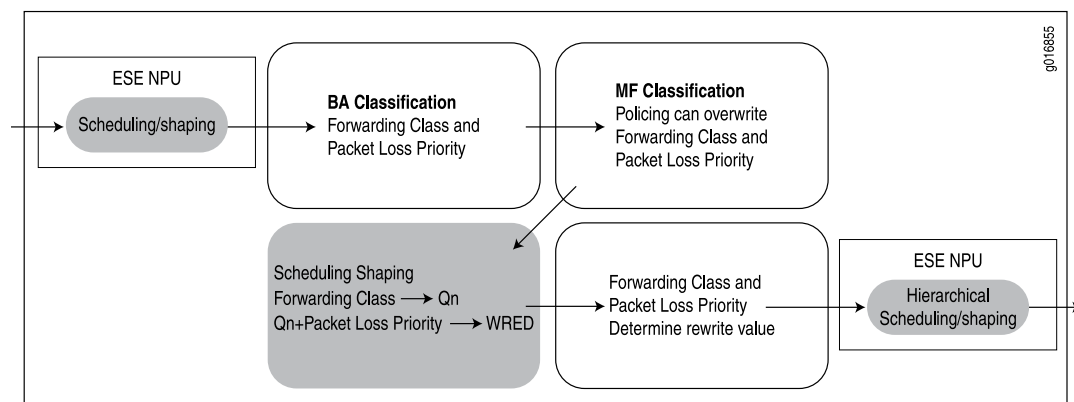
The way that a packet makes its way through an M Series or T Series router with Intelligent Queuing 2 (IQ2) PICs is shown in [Figure 4 on page 20](#). Note that the per-VLAN scheduling and shaping are done on the PIC whereas all other CoS functions at the port level are performed on the Packet Forwarding Engine.

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



The way that a packet makes its way through an MX Series router is shown in [Figure 5 on page 20](#). Note that the scheduling and shaping are done with an integrated architecture along with all other CoS functions. In particular, scheduling and shaping are done on the Ethernet services engine network processing unit (ESE NPU). Hierarchical scheduling is supported on the output side as well as the input side.

Figure 5: Packet Handling on the MX Series Routers



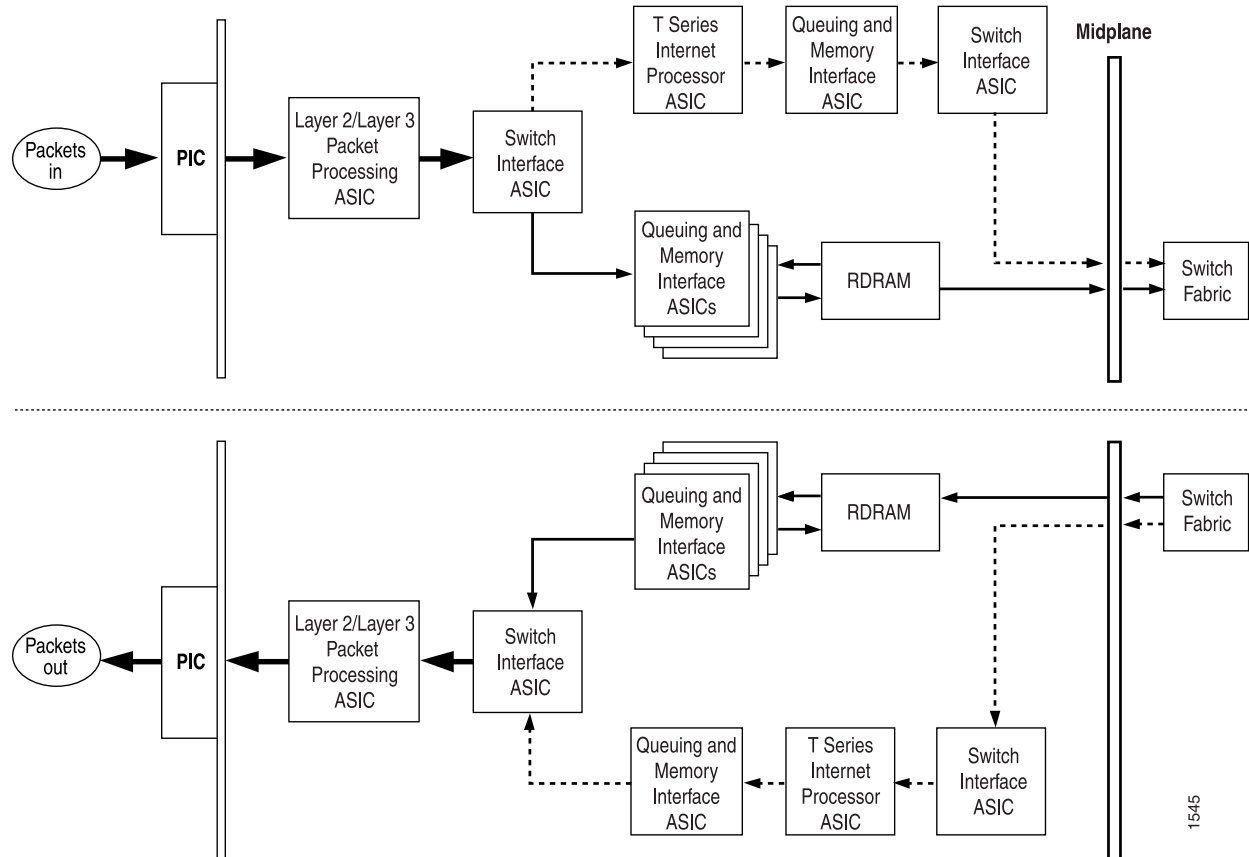
#### Related Documentation

- [Packet Flow Within Routers Overview on page 15](#)
- [Packet Flow on Juniper Networks J Series Services Routers on page 16](#)
- [Packet Flow on Juniper Networks M Series Multiservice Edge Routers on page 16](#)
- [Example of Packet Flow on MX Series 3D Universal Edge Routers on page 30](#)
- [Packet Flow on Juniper Networks T Series Core Routers on page 20](#)

## Packet Flow on Juniper Networks T Series Core Routers

On T Series Core Routers, CoS actions are performed in several locations: the incoming and outgoing Switch Interface ASICs, the T Series router Internet Processor ASIC, and the Queuing and Memory Interface ASICs. These locations are shown in [Figure 6 on page 21](#).

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



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

- [Incoming Switch Interface ASICs on page 21](#)
- [T Series Routers Internet Processor ASIC on page 21](#)
- [Queuing and Memory Interface ASICs on page 22](#)
- [Outgoing Switch Interface ASICs on page 22](#)

## Incoming Switch Interface ASICs

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

## T Series Routers Internet Processor ASIC

The T Series router Internet Processor ASIC receives notification cells representing inbound data packets and performs route lookups in the forwarding table. This lookup determines the outgoing interface on the router and the next-hop IP address for the data packet.

While the packet is being processed by the T Series router Internet Processor ASIC, it might also be evaluated by a firewall filter, which is configured on either the incoming or outgoing interface. This filter can perform the functions of a multifield classifier by matching on multiple elements within the packet and overwriting the forwarding class settings, loss priority settings, or both within the notification cell. Once the route lookup and filter evaluations are complete, the notification cell, now called the result cell, is passed to the Queuing and Memory Interface ASICs.

## Queuing and Memory Interface ASICs

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

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

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

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

## Outgoing Switch Interface ASICs

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

### Related Documentation

- [Packet Flow Within Routers Overview on page 15](#)
- [Packet Flow on Juniper Networks J Series Services Routers on page 16](#)
- [Packet Flow on Juniper Networks M Series Multiservice Edge Routers on page 16](#)
- [Packet Flow on MX Series 3D Universal Edge Routers on page 18](#)

# Packet Flow Through the CoS Process

- [Packet Flow Through the CoS Process Overview on page 23](#)

## Packet Flow Through the CoS Process Overview

---

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

The following steps describe the CoS process:

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



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

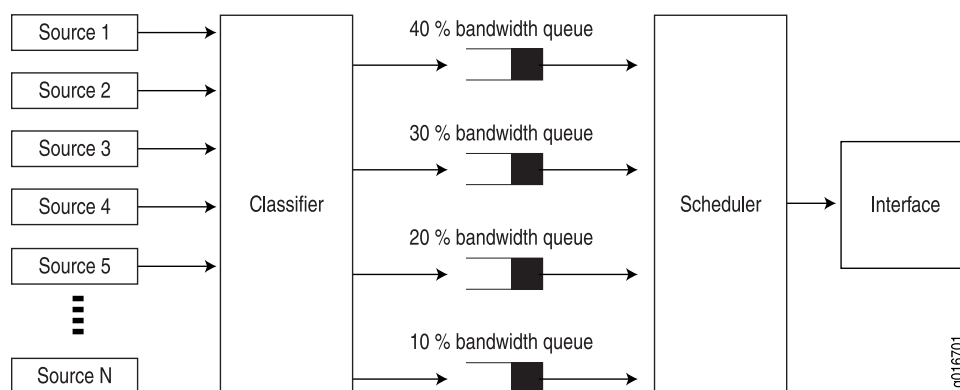
2. The classifier assigns the packet to a forwarding class and a loss priority (at the **[edit class-of-service classifiers]** hierarchy level).
3. Each forwarding class is assigned to a queue (at the **[edit class-of-service forwarding-classes]** hierarchy level).
4. Input (and output) policers meter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

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

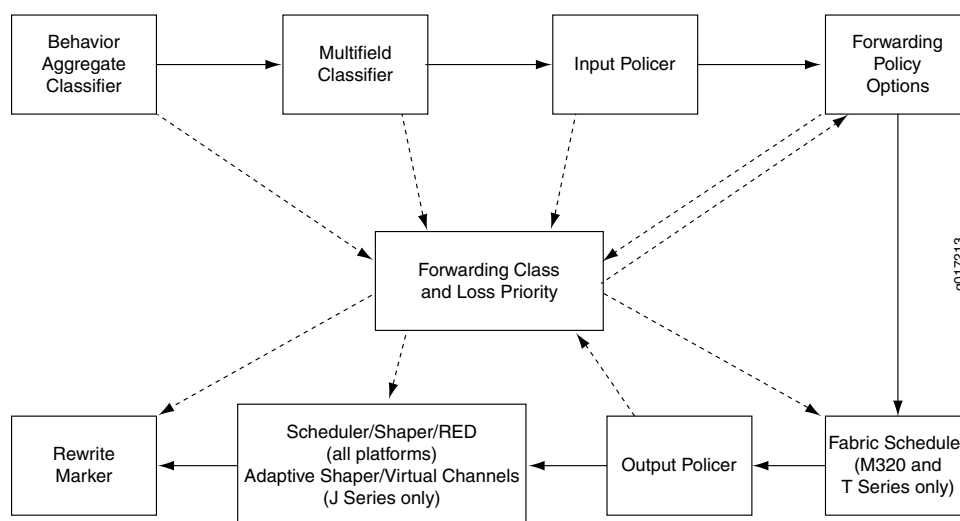
6. The scheduler defines how traffic is treated in the output queue—for example, the transmit rate, buffer size, priority, and drop profile (at the **[edit class-of-service schedulers]** hierarchy level).
7. The scheduler map assigns a scheduler to each forwarding class (at the **[edit class-of-service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

Figure 7 on page 24 and Figure 8 on page 24 show the components of the Junos CoS features, illustrating the sequence in which they interact.

**Figure 7: CoS Classifier, Queues, and Scheduler**



**Figure 8: Packet Flow Through CoS Configurable Components**



Each outer box in [Figure 8 on page 24](#) represents a process component. The components in the upper row apply to inbound packets, and the components in the lower row apply to outbound packets. The arrows with the solid lines point in the direction of packet flow.

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

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

**Related  
Documentation**

- [Packet Flow Through the CoS Process Configuration Example on page 31](#)





## PART 2

# Configuration

- [Examples on page 29](#)
- [Configuration Statement Hierarchies on page 39](#)



## CHAPTER 5

# Examples

- [CoS Inputs and Outputs Examples on page 29](#)
- [Example of Packet Flow on MX Series 3D Universal Edge Routers on page 30](#)
- [Packet Flow Through the CoS Process Configuration Example on page 31](#)
- [Example: Configuring Classifiers, Rewrite Markers, and Schedulers on page 33](#)
- [Example: Configuring a CoS Policy for IPv6 Packets on page 38](#)

### CoS Inputs and Outputs Examples

---

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

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

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

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

```
[edit class-of-service]
schedulers {
  best-effort {
    drop-profile-map loss-priority low protocol any drop-profile low-drop;
    drop-profile-map loss-priority high protocol any drop-profile high-drop;
  }
}
```

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

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

**Related Documentation** • [CoS Inputs and Outputs Overview on page 13](#)

---

## Example of Packet Flow on MX Series 3D Universal Edge Routers

MX Series routers, especially the MX960 3D Universal Edge Router, have several features that differ from the usual CoS features in the Junos OS.

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

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

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

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

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

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

```

        vlan-tag outer;
    }
}

```



**NOTE:** The example above does not apply to single-tagged packets. The following example shows how to configure the classifier on single-tagged interfaces:

```

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

```

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

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

On MX Series routers, the 64 classifier limit is a theoretical upper limit. In practice, you can configure 63 classifiers. Three values are used internally by the default IP precedence, IPv6, and EXP classifiers. Two other classifiers are used for forwarding class and queue operations. This leaves 58 classifiers for configuration purposes. If you configure Differentiated Services code point (DSCP) rewrites for MPLS, the maximum number of classifiers you can configure is less than 58.

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

#### Related Documentation

- [Packet Flow on Juniper Networks M Series Multiservice Edge Routers on page 16](#)
- [Packet Flow on MX Series 3D Universal Edge Routers on page 18](#)

## Packet Flow Through the CoS Process Configuration Example

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

```

[edit class-of-service]
interfaces { # Step 1: Define CoS interfaces.
  so-* {
    scheduler-map sched1;
    unit 0 {
      classifiers {
        exp exp_classifier;
      }
    }
  }
  t3-* {

```

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

```

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

```

**Related Documentation** • [Packet Flow Through the CoS Process Overview on page 23](#)

## Example: Configuring Classifiers, Rewrite Markers, and Schedulers

1. Define a classifier that matches IP traffic arriving on the interface. The affected IP traffic has IP precedence bits with patterns matching those defined by aliases A or B. The loss priority of the matching packets is set to low, and the forwarding class is mapped to best effort (queue 0):

```

[edit]
class-of-service {
    classifiers {
        inet-precedence normal-traffic {
            forwarding-class best-effort {
                loss-priority low code-points [my1 my2];
            }
        }
    }
}

```

Following are the code-point alias and forwarding-class mappings referenced in the **normal-traffic** classifier:

```

[edit]
class-of-service {
    code-point-aliases {
        inet-precedence {
            my1 000;
            my2 001;
            ...
        }
    }
}

[edit]
class-of-service {
    forwarding-classes {
        queue 0 best-effort;
        queue 1 expedited-forwarding;
    }
}

```

2. Use rewrite markers to redefine the bit pattern of outgoing packets. Assign the new bit pattern based on specified forwarding classes, regardless of the loss priority of the packets:

```
[edit]
class-of-service {
  rewrite-rules {
    inet-precedence clear-prec {
      forwarding-class best-effort {
        loss-priority low code-point 000;
        loss-priority high code-point 000;
      }
      forwarding-class expedited-forwarding {
        loss-priority low code-point 100;
        loss-priority high code-point 100;
      }
    }
  }
}
```

3. Configure a scheduler map associating forwarding classes with schedulers and drop-profiles:

```
[edit]
class-of-service {
  scheduler-maps {
    one {
      forwarding-class expedited-forwarding scheduler special;
      forwarding-class best-effort scheduler normal;
    }
  }
}
```

Schedulers establish how to handle the traffic within the output queue for transmission onto the wire. Following is the scheduler referenced in scheduler map **one**:

```
[edit]
class-of-service {
  schedulers {
    special {
      transmit-rate percent 30;
      priority high;
    }
    normal {
      transmit-rate percent 70;
      priority low;
    }
  }
}
```

4. Apply the **normal-traffic** classifier to all SONET/SDH interfaces and all logical interfaces of SONET/SDH interfaces; apply the **clear-prec** rewrite marker to all Gigabit Ethernet interfaces and all logical interfaces of Gigabit Ethernet interfaces; and apply the **one** scheduler map to all interfaces:

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



```

interfaces {
  so-0/0/0 {
    scheduler-map one;
    unit 0 {
      classifiers {
        inet-precedence normal-traffic;
      }
    }
  }
  so-0/0/1 {
    scheduler-map one;
    unit 1 {
      classifiers {
        inet-precedence normal-traffic;
      }
    }
  }
  ge-1/0/0 {
    scheduler-map one;
    unit 0 {
      rewrite-rules {
        inet-precedence clear-prec;
      }
    }
    unit 1 {
      rewrite-rules {
        inet-precedence clear-prec;
      }
    }
  }
  ge-1/0/1 {
    scheduler-map one;
    unit 0 {
      rewrite-rules {
        inet-precedence clear-prec;
      }
    }
    unit 1 {
      rewrite-rules {
        inet-precedence clear-prec;
      }
    }
  }
}

```

Following is the complete configuration:

```

[edit class-of-service]
classifiers {
  inet-precedence normal-traffic {
    forwarding-class best-effort {
      loss-priority low code-points [my1 my2];
    }
  }
}

```

```
code-point-aliases {
  inet-precedence {
    my1 000;
    my2 001;
    cs1 010;
    cs2 011;
    cs3 100;
    cs4 101;
    cs5 110;
    cs6 111;
  }
}
drop-profiles {
  high-priority {
    fill-level 20 drop-probability 100;
  }
  low-priority {
    fill-level 90 drop-probability 95;
  }
  big-queue {
    fill-level 100 drop-probability 100;
  }
}
forwarding-classes {
  queue 0 best-effort;
  queue 1 expedited-forwarding;
}
interfaces {
  so-0/0/0 {
    scheduler-map one;
    unit 0 {
      classifiers {
        inet-precedence normal-traffic;
      }
    }
  }
  so-0/0/1 {
    scheduler-map one;
    unit 1 {
      classifiers {
        inet-precedence normal-traffic;
      }
    }
  }
  ge-1/0/0 {
    scheduler-map one;
    unit 0 {
      rewrite-rules {
        inet-precedence clear-prec;
      }
    }
    unit 1 {
      rewrite-rules {
        inet-precedence clear-prec;
      }
    }
  }
}
```

```
}
ge-1/0/1 {
  scheduler-map one;
  unit 0 {
    rewrite-rules {
      inet-precedence clear-prec;
    }
  }
  unit 1 {
    rewrite-rules {
      inet-precedence clear-prec;
    }
  }
}
rewrite-rules {
  inet-precedence clear-prec {
    forwarding-class best-effort {
      loss-priority low code-point 000;
      loss-priority high code-point 000;
    }
    forwarding-class expedited-forwarding {
      loss-priority low code-point 100;
      loss-priority high code-point 100;
    }
  }
}
scheduler-maps {
  one {
    forwarding-class expedited-forwarding scheduler special;
    forwarding-class best-effort scheduler normal;
  }
}
schedulers {
  special {
    transmit-rate percent 30;
    priority high;
  }
  normal {
    transmit-rate percent 70;
    priority low;
  }
}
```

## Example: Configuring a CoS Policy for IPv6 Packets

---

1. Define a new classifier of type DSCP IPv6.

```
[edit class-of-service]
classifiers {
  dscp-ipv6 core-dscp-map {
    forwarding-class best-effort {
      loss-priority low code-points 000000;
    }
    forwarding-class assured-forwarding {
      loss-priority low code-points 001010;
    }
    forwarding-class network-control {
      loss-priority low code-points 110000;
    }
  }
}
```

2. Define a new rewrite rule of type DSCP IPv6.

```
[edit class-of-service]
rewrite-rules {
  dscp-ipv6 core-dscp-rewrite {
    forwarding-class best-effort {
      loss-priority low code-point 000000;
    }
    forwarding-class assured-forwarding {
      loss-priority low code-point 001010;
    }
    forwarding-class network-control {
      loss-priority low code-point 110000;
    }
  }
}
```

3. Assign the classifier and rewrite rule to a logical interface.

```
[edit class-of-service]
interfaces {
  so-2/0/0 {
    unit 0 {
      classifiers { # Both dscp and dscp-ipv6 classifiers on this interface.
        dscp default;
        dscp-ipv6 core-dscp-map;
      }
      rewrite-rules { # Both dscp and dscp-ipv6 rewrite rules on this interface.
        dscp default;
        dscp-ipv6 core-dscp-rewrite;
      }
    }
  }
}
```

## CHAPTER 6

# Configuration Statement Hierarchies

- [\[edit chassis\] Hierarchy Level on page 39](#)
- [\[edit class-of-service\] Hierarchy Level on page 40](#)
- [\[edit firewall\] Hierarchy Level on page 44](#)
- [\[edit interfaces\] Hierarchy Level on page 45](#)
- [\[edit services cos\] Hierarchy Level on page 47](#)

### [\[edit chassis\] Hierarchy Level](#)

---

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



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

```
[edit chassis]
fpc slot-number {
  pic pic-number {
    max-queues-per-interface (4 | 8);
    q-pic-large-buffer {
      [ large-scale | small-scale ];
    }
    red-buffer-occupancy {
      weighted-averaged [ instant-usage-weight-exponent weight-value ];
    }
    traffic-manager {
      egress-shaping-overhead number;
      ingress-shaping-overhead number;
      mode session-shaping;
    }
  }
}
```

## [edit class-of-service] Hierarchy Level

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

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

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

```

forwarding-class-map forwarding-class-map-name {
  class class-name queue-num queue-number [ restricted-queue queue-number ];
}
forwarding-policy {
  next-hop-map map-name {
    forwarding-class class-name {
      next-hop [ next-hop-name ];
      lsp-next-hop [ lsp-regular-expression ];
      non-lsp-next-hop;
      discard;
    }
  }
  class class-name {
    classification-override {
      forwarding-class class-name;
    }
  }
}
fragmentation-maps {
  map-name {
    forwarding-class class-name {
      drop-timeout milliseconds;
      fragment-threshold bytes;
      multilink-class number;
      no-fragmentation;
    }
  }
}
host-outbound-traffic {
  dscp-code-point value;
  forwarding-class class-name;
  ieee-802.1 {
    default value;
    rewrite-rules;
  }
}
interfaces {
  interface-name {
    classifiers {
      dscp (classifier-name | default);
      ieee-802.1 (classifier-name | default) vlan-tag (inner | outer | classifier-name);
      inet-precedence (classifier-name | default);
    }
    input-scheduler-map map-name;
    input-shaping-rate rate;
    irb {
      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-802.1 (classifier-name | default) vlan-tag (inner | outer | transparent);
        }
      }
    }
  }
}

```

```

rewrite-rules {
    dscp (rewrite-name | default);
    dscp-ipv6 (rewrite-name | default);
    exp (rewrite-name | default) protocol protocol-types;
    ieee-802.1 (rewrite-name | default) vlan-tag (outer | outer-and-inner);
    inet-precedence (rewrite-name | default);
}
}
output-forwarding-class-map forwarding-class-map-name;
member-link-scheduler (replicate | scale);
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 {
        (dscp | dscp-ipv6 | exp | ieee-802.1 | inet-precedence) (classifier-name | default)
        family (mpls | inet);
    }
    forwarding-class class-name;
    fragmentation-map map-name;
    input-scheduler-map map-name;
    input-shaping-rate (percent percentage | rate);
    input-traffic-control-profile profile-name shared-instance instance-name;
    loss-priority-maps {
        frame-relay-de (name | default);
    }
    loss-priority-rewrites {
        frame-relay-de (name | default);
    }
    output-traffic-control-profile profile-name shared-instance instance-name;
    per-session-scheduler;
    rewrite-rules {
        dscp (rewrite-name | default) protocol protocol-types;
        dscp-ipv6 (rewrite-name | default);
        exp (rewrite-name | default) protocol protocol-types;
        exp-push-push-push default;
        exp-swap-push-push default;
        ieee-802.1 (rewrite-name | default) vlan-tag (outer | outer-and-inner);
        inet-precedence (rewrite-name | default) protocol protocol-types;
    }
    scheduler-map map-name;
    shaping-rate rate;
    translation-table (to-dscp-from-dscp | to-dscp-ipv6-from-dscp-ipv6 |
        to-exp-from-exp | to-inet-precedence-from-inet-precedence) table-name;
}
}
}
loss-priority-maps {
    frame-relay-de (Defining Loss Priority Maps) name {

```



```

    loss-priority level code-points [alias | bits ];
  }
}
loss-priority-rewrites {
  frame-relay-de (Defining Loss Priority Maps) name {
    loss-priority level code-point (alias | bits );
  }
}
restricted-queues {
  forwarding-class class-name queue queue-number;
}
rewrite-rules {
  (dscp | dscp-ipv6 | exp | ieee-802.1 | ieee-802.1ad | inet-precedence) rewrite-name {
    import (rewrite-name | default);
    forwarding-class class-name {
      loss-priority level code-point (alias | bits);
    }
  }
}
routing-instances routing-instance-name {
  classifiers {
    exp (classifier-name | default);
    dscp (classifier-name | default);
    dscp-ipv6 (classifier-name | default);
  }
}
scheduler-maps {
  map-name {
    forwarding-class class-name scheduler scheduler-name;
  }
}
schedulers {
  scheduler-name {
    buffer-size (percent percentage | remainder | temporal microseconds);
    drop-profile-map loss-priority (any | low | medium-low | medium-high | high) protocol
      (any | non-tcp | tcp) drop-profile profile-name;
    excess-priority (low | high);
    excess-rate percent percentage;
    excess-rate (percent percentage | proportion value);
    priority priority-level;
    transmit-rate (rate | percent percentage | remainder) <exact | rate-limit>;
  }
}
system-defaults {
  classifiers (classifier-name | exp)
traffic-control-profiles profile-name {
  delay-buffer-rate (percent percentage | rate);
  excess-rate (percent percentage | proportion value);
  guaranteed-rate (percent percentage | rate);
  overhead-accounting (frame-mode | cell-mode) <bytes byte-value>;
  scheduler-map map-name;
  shaping-rate (percent percentage | rate);
}
translation-table {
  (to-dscp-from-dscp | to-dscp-ipv6-from-dscp-ipv6 | to-exp-from-exp |
  to-inet-precedence-from-inet-precedence) table-name {

```

```
        to-code-point value from-code-points (* | [ values ] );
    }
}
tri-color;
```

On Juniper Networks MX Series 3D Universal Edge Routers with Enhanced Queuing DPCs, you can configure the following CoS statements at the **[edit class-of-service interfaces]** hierarchy level:

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

---

## [edit firewall] Hierarchy Level

The following CoS statements can be configured at the **[edit firewall]** hierarchy level. This is not a comprehensive list of statements available at the **[edit firewall]** hierarchy level.

```
[edit firewall]
  atm-policer policer-name {
    cdvt rate;
    logical-interface-policer;
    max-burst-size max-burst-size;
    peak-rate rate;
    policing-action (discard | discard-tag | count);
    sustained-rate rate;
  }
  family family-name {
    filter filter-name {
      term term-name {
        from {
          match-conditions;
        }
        then {
          dscp 0;
          forwarding-class class-name;
          loss-priority (high | low);
          three-color-policer {
            (single-rate | two-rate) policer-name;
          }
        }
      }
    }
  }
  simple-filter filter-name {
    term term-name {
      from {
        match-conditions;
      }
      then {
        forwarding-class class-name;
        loss-priority (high | low | medium);
      }
    }
  }
```

```

    }
  }
}
policer policer-name {
  logical-bandwidth-policer;
  shared-bandwidth-policer ;
  if-exceeding {
    bandwidth-limit rate;
    bandwidth-percent number;
    burst-size-limit bytes;
  }
  then {
    policer-action;
  }
}
three-color-policer policer-name {
  action {
    loss-priority high then discard;
  }
  logical-interface-policer;
  shared-bandwidth-policer ;
  single-rate {
    (color-aware | color-blind);
    committed-information-rate bps;
    committed-burst-size bytes;
    excess-burst-size bytes;
  }
  two-rate {
    (color-aware | color-blind);
    committed-information-rate bps;
    committed-burst-size bytes;
    peak-information-rate bps;
    peak-burst-size bytes;
  }
}
}

```

## [edit interfaces] Hierarchy Level

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

```

[edit interfaces]
interface-name {
  atm-options {
    linear-red-profiles profile-name {
      high-plp-max-threshold percent;
      low-plp-max-threshold percent;
      queue-depth cells high-plp-threshold percent low-plp-threshold percent;
    }
  }
  plp-to-clp;
  scheduler-maps map-name {
    forwarding-class class-name {
      epd-threshold cells plp1 cells;
    }
  }
}

```

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

On the Juniper Networks MX Series 3D Universal Edge Routers with Enhanced Queuing DPCs and on M Series and T Series routers with IQ2E PIC, you can configure the following CoS statements at the [edit interfaces] hierarchy level:

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

## [edit services cos] Hierarchy Level

The following CoS statements can be configured at the **[edit services cos]** hierarchy level. This is not a comprehensive list of statements available at the **[edit services cos]** hierarchy level. Only the statements documented in this manual are listed here. For more information about services configuration, see the Junos Services Interfaces Configuration Release 12.3.

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



## PART 3

# Index

- [Index on page 51](#)





# Index

## Symbols

|  |     |
|--|-----|
| #, comments in configuration statements..... | xii |
| ( ), in syntax descriptions.....             | xii |
| < >, in syntax descriptions.....             | xii |
| [ ], in configuration statements.....        | xii |
| { }, in configuration statements.....        | xii |
| (pipe), in syntax descriptions.....          | xii |

## A

|                                      |   |
|--------------------------------------|---|
| across-the-network applications..... | 9 |
|--------------------------------------|---|

## B

|  |     |
|--|-----|
| braces, in configuration statements..... | xii |
| brackets                                 |     |
| angle, in syntax descriptions.....       | xii |
| square, in configuration statements..... | xii |

## C

|  |     |
|--|-----|
| classifiers                                |     |
| default for VPLS.....                      | 11  |
| comments, in configuration statements..... | xii |
| components                                 |     |
| CoS.....                                   | 5   |
| conventions                                |     |
| text and syntax.....                       | xi  |
| CoS  |     |
| applications of.....                       | 9   |
| components.....                            | 5   |
| default settings.....                      | 7   |
| example classifiers configuration.....     | 33  |
| example interfaces configuration.....      | 33  |
| example IPv6 configuration.....            | 38  |
| example policy configuration.....          | 38  |
| example rewrite configuration.....         | 33  |
| example schedulers configuration.....      | 33  |
| inputs and outputs examples.....           | 29  |
| inputs and outputs overview.....           | 13  |
| interfaces without support.....            | 10  |
| J Series router packet flow.....           | 16  |
| M Series router packet flow.....           | 16  |
| MX Series router packet flow.....          | 18  |

|  |      |
|--|------|
| MX Series router packet flow example.....      | 30   |
| network packet flow.....                       | 4    |
| overview.....                                  | 3    |
| packet flow within routers.....                | 15   |
| process packet flow.....                       | 23   |
| process packet flow example.....               | 31   |
| T Series router packet flow.....               | 20   |
| curly braces, in configuration statements..... | xii  |
| customer support.....                          | xiii |
| contacting JTAC.....                           | xiii |

## D

|                  |      |
|------------------|------|
| default setting  |      |
| CoS.....         | 7    |
| DiffServ.....    | 9    |
| documentation    |      |
| comments on..... | xiii |

## E

|                                    |    |
|------------------------------------|----|
| example                            |    |
| CoS classifiers configuration..... | 33 |
| CoS configuration.....             | 33 |
| CoS inputs and outputs.....        | 29 |
| CoS IPv6 configuration.....        | 38 |
| CoS policy configuration.....      | 38 |
| CoS rewrite configuration.....     | 33 |
| CoS schedulers configuration.....  | 33 |

## F

|                       |    |
|-----------------------|----|
| font conventions..... | xi |
|-----------------------|----|

## I

|                                |    |
|--------------------------------|----|
| in-the-box applications.....   | 9  |
| interfaces                     |    |
| without CoS support.....       | 10 |
| IPv6                           |    |
| CoS configuration example..... | 38 |

## M

|                  |      |
|------------------|------|
| manuals          |      |
| comments on..... | xiii |

## N

|                     |   |
|---------------------|---|
| network packet flow |   |
| CoS.....            | 4 |

## O

### overview

|                             |    |
|-----------------------------|----|
| CoS.....                    | 3  |
| CoS inputs and outputs..... | 13 |

## P

### packet flow

|                                      |    |
|--------------------------------------|----|
| CoS on J Series router.....          | 16 |
| CoS on M Series router.....          | 16 |
| CoS on MX Series router.....         | 18 |
| CoS on MX Series router example..... | 30 |
| CoS on T Series router.....          | 20 |
| CoS process.....                     | 23 |
| CoS within routers.....              | 15 |

### packet flow example

|                  |    |
|------------------|----|
| CoS process..... | 31 |
|------------------|----|

|  |     |
|--|-----|
| parentheses, in syntax descriptions..... | xii |
|--|-----|

## S

support, technical See technical support

|                         |    |
|-------------------------|----|
| syntax conventions..... | xi |
|-------------------------|----|

## T

### technical support

|                      |      |
|----------------------|------|
| contacting JTAC..... | xiii |
|----------------------|------|

## V

### VPLS

|                              |    |
|------------------------------|----|
| default classifiers for..... | 11 |
|------------------------------|----|