Junos® OS

Class of Service Feature Guide for Security Devices

Modified: 2017-08-11

Copyright © 2017, Juniper Networks, Inc.

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

Juniper Networks, the Juniper Networks logo, Juniper, and Junos are registered trademarks of Juniper Networks, Inc. and/or its affiliates in the United States and other countries. All other trademarks may be property of their respective owners.

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

Junos® OS Class of Service Feature Guide for Security Devices
Copyright © 2017 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/. By downloading, installing or using such software, you agree to the terms and conditions of that EULA.

Table of Contents

About the Documentation xv
Documentation and Release Notes xv
Supported Platforms xv
Using the Examples in This Manual xv
Merging a Full Example xvi
Merging a Snippet xvi
Documentation Conventions xvii
Documentation Feedback xix
Requesting Technical Support xix
Self-Help Online Tools and Resources xix
Opening a Case with JTAC xx

Part 1 Overview

Chapter 1 Introduction to Class of Service 3

Understanding Class of Service 3
Benefits of CoS 4
CoS Across the Network 5
Junos OS CoS Components 6
CoS Components Packet Flow 7
CoS Process on Incoming Packets 8
CoS Process on Outgoing Packets 8
CoS Device Configuration Overview 9
Understanding CoS Default Settings 10

Part 2 Configuring Class of Service Components

Chapter 2 Assigning Service Levels with Classifiers 13

Classification Overview 13
Behavior Aggregate Classifiers 14
Multifield Classifiers 14
Default IP Precedence Classifier 15
Understanding Packet Loss Priorities 16
Default Behavior Aggregate Classification 17
Sample Behavior Aggregate Classification 18
Example: Configuring Behavior Aggregate Classifiers 19

Chapter 3 Controlling Network Access with Traffic Policing 29

Simple Filters and Policers Overview 29
Two-Rate Three-Color Policer Overview 30
Example: Configuring a Two-Rate Three-Color Policer 31

Copyright © 2017, Juniper Networks, Inc.

Logical Interface (Aggregate) Policer Overview . . . 36
Two-Color Policer Configuration Overview . . . 37
Example: Configuring a Two-Color Logical Interface (Aggregate) Policer . . . 40
Guidelines for Configuring Simple Filters . . . 46
Statement Hierarchy for Configuring Simple Filters . . . 46
Simple Filter Protocol Families . . . 46
Simple Filter Names . . . 47
Simple Filter Terms . . . 47
Simple Filter Match Conditions . . . 47
Simple Filter Terminating Actions . . . 48
Simple Filter Nonterminating Actions . . . 49
Example: Configuring and Applying a Firewall Filter for a Multifield Classifier . . . 49

Chapter 4 Controlling Output Queues with Forwarding Classes . . . 57

Forwarding Classes Overview . . . 57
Forwarding Class Queue Assignments . . . 58
Forwarding Policy Options . . . 59
Example: Configuring Forwarding Classes . . . 59
Example: Assigning Forwarding Classes to Output Queues . . . 64
Example: Assigning a Forwarding Class to an Interface . . . 66
Understanding the SPC High-Priority Queue . . . 67
Example: Configuring the SPC High-Priority Queue . . . 68
Understanding Queuing and Marking of Host Outbound Traffic . . . 70
Host Outbound Traffic Overview . . . 70
Routing Engine Sourced Traffic . . . 70
Distributed Protocol Handler Traffic . . . 70
Default Queuing and Marking of Host Outbound Traffic . . . 70
Configured Queuing and Marking of Host Outbound Traffic . . . 71
Configured Queuing and Marking of Outbound Routing Engine Traffic Only . . . 71
Default Routing Engine Protocol Queue Assignments . . . 72

Chapter 5 Altering Outgoing Packets Headers with Rewrite Rules . . . 75

Rewrite Rules Overview . . . 75
Rewriting Frame Relay Headers . . . 75
Assigning the Default Frame Relay Rewrite Rule to an Interface . . . 76
Defining a Custom Frame Relay Rewrite Rule . . . 76
Example: Configuring and Applying Rewrite Rules . . . 77

Chapter 6 Defining Output Queue Properties with Schedulers . . . 81

Schedulers Overview . . . 81
Transmit Rate . . . 82
Delay Buffer Size . . . 83
Scheduling Priority . . . 84
Shaping Rate . . . 85
Default Scheduler Settings . . . 86
Transmission Scheduling Overview . . . 87
Excess Bandwidth Sharing and Minimum Logical Interface Shaping . . . 88
Excess Bandwidth Sharing Proportional Rates . . . 89
Calculated Weights Mapped to Hardware Weights . . . 90

Copyright © 2017, Juniper Networks, Inc.

Weight Allocation with Only Shaping Rates or Unshaped Logical Interfaces 91
Shared Bandwidth Among Logical Interfaces 92
Example: Configuring Class-of-Service Schedulers 94
Scheduler Buffer Size Overview 98
Maximum Delay Buffer Sizes Available to Channelized T1/E1 Interfaces 98
Maximum Delay Buffer Size for vSRX Interfaces 99
Delay Buffer Size Allocation Methods 100
Delay Buffer Sizes for Queues 100
Example: Configuring a Large Delay Buffer on a Channelized T1 Interface 101
Configuring Large Delay Buffers in CoS 104
Example: Configuring and Applying Scheduler Maps 109

Chapter 7 Removing Delays with Strict-Priority Queues 113

Strict-Priority Queue Overview 113
Understanding Strict-Priority Queues 114
Example: Configuring Priority Scheduling 115
Example: Configuring Strict-Priority Queuing 117
Example: Configuring CoS Non-Strict Priority Scheduling 126

Chapter 8 Controlling Congestion with Drop Profiles 131

RED Drop Profiles Overview 131
Default Drop Profiles 132
RED Drop Profiles and Congestion Control 132
Configuring RED Drop Profiles 134
Example: Configuring RED Drop Profiles 135
Example: Configuring Segmented and Interpolated Style Profiles 137

Chapter 9 Controlling Congestion with Adaptive Shapers 143

Adaptive Shaping Overview 143
Assigning the Default Frame Relay Loss Priority Map to an Interface 144
Defining a Custom Frame Relay Loss Priority Map 144
Example: Configuring and Applying an Adaptive Shaper 145

Chapter 10 Limiting Traffic Using Virtual Channels 147

Virtual Channels Overview 147
Understanding Virtual Channels 148
Example: Configuring Virtual Channels 149

Chapter 11 Enabling Queuing for Tunnel Interfaces 155

CoS Queuing for Tunnels Overview 155
Benefits of CoS Queuing for Tunnel Interfaces 156
Configuring CoS on Logical Tunnels 156
How CoS Queuing Works 158
Limitations on CoS Shapers for Tunnel Interfaces 159
Understanding the ToS Value of a Tunnel Packet 159
Example: Configuring CoS Queuing for GRE or IP-IP Tunnels 160
Copying Outer IP Header DSCP and ECN to Inner IP Header 164

Copyright © 2017, Juniper Networks, Inc.

Chapter 12 Naming Components with Code-Point Aliases 167
Code-Point Aliases Overview 167
Default CoS Values and Aliases 168
Example: Defining Code-Point Aliases for Bits 171

Part 3 Configuring Class of Service Scheduler Hierarchy

Chapter 13 Controlling Traffic by Configuring Scheduler Hierarchy 175
Understanding Hierarchical Schedulers 175
Understanding Internal Scheduler Nodes 178
SRX1400, SRX3400, and SRX3600 Device Hardware Capabilities and Limitations 179
Example: Configuring a Four-Level Scheduler Hierarchy 181
Example: Controlling Remaining Traffic 193

Part 4 Configuring Class of Service for IPv6

Chapter 14 Configuring Class of Service for IPv6 Traffic 201
CoS Functions for IPv6 Traffic Overview 201
Understanding CoS with DSCP IPv6 BA Classifier 203
Example: Configuring CoS with DSCP IPv6 BA Classifiers 205
Understanding DSCP IPv6 Rewrite Rules 208
Example: Configuring CoS with DSCP IPv6 Rewrite Rules 209

Part 5 Configuring Class of Service for I/O Cards

Chapter 15 Configuring Class of Service for I/O Cards 215
PIR-Only and CIR Mode Overview 215
PIR-only Mode 215
CIR Mode 216
Understanding Priority Propagation 217
Understanding IOC Hardware Properties 218
Understanding IOC Map Queues 220
WRED on the IOC Overview 221
Shapers at the Logical Interface Level (Level 3) 222
Shapers at the Interface Set Level (Level 2) 223
Shapers at the Port Level (Level 1) 224
MDRR on the IOC Overview 225
CoS Support on the SRX5000 Module Port Concentrator Overview 227
Example: Configuring CoS on SRX5000 Devices with an MPC 228

Part 6 Configuration Statements and Operational Commands

Chapter 16 Configuration Statements 241
adaptive-shaper 242
adaptive-shapers 243
application-traffic-control 244
buffer-size (Schedulers) 245
classifiers (CoS) 246
code-points (CoS) 247

Copyright © 2017, Juniper Networks, Inc.

copy-outer-dscp . . . 247
default (CoS) . . . 248
dscp-code-point (CoS Host Outbound Traffic) . . . 249
egress-shaping-overhead . . . 250
forwarding-class (CoS Host Outbound Traffic) . . . 252
forwarding-classes (CoS) . . . 253
frame-relay-de (CoS Interfaces) . . . 254
frame-relay-de (CoS Loss Priority) . . . 255
frame-relay-de (CoS Rewrite Rule) . . . 256
host-outbound-traffic (Class-of-Service) . . . 257
ingress-policer-overhead . . . 258
interfaces (CoS) . . . 260
logical-interface-policer . . . 261
loss-priority (CoS Loss Priority) . . . 262
loss-priority (CoS Rewrite Rules) . . . 263
loss-priority-maps (CoS Interfaces) . . . 264
loss-priority-maps (CoS) . . . 264
non-strict-priority-scheduling . . . 265
rate-limiters . . . 266
rewrite-rules (CoS) . . . 267
rewrite-rules (CoS Interfaces) . . . 268
rule-sets (CoS AppQoS) . . . 269
scheduler-map (CoS Virtual Channels) . . . 271
shaping-rate (CoS Adaptive Shapers) . . . 272
shaping-rate (CoS Interfaces) . . . 273
shaping-rate (CoS Virtual Channels) . . . 274
trigger (CoS) . . . 275
tunnel-queuing . . . 275
virtual-channels . . . 276
virtual-channel-group (CoS Interfaces) . . . 277
virtual-channel-groups . . . 278

Chapter 17 Operational Commands . . . 279

show class-of-service application-traffic-control counter . . . 280
show class-of-service application-traffic-control statistics rate-limiter . . . 282
show class-of-service application-traffic-control statistics rule . . . 284
show class-of-service forwarding-class . . . 286

Copyright © 2017, Juniper Networks, Inc.

Copyright © 2017, Juniper Networks, Inc.

List of Figures

Part 1 Overview

Chapter 1 **Introduction to Class of Service . . . 3**

Figure 1: CoS Across the Network . . . 5
Figure 2: Packet Flow Through Juniper Networks Device . . . 7

Part 2 Configuring Class of Service Components

Chapter 2 **Assigning Service Levels with Classifiers . . . 13**

Figure 3: Behavior Aggregate Classifier Scenario . . . 20

Chapter 3 **Controlling Network Access with Traffic Policing . . . 29**

Figure 4: Multifield Classifier Based on TCP Source Ports . . . 50
Figure 5: Multifield Classifier Scenario . . . 50

Chapter 7 **Removing Delays with Strict-Priority Queues . . . 113**

Figure 6: CoS Traffic with HIgh and Low Priority Queues . . . 126

Chapter 8 **Controlling Congestion with Drop Profiles . . . 131**

Figure 7: Segmented and Interpolated Drop Profiles . . . 134

Chapter 11 **Enabling Queuing for Tunnel Interfaces . . . 155**

Figure 8: CoS Solutions Using Logical Tunnels . . . 157
Figure 9: CoS Processing for Tunnel Traffic . . . 158
Figure 10: Configuring CoS Queuing for GRE Tunnels . . . 161

Part 3 Configuring Class of Service Scheduler Hierarchy

Chapter 13 **Controlling Traffic by Configuring Scheduler Hierarchy . . . 175**

Figure 11: Building a Scheduler Hierarchy . . . 182
Figure 12: Example 1 Handling Remaining Traffic with no Explicit Traffic Control Profile . . . 194
Figure 13: Example 2 Handling Remaining Traffic with an Interface Set . . . 194

Part 4 Configuring Class of Service for IPv6

Chapter 14 **Configuring Class of Service for IPv6 Traffic . . . 201**

Figure 14: Packet Flow Through an SRX Series Device . . . 201

Part 5 Configuring Class of Service for I/O Cards

Chapter 15 **Configuring Class of Service for I/O Cards . . . 215**

Figure 15: Hierarchical Schedulers and Priorities . . . 218

Copyright © 2017, Juniper Networks, Inc.

Copyright © 2017, Juniper Networks, Inc.

List of Tables

About the Documentation xv

Table 1: Notice Icons xvii
Table 2: Text and Syntax Conventions xviii

Part 1 Overview

Chapter 1 Introduction to Class of Service 3

Table 3: Supported Junos OS CoS Components 6
Table 4: Reasons to Configure Class of Service (CoS) 10

Part 2 Configuring Class of Service Components

Chapter 2 Assigning Service Levels with Classifiers 13

Table 5: BA Classification 14
Table 6: MF Classification 15
Table 7: Default IP Precedence Classifier 16
Table 8: Default Behavior Aggregate Classification 17
Table 9: Sample Behavior Aggregate Classification Forwarding Classes and Queues 18
Table 10: Sample ba-classifier Loss Priority Assignments 20

Chapter 3 Controlling Network Access with Traffic Policing 29

Table 11: Two-Color Policer Configuration and Application Overview 37
Table 12: Simple Filter Match Conditions 48

Chapter 4 Controlling Output Queues with Forwarding Classes 57

Table 13: Default Forwarding Class Queue Assignments 58
Table 14: Sample Output Queue Assignments for mf-classifier Forwarding Queues 64
Table 15: Default Queue Assignments for Packets Generated by the Routing Engine 72

Chapter 5 Altering Outgoing Packets Headers with Rewrite Rules 75

Table 16: Sample rewrite-dscps Rewrite Rules to Replace DSCPs 77

Chapter 6 Defining Output Queue Properties with Schedulers 81

Table 17: Sample Transmission Scheduling 87
Table 18: Shaping Rates and WFQ Weights 89
Table 19: Example Shaping Rates and WFQ Weights 89
Table 20: Rounding Configured Weights to Hardware Weights 90
Table 21: Allocating Weights with PIR and CIR on Logical Interfaces 91
Table 22: Example of Shared Bandwidth Among Logical Interfaces 92

Copyright © 2017, Juniper Networks, Inc.

Table 23: First Example of Bandwidth Sharing . . . 93
Table 24: Second Example of Bandwidth Sharing . . . 93
Table 25: Final Example of Bandwidth Sharing . . . 93
Table 26: Sample Schedulers . . . 95
Table 27: Maximum Available Delay Buffer Time by Channelized Interface and Rate . . . 98
Table 28: Delay Buffer Size Allocation Methods . . . 100
Table 29: Delay Buffer Allocation Method and Queue Buffer . . . 101
Table 30: Interface Delay Times Enabled By q-pic-large-buffer . . . 107
Table 31: Sample diffserv-cos-map Scheduler Mapping . . . 109

Chapter 8 **Controlling Congestion with Drop Profiles . . . 131**

Table 32: Sample RED Drop Profiles . . . 132
Table 33: Configuring RED Drop Profiles for Assured Forwarding Congestion Control . . . 133
Table 34: Sample RED Drop Profiles . . . 135

Chapter 12 **Naming Components with Code-Point Aliases . . . 167**

Table 35: Standard CoS Aliases and Bit Values . . . 169

Part 3 Configuring Class of Service Scheduler Hierarchy

Chapter 13 **Controlling Traffic by Configuring Scheduler Hierarchy . . . 175**

Table 36: Hierarchical Scheduler Nodes . . . 175
Table 37: Available NPCs and IO Ports for SRX1400, SRX3400, and SRX3600 Devices . . . 179

Part 4 Configuring Class of Service for IPv6

Chapter 14 **Configuring Class of Service for IPv6 Traffic . . . 201**

Table 38: Default IPv6 BA Classifier Mapping . . . 203

Part 5 Configuring Class of Service for I/O Cards

Chapter 15 **Configuring Class of Service for I/O Cards . . . 215**

Table 39: Internal Node Queue Priority for PIR-Only Mode . . . 215
Table 40: Internal Node Queue Priority for CIR Mode . . . 216
Table 41: Queue Priority . . . 217
Table 42: Packet Forwarding Engine Properties within 40x1GE IOC and 4x10GE IOC . . . 219
Table 43: Shaper Accuracy of 1-Gbps Ethernet at the Logical Interface Level . . . 222
Table 44: Shaper Accuracy of 10-Gbps Ethernet at the Logical Interface Level . . . 223
Table 45: Shaper Accuracy of 1-Gbps Ethernet at the Interface Set Level . . . 224
Table 46: Shaper Accuracy of 10-Gbps Ethernet at the Interface Set Level . . . 224
Table 47: Shaper Accuracy of 1-Gbps Ethernet at the Physical Port Level . . . 224
Table 48: Shaper Accuracy of 10-Gbps Ethernet at the Physical Port Level . . . 224
Table 49: Junos Priorities Mapped to IOC Hardware Priorities . . . 225
Table 50: Forwarding Class Samples . . . 229
Table 51: Scheduler Samples . . . 229

Copyright © 2017, Juniper Networks, Inc.

Part 6 Configuration Statements and Operational Commands

Chapter 17 Operational Commands 279

Table 52: show class-of-service application-traffic-control counter Output Fields 280

Table 53: show class-of-service application-traffic-control statistics rate-limiter Output Fields 282

Table 54: show class-of-service application-traffic-control statistics rule Output Fields 284

Table 55: show class-of-service forwarding-class Output Fields 286

Copyright © 2017, Juniper Networks, Inc.

Copyright © 2017, Juniper Networks, Inc.

About the Documentation

- Documentation and Release Notes on page xv
- Supported Platforms on page xv
- Using the Examples in This Manual on page xv
- Documentation Conventions on page xvii
- Documentation Feedback on page xix
- Requesting Technical Support on page xix

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:

- vSRX
- SRX 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.

Copyright © 2017, Juniper Networks, Inc.

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

Merging a Full Example

To merge a full example, follow these steps:

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

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

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

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

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

Merging a Snippet

To merge a snippet, follow these steps:

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

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

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


Copyright © 2017, Juniper Networks, Inc.

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 CLI Explorer.

Documentation Conventions

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

Table 1: Notice Icons

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

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

Copyright © 2017, Juniper Networks, Inc.

Table 2: Text and Syntax Conventions

Convention	Description	Examples
Bold text like this	Represents text that you type.	To enter configuration mode, type the **configure** command: user@host> **configure**
`Fixed-width text like this`	Represents output that appears on the terminal screen.	`user@host>` **show chassis alarms** `No alarms currently active`
Italic text like this	• Introduces or emphasizes important new terms. • Identifies guide names. • Identifies RFC and Internet draft titles.	• A policy *term* is a named structure that defines match conditions and actions. • *Junos OS CLI User Guide* • RFC 1997, *BGP Communities Attribute*
Italic text like this	Represents variables (options for which you substitute a value) in commands or configuration statements.	Configure the machine's domain name: [edit] root@# **set system domain-name** *domain-name*
Text like this	Represents names of configuration statements, commands, files, and directories; configuration hierarchy levels; or labels on routing platform components.	• To configure a stub area, include the **stub** statement at the **[edit protocols ospf area area-id]** hierarchy level. • The console port is labeled **CONSOLE**.
< > (angle brackets)	Encloses optional keywords or variables.	**stub <default-metric** *metric***>;**
| (pipe symbol)	Indicates a choice between the mutually exclusive keywords or variables on either side of the symbol. The set of choices is often enclosed in parentheses for clarity.	**broadcast | multicast** **(***string1* **|** *string2* **|** *string3***)**
# (pound sign)	Indicates a comment specified on the same line as the configuration statement to which it applies.	**rsvp { # Required for dynamic MPLS only**
[ ] (square brackets)	Encloses a variable for which you can substitute one or more values.	**community name members [** *community-ids* **]**

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

; (semicolon)	Identifies a leaf statement at a configuration hierarchy level.	

GUI Conventions

Copyright © 2017, Juniper Networks, Inc.

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

Convention	Description	Examples
Bold text like this	Represents graphical user interface (GUI) items you click or select.	• In the Logical Interfaces box, select **All Interfaces**. • To cancel the configuration, click **Cancel**.
> (bold right angle bracket)	Separates levels in a hierarchy of menu selections.	In the configuration editor hierarchy, select **Protocols>Ospf**.

Documentation Feedback

We encourage you to provide feedback, comments, and suggestions so that we can improve the documentation. You can provide feedback by using either of the following methods:

- Online feedback rating system—On any page of the Juniper Networks TechLibrary site at http://www.juniper.net/techpubs/index.html, simply click the stars to rate the content, and use the pop-up form to provide us with information about your experience. Alternately, you can use the online feedback form at http://www.juniper.net/techpubs/feedback/.
- E-mail—Send your comments to **techpubs-comments@juniper.net**. Include the document or topic name, URL or page number, and software version (if applicable).

Requesting Technical Support

Technical product support is available through the Juniper Networks Technical Assistance Center (JTAC). If you are a customer with an active J-Care or Partner Support Service 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:

Copyright © 2017, Juniper Networks, Inc.

- Find CSC offerings: http://www.juniper.net/customers/support/
- Search for known bugs: http://www2.juniper.net/kb/
- Find product documentation: http://www.juniper.net/techpubs/
- Find solutions and answer questions using our Knowledge Base: http://kb.juniper.net/
- Download the latest versions of software and review release notes: http://www.juniper.net/customers/csc/software/
- Search technical bulletins for relevant hardware and software notifications: http://kb.juniper.net/InfoCenter/
- Join and participate in the Juniper Networks Community Forum: http://www.juniper.net/company/communities/
- Open a case online in the CSC Case Management tool: http://www.juniper.net/cm/

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

Opening a Case with JTAC

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

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

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

Copyright © 2017, Juniper Networks, Inc.

PART 1

Overview

- Introduction to Class of Service on page 3

Copyright © 2017, Juniper Networks, Inc.

Copyright © 2017, Juniper Networks, Inc.

CHAPTER 1

Introduction to Class of Service

- Understanding Class of Service on page 3
- Benefits of CoS on page 4
- CoS Across the Network on page 5
- Junos OS CoS Components on page 6
- CoS Components Packet Flow on page 7
- CoS Device Configuration Overview on page 9
- Understanding CoS Default Settings on page 10

Understanding Class of Service

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

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

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

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

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

Copyright © 2017, Juniper Networks, Inc.

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

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

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

Related Documentation

- Junos OS CoS Components on page 6
- CoS Components Packet Flow on page 7
- Understanding CoS Default Settings on page 10
- CoS Device Configuration Overview on page 9

Benefits of CoS

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

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

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

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

Copyright © 2017, Juniper Networks, Inc.

Related Documentation
- Understanding Class of Service on page 3

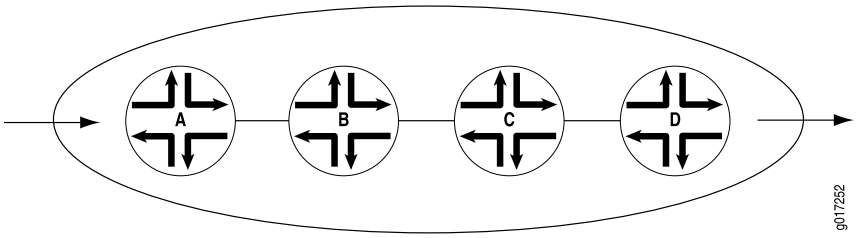
CoS Across the Network

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

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

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

Figure 1: CoS Across the Network



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

Related Documentation
- Understanding Class of Service on page 3

Copyright © 2017, Juniper Networks, Inc.

Junos OS CoS Components

Junos OS supports CoS components on Juniper Networks devices as indicated in Table 3 on page 6.

Table 3: Supported Junos OS CoS Components

Junos OS CoS Component	Description	For More Information
Code-point aliases	A code-point alias assigns a name to a pattern of code-point bits. You can use this name, instead of the bit pattern, when you configure other CoS components such as classifiers, drop-profile maps, and rewrite rules.	"Code-Point Aliases Overview" on page 167
Classifiers	Packet classification refers to the examination of an incoming packet. This function associates the packet with a particular CoS servicing level. Two general types of classifiers are supported—behavior aggregate (BA) classifiers and multifield (MF) classifiers. When both BA and MF classifications are performed on a packet, the MF classification has higher precedence.	"Classification Overview" on page 13
Forwarding classes	Forwarding classes allow you to group packets for transmission. Based on forwarding classes, you assign packets to output queues. The forwarding class plus the loss priority define the per-hop behavior (PHB in DiffServ) of a packet. Juniper Networks routers and services gateways support eight queues (0 through 7).	"Forwarding Classes Overview" on page 57
Loss priorities	Loss priorities allow you to set the priority of dropping a packet. You can use the loss priority setting to identify packets that have experienced congestion.	"Understanding Packet Loss Priorities" on page 16
Forwarding policy options	CoS-based forwarding (CBF) enables you to control next-hop selection based on a packet's class of service and, in particular, the value of the IP packet's precedence bits. For example, you can specify a particular interface or next hop to carry high-priority traffic while all best-effort traffic takes some other path.	"Example: Assigning a Forwarding Class to an Interface" on page 66
Transmission queues	After a packet is sent to the outgoing interface on a device, it is queued for transmission on the physical media. The amount of time a packet is queued on the device is determined by the availability of the outgoing physical media as well as the amount of traffic using the interface. Juniper Networks routers and services gateways support queues 0 through 7.	"Transmission Scheduling Overview" on page 87
Schedulers	An individual device interface has multiple queues assigned to store packets temporarily before transmission. To determine the order to service the queues, the device uses a round-robin scheduling method based on priority and the queue's weighted round-robin (WRR) credits. Junos OS schedulers allow you to define the priority, bandwidth, delay buffer size, rate control status, and RED drop profiles to be applied to a particular queue for packet transmission.	"Schedulers Overview" on page 81
Virtual channels	On Juniper Networks routers and services gateways, you can configure virtual channels to limit traffic sent from a corporate headquarters to branch offices. Virtual channels might be required when the headquarters site has an expected aggregate bandwidth higher than that of the individual branch offices. The router at the headquarters site must limit the traffic sent to each branch office router to avoid oversubscribing their links.	"Virtual Channels Overview" on page 147

Copyright © 2017, Juniper Networks, Inc.

Table 3: Supported Junos OS CoS Components *(continued)*

Junos OS CoS Component	Description	For More Information
Policers for traffic classes	Policers allow you to limit traffic of a certain class to a specified bandwidth and burst size. Packets exceeding the policer limits can be discarded, or can be assigned to a different forwarding class, a different loss priority, or both. You define policers with firewall filters that can be associated with input or output interfaces.	"Simple Filters and Policers Overview" on page 29
Rewrite rules	A rewrite rule modifies the appropriate CoS bits in an outgoing packet. Modification of CoS bits allows the next downstream device to classify the packet into the appropriate service group. Rewriting or marking outbound packets is useful when the device is at the border of a network and must alter the CoS values to meet the policies of the targeted peer.	"Rewrite Rules Overview" on page 75

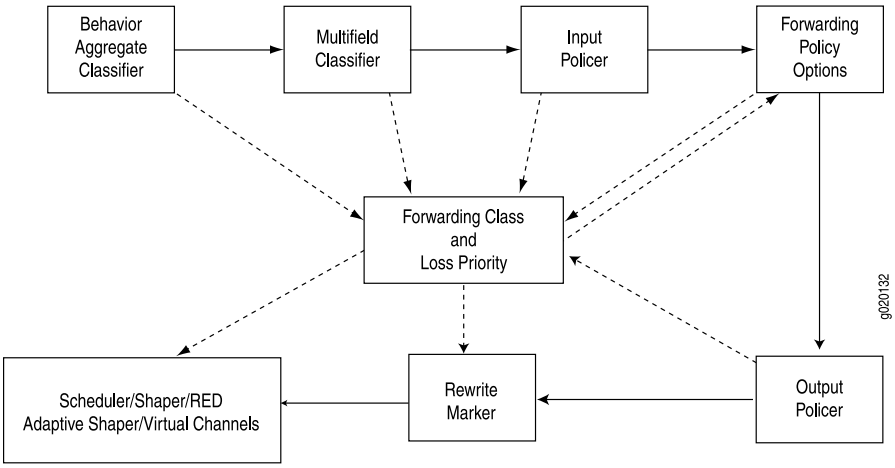
Related Documentation

- Understanding Class of Service on page 3
- CoS Components Packet Flow on page 7
- Understanding CoS Default Settings on page 10
- CoS Device Configuration Overview on page 9

CoS Components Packet Flow

On Juniper Networks devices, you configure CoS functions using different components. These components are configured individually or in a combination to define particular CoS services. Figure 2 on page 7 displays the relationship of different CoS components to each other and illustrates the sequence in which they interact.

Figure 2: Packet Flow Through Juniper Networks Device



Each box in Figure 2 on page 7 represents a CoS component. The solid lines show the direction of packet flow in a device. The upper row indicates an incoming packet, and the lower row an outgoing packet. The dotted lines show the inputs and outputs of particular CoS components. For example, the forwarding class and loss priority are

Copyright © 2017, Juniper Networks, Inc.

outputs of behavior aggregate classifiers and multifield classifiers and inputs for rewrite markers and schedulers.

Typically, only a combination of some components shown in Figure 2 on page 7 (not all) is used to define a CoS service offering. For example, if a packet's class is determined by a behavior aggregate classifier, it is associated with a forwarding class and loss priority and does not need further classification by the multifield classifier.

This section contains the following topics:

- CoS Process on Incoming Packets on page 8
- CoS Process on Outgoing Packets on page 8

CoS Process on Incoming Packets

Classifiers and policers perform the following operations on incoming packets:

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

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

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

CoS Process on Outgoing Packets

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

1. Scheduler maps are applied to interfaces and associate the outgoing packets with a scheduler and a forwarding class.

2. The scheduler defines how the packet is treated in the output transmission queue based on the configured transmit rate, buffer size, priority, and drop profile.
   - The buffer size defines the period for which the packet is stored during congestion.
   - The scheduling priority and transmit rate determine the order in which the packet is transmitted.
   - The drop profile defines how aggressively to drop packets that are using a particular scheduler.

Copyright © 2017, Juniper Networks, Inc.

3. Output policers meter traffic and might change the forwarding class and loss priority of a packet if a traffic flow exceeds its service level.

4. The rewrite rule writes information to the packet (for example, EXP or DSCP bits) according to the forwarding class and loss priority of the packet.

Related Documentation

- Understanding Class of Service on page 3
- Junos OS CoS Components on page 6
- Understanding CoS Default Settings on page 10
- CoS Device Configuration Overview on page 9

CoS Device Configuration Overview

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

- Determine whether the device needs to support different traffic streams, such as voice or video. If so, CoS helps to make sure this traffic receives more than basic best-effort packet delivery service.
- Determine whether the device is directly attached to any applications that send CoS-classified packets. If no sources are enabled for CoS, you must configure and apply rewrite rules on the interfaces facing the sources.
- Determine whether the device must support assured forwarding (AF) classes. Assured forwarding usually requires random early detection (RED) drop profiles to be configured and applied.
- Determine whether the device must support expedited forwarding (EF) classes with a policer. Policers require you to apply a burst size and bandwidth limit to the traffic flow, and set a consequence for packets that exceed these limits—usually a high loss priority, so that packets exceeding the policer limits are discarded first.

NOTE: When the T1/E1 mPIM is oversubscribed, we recommend that you configure its shaping rate for consistent CoS functionality. The shaping rate should be less than the total link speed.

Related Documentation

- CLI Explorer
- Understanding Class of Service on page 3
- CoS Components Packet Flow on page 7
- Understanding CoS Default Settings on page 10

Copyright © 2017, Juniper Networks, Inc.

Understanding CoS Default Settings

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

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

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

You configure CoS when you need to override the default packet forwarding behavior of a Juniper Networks device—especially in the three areas identified in Table 4 on page 10.

Table 4: Reasons to Configure Class of Service (CoS)

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

Related Documentation

- Understanding Class of Service on page 3
- CoS Components Packet Flow on page 7
- CoS Device Configuration Overview on page 9

Copyright © 2017, Juniper Networks, Inc.

PART 2

Configuring Class of Service Components

- Assigning Service Levels with Classifiers on page 13
- Controlling Network Access with Traffic Policing on page 29
- Controlling Output Queues with Forwarding Classes on page 57
- Altering Outgoing Packets Headers with Rewrite Rules on page 75
- Defining Output Queue Properties with Schedulers on page 81
- Removing Delays with Strict-Priority Queues on page 113
- Controlling Congestion with Drop Profiles on page 131
- Controlling Congestion with Adaptive Shapers on page 143
- Limiting Traffic Using Virtual Channels on page 147
- Enabling Queuing for Tunnel Interfaces on page 155
- Naming Components with Code-Point Aliases on page 167

Copyright © 2017, Juniper Networks, Inc.

Copyright © 2017, Juniper Networks, Inc.

CHAPTER 2

Assigning Service Levels with Classifiers

- Classification Overview on page 13
- Understanding Packet Loss Priorities on page 16
- Default Behavior Aggregate Classification on page 17
- Sample Behavior Aggregate Classification on page 18
- Example: Configuring Behavior Aggregate Classifiers on page 19

Classification Overview

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

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

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

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

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

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

Copyright © 2017, Juniper Networks, Inc.

according to its FC and then manage the queues by checking the packet's PLP. Junos OS supports up to eight FCs and four PLPs.

This topic includes the following sections:

- Behavior Aggregate Classifiers on page 14
- Multifield Classifiers on page 14
- Default IP Precedence Classifier on page 15

Behavior Aggregate Classifiers

A BA classifier operates on a packet as it enters the device. Using BA classifiers, the device aggregates different types of traffic into a single FC so that all the types of traffic will receive the same forwarding treatment. The CoS value in the packet header is the single field that determines the CoS settings applied to the packet. BA classifiers allow you to set a packet's FC and PLP based on the Differentiated Services (DiffServ) code point (DSCP) value, DSCP IPv4 value, DSCP IPv6 value, IP precedence value, MPLS EXP bits, or IEEE 802.1p value. The default classifier is based on the IP precedence value. For more information, see "Default IP Precedence Classifier" on page 15.

Junos OS performs BA classification for a packet by examining its Layer 2, Layer 3, and related CoS parameters, as shown in Table 5 on page 14.

Table 5: BA Classification

Layer	CoS Parameter
Layer 2	IEEE 802.1p value: User Priority
Layer 3	IPv4 precedence IPv4 Differentiated Services code point (DSCP) value IPv6 DSCP value

NOTE: A BA classifier evaluates Layer 2 and Layer 3 parameters independently. The results from Layer 2 parameters override the results from the Layer 3 parameters.

Multifield Classifiers

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

Copyright © 2017, Juniper Networks, Inc.

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

Junos OS performs MF traffic classification by directly scrutinizing multiple fields of a packet to classify a packet. This avoids having to rely on the output of the previous BA traffic classification. Junos OS can simultaneously check a packet's data for Layers 2, 3, 4, and 7, as shown in Table 6 on page 15.

Table 6: MF Classification

Layer	CoS Parameter
Layer 2	IEEE 802.1Q: VLAN ID IEEE 802.1p: User priority
Layer 3	IP precedence value DSCP or DSCP IPv6 value Source IP address Destination IP address Protocol ICMP: Code and type
Layer 4	TCP/UDP: Source port TCP/UDP: Destination port TCP: Flags AH/ESP: SPI
Layer 7	Not supported.

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

Default IP Precedence Classifier

With Junos OS, all logical interface are automatically assigned a default IP precedence classifier when the logical interface is configured. This default traffic classifier maps IP precedence values to an FC and a PLP as shown in Table 7 on page 16. These mapping results are in effect for an ingress packet until the packet is further processed by another classification method.

Copyright © 2017, Juniper Networks, Inc.

Table 7: Default IP Precedence Classifier

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

Related Documentation

- Default Behavior Aggregate Classification on page 17
- Sample Behavior Aggregate Classification on page 18
- Example: Configuring Behavior Aggregate Classifiers on page 19

Understanding Packet Loss Priorities

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

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

Related Documentation

- Classification Overview on page 13
- Default Behavior Aggregate Classification on page 17
- Sample Behavior Aggregate Classification on page 18
- Example: Configuring Behavior Aggregate Classifiers on page 19

Copyright © 2017, Juniper Networks, Inc.

Default Behavior Aggregate Classification

Table 8 on page 17 shows the forwarding class (FC) and packet loss priority (PLP) that are assigned by default to each well-known Differentiated Services (DiffServ) code point (DSCP). Although several DSCPs map to the expedited-forwarding (ef) and assured-forwarding (af) classes, by default no resources are assigned to these forwarding classes. All af classes other than af1*x* are mapped to best-effort, because RFC 2597, *Assured Forwarding PHB Group*, prohibits a node from aggregating classes. Assignment to the best-effort FC implies that the node does not support that class. You can modify the default settings through configuration.

Table 8: Default Behavior Aggregate Classification

DSCP and DSCP IPv6 Alias	Forwarding Class	Packet Loss Priority
ef	expedited-forwarding	low
af11	assured-forwarding	low
af12	assured-forwarding	high
af13	assured-forwarding	high
af21	best-effort	low
af22	best-effort	low
af23	best-effort	low
af31	best-effort	low
af32	best-effort	low
af33	best-effort	low
af41	best-effort	low
af42	best-effort	low
af43	best-effort	low
be	best-effort	low
cs1	best-effort	low
cs2	best-effort	low
cs3	best-effort	low
cs4	best-effort	low

Copyright © 2017, Juniper Networks, Inc.

Table 8: Default Behavior Aggregate Classification *(continued)*

DSCP and DSCP IPv6 Alias	Forwarding Class	Packet Loss Priority
cs5	best-effort	low
nc1/cs6	network-control	low
nc2/cs7	network-control	low
other	best-effort	low

Related Documentation

- Classification Overview on page 13
- Sample Behavior Aggregate Classification on page 18
- Example: Configuring Behavior Aggregate Classifiers on page 19
- Understanding Packet Loss Priorities on page 16

Sample Behavior Aggregate Classification

Table 9 on page 18 shows the device forwarding classes (FCs) associated with each well-known Differentiated Services (DiffServ) code point (DSCP) and the resources assigned to the output queues for a sample DiffServ CoS implementation. This example assigns expedited forwarding to queue 1 and a subset of the assured FCs (af*x*) to queue 2, and distributes resources among all four forwarding classes. Other DiffServ-based implementations are possible.

Table 9: Sample Behavior Aggregate Classification Forwarding Classes and Queues

DSCP and DSCP IPv6 Alias	DSCP and DSCP IPv6 Bits	Forwarding Class	Packet Loss Priority	Queue
ef	101110	expedited-forwarding	low	1
af11	001010	assured-forwarding	low	2
af12	001100	assured-forwarding	high	2
af13	001110	assured-forwarding	high	2
af21	010010	best-effort	low	0
af22	010100	best-effort	low	0
af23	010110	best-effort	low	0
af31	011010	best-effort	low	0
af32	011100	best-effort	low	0

Copyright © 2017, Juniper Networks, Inc.

Table 9: Sample Behavior Aggregate Classification Forwarding Classes and Queues *(continued)*

DSCP and DSCP IPv6 Alias	DSCP and DSCP IPv6 Bits	Forwarding Class	Packet Loss Priority	Queue
af33	011110	best-effort	low	0
af41	100010	best-effort	low	0
af42	100100	best-effort	low	0
af43	100110	best-effort	low	0
be	000000	best-effort	low	0
cs1	0010000	best-effort	low	0
cs2	010000	best-effort	low	0
cs3	011000	best-effort	low	0
cs4	100000	best-effort	low	0
cs5	101000	best-effort	low	0
nc1/cs6	110000=	network-control	low	3
nc2/cs7	111000=	network-control	low	3
other	—	best-effort	low	0

Related Documentation

- Classification Overview on page 13
- Default Behavior Aggregate Classification on page 17
- Example: Configuring Behavior Aggregate Classifiers on page 19
- Understanding Packet Loss Priorities on page 16

Example: Configuring Behavior Aggregate Classifiers

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

- Requirements on page 20
- Overview on page 20
- Configuration on page 21
- Verification on page 23

Copyright © 2017, Juniper Networks, Inc.

Requirements

Before you begin, determine the forwarding class and PLP that are assigned by default to each well-known DSCP that you want to configure for the behavior aggregate classifier. See "Default Behavior Aggregate Classification" on page 17.

Overview

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

In this example, you set the DSCP behavior aggregate classifier to ba-classifier as the default DSCP map. You set a best-effort forwarding class as be-class, an expedited forwarding class as ef-class, an assured forwarding class as af-class, and a network control forwarding class as nc-class. Finally, you apply the behavior aggregate classifier to an interface called ge-0/0/0.

Table 10 on page 20 shows how the behavior aggregate classifier assigns loss priorities, to incoming packets in the four forwarding classes.

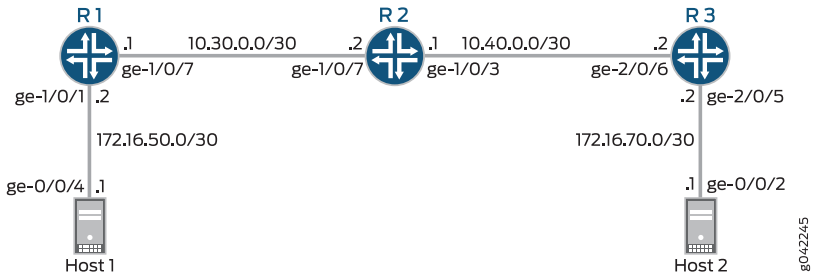
Table 10: Sample ba-classifier Loss Priority Assignments

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

Topology

Figure 3 on page 20 shows the sample network.

Figure 3: Behavior Aggregate Classifier Scenario



Copyright © 2017, Juniper Networks, Inc.

"CLI Quick Configuration" on page 21 shows the configuration for all of the Juniper Networks devices in Figure 3 on page 20.

The section "Step-by-Step Procedure" on page 21 describes the steps on Device R2.

Configuration

CLI Quick Configuration

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

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

Step-by-Step Procedure

The following example requires you to navigate various levels in the configuration hierarchy. For instructions on how to do that, see *Using the CLI Editor in Configuration Mode* in the CLI User Guide.

To configure behavior aggregate classifiers for a device:

1. Configure the class of service.

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

2. Configure behavior aggregate classifiers for DiffServ CoS.

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

3. Configure a best-effort forwarding class classifier.

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

4. Configure an expedited forwarding class classifier.

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

5. Configure an assured forwarding class classifier.

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

Copyright © 2017, Juniper Networks, Inc.

6. Configure a network control forwarding class classifier.

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

7. Apply the behavior aggregate classifier to an interface.

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

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

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

```
[edit]
user@host# show class-of-service
classifiers {
  dscp ba-classifier {
    import default;
    forwarding-class be-class {
      loss-priority high code-points 000001;
    }
    forwarding-class ef-class {
      loss-priority high code-points 101111;
    }
    forwarding-class af-class {
      loss-priority high code-points 001100;
    }
    forwarding-class nc-class {
      loss-priority high code-points 110001;
    }
  }
  forwarding-classes {
    class BE-data queue-num 0;
    class Premium-data queue-num 1;
    class Voice queue-num 2;
    class NC queue-num 3;
  }
  interfaces {
    ge-0/0/0 {
      unit 0 {
        classifiers {
          dscp ba-classifier;
        }
      }
    }
    ge-1/0/9 {
```


Copyright © 2017, Juniper Networks, Inc.

```
unit 0 {
  classifiers {
    dscp v4-ba-classifier;
  }
  ge-1/0/9 {
    unit 0 {
      classifiers {
        dscp v4-ba-classifier;
      }
      ge-1/0/9 {
        unit 0 {
          classifiers {
            dscp v4-ba-classifier;
          }
        }
      }
    }
```

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

Verification

Confirm that the configuration is working properly.

- Verifying the Code-Point Aliases on page 23
- Verifying the DSCP Classifier on page 24
- Verifying the Forwarding Classes and Output Queues on page 25
- Verifying That the Classifier Is Applied to the Interfaces on page 26
- Verifying Behavior Aggregate Classifiers on page 26

Verifying the Code-Point Aliases

Purpose Make sure that the code-point aliases are configured as expected.

Action On Device R2, run the **show class-of-service code-point-aliases dscp** command.

```
user@R2> show class-of-service code-point-aliases dscp

Code point type: dscp
  Alias              Bit pattern
  af11               001010
  af12               001100
  af13               001110
  af21               010010
  af22               010100
  af23               010110
  af31               011010
  af32               011100
  af33               011110
  af41               100010
  af42               100100
  af43               100110
  be                 000000
```

Copyright © 2017, Juniper Networks, Inc.

be1	**000001**
cs1	001000
cs2	010000
cs3	011000
cs4	100000
cs5	101000
cs6	110000
cs7	111000
ef	101110
ef1	**101111**
nc1	110000
nc2	111000

Meaning The code-point aliases are configured as expected. Notice that the custom aliases that you configure are added to the default code-point aliases.

Verifying the DSCP Classifier

Purpose Make sure that the DSCP classifier is configured as expected.

Action On Device R2, run the **show class-of-service classifiers name v4-ba-classifier** command.

```
user@R2> show class-of-service classifiers name v4-ba-classifier

Classifier: v4-ba-classifier, Code point type: dscp, Index: 10755
  Code point         Forwarding class                  Loss priority
  000000             BE-data                           high
  000001             BE-data                           low
  000010             BE-data                           low
  000011             BE-data                           low
  000100             BE-data                           low
  000101             BE-data                           low
  000110             BE-data                           low
  000111             BE-data                           low
  001000             BE-data                           low
  001001             BE-data                           low
  001010             Voice                             low
  001011             BE-data                           low
  001100             Voice                             high
  001101             BE-data                           low
  001110             Voice                             high
  001111             BE-data                           low
  010000             BE-data                           low
  010001             BE-data                           low
  010010             BE-data                           low
  010011             BE-data                           low
  010100             BE-data                           low
  010101             BE-data                           low
  010110             BE-data                           low
  010111             BE-data                           low
  011000             BE-data                           low
  011001             BE-data                           low
  011010             BE-data                           low
  011011             BE-data                           low
```


Copyright © 2017, Juniper Networks, Inc.

```
  011100            BE-data                       low
  011101            BE-data                       low
  011110            BE-data                       low
  011111            BE-data                       low
  100000            BE-data                       low
  100001            BE-data                       low
  100010            BE-data                       low
  100011            BE-data                       low
  100100            BE-data                       low
  100101            BE-data                       low
  100110            BE-data                       low
  100111            BE-data                       low
  101000            BE-data                       low
  101001            BE-data                       low
  101010            BE-data                       low
  101011            BE-data                       low
  101100            BE-data                       low
  101101            BE-data                       low
  101110            Premium-data                  high
  101111            Premium-data                  low
  110000            NC                            low
  110001            BE-data                       low
  110010            BE-data                       low
  110011            BE-data                       low
  110100            BE-data                       low
  110101            BE-data                       low
  110110            BE-data                       low
  110111            BE-data                       low
  111000            NC                            low
  111001            BE-data                       low
  111010            BE-data                       low
  111011            BE-data                       low
  111100            BE-data                       low
  111101            BE-data                       low
  111110            BE-data                       low
  111111            BE-data                       low
```

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

```
user@R2> show class-of-service classifier name v4-ba-classifier
Classifier: v4-ba-classifier, Code point type: dscp, Index: 10755
  Code point         Forwarding class              Loss priority
  000000             BE-data                       high
  000001             BE-data                       low
  101110             Premium-data                  high
  101111             Premium-data                  low
```

Verifying the Forwarding Classes and Output Queues

Purpose Make sure that the forwarding classes are configured as expected.

Copyright © 2017, Juniper Networks, Inc.

Action On Device R2, run the **show class-of-service forwarding-class** command.

```
user@R2> show class-of-service forwarding-class

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

Meaning The forwarding classes are configured as expected.

Verifying That the Classifier Is Applied to the Interfaces

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

Action On Device R2, run the **show class-of-service interface** command.

```
user@R2> show class-of-service interface ge-1/0/3

Physical interface: ge-1/0/3, Index: 144
Queues supported: 8, Queues in use: 4
  Scheduler map: <default>, Index: 2
  Congestion-notification: Disabled

  Logical interface: ge-1/0/3.0, Index: 333
Object                  Name                  Type                  Index
Classifier              v4-ba-classifier      dscp                  10755
```

```
user@R2> show class-of-service interface ge-1/0/9

Physical interface: ge-1/0/9, Index: 150
Queues supported: 8, Queues in use: 4
  Scheduler map: <default>, Index: 2
  Congestion-notification: Disabled

  Logical interface: ge-1/0/9.0, Index: 332
Object                  Name                  Type                  Index
Classifier              v4-ba-classifier      dscp                  10755
```

Meaning The interfaces are configured as expected.

Verifying Behavior Aggregate Classifiers

Purpose Verify that the behavior aggregate classifiers were configured properly on the device.

Copyright © 2017, Juniper Networks, Inc.

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

When you are using hping to set the DSCP code points in the IPv4 packet header, the type-of-service (ToS) hex value (in this case, BC) is required in the **--tos** option of the **hping** command.

If your binary-to-hex or binary-to-decimal conversion skills are rusty, you can use an online calculator, such as http://www.mathsisfun.com/binary-decimal-hexadecimal-converter.html .

NOTE: When you convert a binary DSCP code point value, be sure to add two extra zeros at the end. So instead of 101111, use 10111100. These 0 values (the 7th and 8th bits) are reserved and ignored, but if you do not include them in the conversion, your hex and decimal values will be incorrect.

Extended Ping Sent from Device R1

```
user@R1> ping 172.16.70.1 tos 188 rapid count 25

PING 172.16.70.1 (172.16.70.1): 56 data bytes
!!!!!!!!!!!!!!!!!!!!!!!!!
--- 172.16.70.1 ping statistics ---
25 packets transmitted, 25 packets received, 0% packet loss
round-trip min/avg/max/stddev = 0.404/0.483/1.395/0.207 ms
```

hping Sent from Host 1

```
root@host1> hping 172.16.70.1 --tos BC -c 25

HPING 172.16.70.1 (eth1 172.16.70.1): NO FLAGS are set, 40 headers + 0 data bytes
len=46 ip=172.16.70.1 ttl=61 DF id=0 sport=0 flags=RA seq=0 win=0 rtt=0.3 ms
len=46 ip=172.16.70.1 ttl=61 DF id=0 sport=0 flags=RA seq=1 win=0 rtt=0.6 ms
len=46 ip=172.16.70.1 ttl=61 DF id=0 sport=0 flags=RA seq=2 win=0 rtt=0.4 ms
len=46 ip=172.16.70.1 ttl=61 DF id=0 sport=0 flags=RA seq=3 win=0 rtt=0.4 ms
len=46 ip=172.16.70.1 ttl=61 DF id=0 sport=0 flags=RA seq=4 win=0 rtt=0.6 ms
len=46 ip=172.16.70.1 ttl=61 DF id=0 sport=0 flags=RA seq=5 win=0 rtt=0.3 ms
len=46 ip=172.16.70.1 ttl=61 DF id=0 sport=0 flags=RA seq=6 win=0 rtt=0.4 ms
len=46 ip=172.16.70.1 ttl=61 DF id=0 sport=0 flags=RA seq=7 win=0 rtt=0.4 ms
len=46 ip=172.16.70.1 ttl=61 DF id=0 sport=0 flags=RA seq=8 win=0 rtt=0.4 ms
len=46 ip=172.16.70.1 ttl=61 DF id=0 sport=0 flags=RA seq=9 win=0 rtt=0.4 ms
len=46 ip=172.16.70.1 ttl=61 DF id=0 sport=0 flags=RA seq=10 win=0 rtt=0.5 ms
len=46 ip=172.16.70.1 ttl=61 DF id=0 sport=0 flags=RA seq=11 win=0 rtt=0.4 ms
len=46 ip=172.16.70.1 ttl=61 DF id=0 sport=0 flags=RA seq=12 win=0 rtt=0.5 ms
len=46 ip=172.16.70.1 ttl=61 DF id=0 sport=0 flags=RA seq=13 win=0 rtt=0.4 ms
len=46 ip=172.16.70.1 ttl=61 DF id=0 sport=0 flags=RA seq=14 win=0 rtt=0.4 ms
len=46 ip=172.16.70.1 ttl=61 DF id=0 sport=0 flags=RA seq=15 win=0 rtt=0.4 ms
len=46 ip=172.16.70.1 ttl=61 DF id=0 sport=0 flags=RA seq=16 win=0 rtt=0.4 ms
len=46 ip=172.16.70.1 ttl=61 DF id=0 sport=0 flags=RA seq=17 win=0 rtt=0.5 ms
len=46 ip=172.16.70.1 ttl=61 DF id=0 sport=0 flags=RA seq=18 win=0 rtt=0.5 ms
len=46 ip=172.16.70.1 ttl=61 DF id=0 sport=0 flags=RA seq=19 win=0 rtt=0.4 ms
len=46 ip=172.16.70.1 ttl=61 DF id=0 sport=0 flags=RA seq=20 win=0 rtt=0.4 ms
len=46 ip=172.16.70.1 ttl=61 DF id=0 sport=0 flags=RA seq=21 win=0 rtt=0.5 ms
len=46 ip=172.16.70.1 ttl=61 DF id=0 sport=0 flags=RA seq=22 win=0 rtt=0.4 ms
len=46 ip=172.16.70.1 ttl=61 DF id=0 sport=0 flags=RA seq=23 win=0 rtt=0.5 ms
len=46 ip=172.16.70.1 ttl=61 DF id=0 sport=0 flags=RA seq=24 win=0 rtt=0.4 ms
```

Copyright © 2017, Juniper Networks, Inc.

On Device R2, Verify that Queue 2 is Incrementing.

Code point 101111 is associated with Premium-data, which uses queue 1.

```
user@R2> show interfaces extensive ge-1/0/3 | find "queue counters"


Queue counters:   Queued packets   Transmitted packets   Dropped packets
  0                           0                     0                 0
  1                          50                    50                 0
  2                           0                     0                 0
  3                          42                    42                 0
  Queue number:         Mapped forwarding classes
    0                   BE-data
    1                   Premium-data
    2                   Voice
    3                   NC
...
```

Meaning The output shows that queue 1 has incremented by 50 packets after sending 50 packets through the router.

Related Documentation

- *Interfaces Feature Guide for Security Devices*
- Classification Overview on page 13
- Sample Behavior Aggregate Classification on page 18
- Understanding Packet Loss Priorities on page 16

Copyright © 2017, Juniper Networks, Inc.

CHAPTER 3

Controlling Network Access with Traffic Policing

- Simple Filters and Policers Overview on page 29
- Two-Rate Three-Color Policer Overview on page 30
- Example: Configuring a Two-Rate Three-Color Policer on page 31
- Logical Interface (Aggregate) Policer Overview on page 36
- Two-Color Policer Configuration Overview on page 37
- Example: Configuring a Two-Color Logical Interface (Aggregate) Policer on page 40
- Guidelines for Configuring Simple Filters on page 46
- Example: Configuring and Applying a Firewall Filter for a Multifield Classifier on page 49

Simple Filters and Policers Overview

You can configure simple filters and policers to handle oversubscribed traffic in SRX1400, SRX3400, SRX3600, SRX5600 and SRX5800 devices. In Junos OS, policers can be configured as part of the firewall filter hierarchy. (Platform support depends on the Junos OS release in your installation.)

NOTE: For SRX5600 and SRX5800 devices, the simple filter or policing actions can be applied only to logical interfaces residing in an SRX5000 line Flex IOC (FIOC) because only an SRX5000 line FIOC supports the simple filter and policing features on the SRX5600 and SRX5800 devices.

The simple filter functionality consists of the following:

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

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

- To enforce traffic rates to conform to the service-level agreement (SLA)

Copyright © 2017, Juniper Networks, Inc.

- To protect next hops, such as protecting the central point and the SPU from being overwhelmed by excess traffic like DOS attacks

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

Related Documentation

- Guidelines for Configuring Simple Filters on page 46
- Example: Configuring a Two-Rate Three-Color Policer on page 31
- Example: Configuring and Applying a Firewall Filter for a Multifield Classifier on page 49

Two-Rate Three-Color Policer Overview

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

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

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

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

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

Copyright © 2017, Juniper Networks, Inc.

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

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

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

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

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

Related Documentation

- Example: Configuring a Two-Rate Three-Color Policer on page 31

Example: Configuring a Two-Rate Three-Color Policer

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

- Requirements on page 31
- Overview on page 31
- Configuration on page 32
- Verification on page 36

Requirements

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

Overview

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

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

Each category is associated with an action. For green traffic, packets are implicitly set with a loss-priority value of **low** and then transmitted. For yellow traffic, packets are implicitly set with a loss-priority value of **medium-high** and then transmitted. For red

Copyright © 2017, Juniper Networks, Inc.

traffic, packets are implicitly set with a loss-priority value of **high** and then transmitted. If the policer configuration includes the optional **action** statement (**action loss-priority high then discard**), then packets in a red flow are discarded instead.

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

Topology

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

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

Configuration

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

To configure this example, perform the following tasks:

- Configuring a Two-Rate Three-Color Policer on page 33
- Configuring an IPv4 Stateless Firewall Filter That References the Policer on page 34
- Applying the Filter to a Logical Interface at the Protocol Family Level on page 35

CLI Quick Configuration

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

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


Copyright © 2017, Juniper Networks, Inc.

Configuring a Two-Rate Three-Color Policer

Step-by-Step Procedure

To configure a two-rate three-color policer:

1. Enable configuration of a three-color policer.

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

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

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

3. Configure the two-rate guaranteed traffic limits.

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

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

4. Configure the two-rate peak traffic limits.

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

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

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

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

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

Results

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

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

Copyright © 2017, Juniper Networks, Inc.

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

Configuring an IPv4 Stateless Firewall Filter That References the Policer

Step-by-Step Procedure

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

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

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

2. Specify the filter term that references the policer.

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

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

Results

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

```
[edit]
user@host# show firewall
family inet {
  filter filter-trtcm1ca-all {
    term 1 {
      then {
        three-color-policer {
          two-rate trTCM1-ca;
        }
      }
    }
  }
}
three-color-policer trTCM1-ca {
  action {
    loss-priority high then discard;
  }
  two-rate {
    color-aware;
    committed-information-rate 40m;
    committed-burst-size 100k;
    peak-information-rate 60m;
    peak-burst-size 200k;
```


Copyright © 2017, Juniper Networks, Inc.

```
    }
  }
```

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

Step-by-Step Procedure

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

1. Enable configuration of an IPv4 firewall filter.

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

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

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

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

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

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

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

Results

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

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

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

Copyright © 2017, Juniper Networks, Inc.

Verification

Confirm that the configuration is working properly.

- Displaying the Firewall Filters Applied to the Logical Interface on page 36

Displaying the Firewall Filters Applied to the Logical Interface

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

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

```
user@host> show interfaces ge-2/0/5.0 detail
  Logical interface ge-2/0/5.0 (Index 105) (SNMP ifIndex 556) (Generation 170)
    Flags: Device-Down SNMP-Traps 0x4004000 Encapsulation: ENET2
    Traffic statistics:
     Input  bytes  :                   0
     Output bytes  :                   0
     Input  packets:                   0
     Output packets:                   0
    Local statistics:
     Input  bytes  :                   0
     Output bytes  :                   0
     Input  packets:                   0
     Output packets:                   0
    Transit statistics:
     Input  bytes  :                   0                   0 bps
     Output bytes  :                   0                   0 bps
     Input  packets:                   0                   0 pps
     Output packets:                   0                   0 pps
    Protocol inet, MTU: 1500, Generation: 242, Route table: 0
      Flags: Sendbcast-pkt-to-re
      Input Filters: filter-trtcm1ca-all
      Addresses, Flags: Dest-route-down Is-Preferred Is-Primary
        Destination: 10.20.130/24, Local: 10.20.130.1, Broadcast: 10.20.130.255,

        Generation: 171
    Protocol multiservice, MTU: Unlimited, Generation: 243, Route table: 0
      Policer: Input: __default_arp_policer__
```

Related Documentation

- Two-Rate Three-Color Policer Overview on page 30

Logical Interface (Aggregate) Policer Overview

A *logical interface policer*—also called an *aggregate policer*—is a two-color or three-color policer that defines traffic rate limiting. Starting in Junos OS Release 15.1X49-D40, you can apply a policer to input or output traffic for multiple protocol families on the same logical interface without needing to create multiple instances of the policer.

Copyright © 2017, Juniper Networks, Inc.

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

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

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

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

Release History Table

Release	Description
15.1X49-D40	Starting in Junos OS Release 15.1X49-D40, you can apply a policer to input or output traffic for multiple protocol families on the same logical interface without needing to create multiple instances of the policer.

Related Documentation

- Two-Color Policer Configuration Overview on page 37
- Example: Configuring a Two-Color Logical Interface (Aggregate) Policer on page 40
- **logical-interface-policer on page 261**

Two-Color Policer Configuration Overview

Starting in Junos OS Release 15.1X49-D40, you can configure and apply single-rate two-color policers to Layer 3 traffic. Table 11 on page 37 describes the hierarchy levels at which you can configure and apply them.

Table 11: Two-Color Policer Configuration and Application Overview

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

Basic policer configuration:
[edit firewall]
policer *policer-name* {
if-exceeding {
bandwidth-limit *bps*;
burst-size-limit *bytes*;
}
then {
discard;
forwarding-class *class-name*;
loss-priority *supported-value*;
	Method A—Apply as an interface policer at the protocol family level:
	[edit interfaces]
	interface-name {
	unit *unit-number* {
	family *family-name* {
	policer {
	input *policer-name*;
	output *policer-name*;
	}
	}
		Policer configuration:
		• Use **bandwidth-limit** *bps* to specify an absolute value.
		Firewall filter configuration (*)
		• If applying to multiple interfaces, include the **interface-specific** statement to create unique policers and counters for each interface.

Copyright © 2017, Juniper Networks, Inc.

Table 11: Two-Color Policer Configuration and Application Overview *(continued)*

Policer Configuration	Layer 3 Application	Key Points

}
}
	}
	}
	
	Method B—Apply as a firewall filter policer at the protocol family level:
	
	[edit firewall]
	family *family-name* {
	filter *filter-name* {
	interface-specific; # (*)
	from {
	... *match-conditions* ...
	}
	then {
	policer *policer-name*;
	}
	}
	}
	
	[edit interfaces]
	interface-name {
	unit *unit-number* {
	family *family-name* {
	filter {
	input *filter-name*;
	output *filter-name*;
	}
	... *protocol-configuration* ...
	}
	}
	}
		Interface policer verification:
		
		• Use the **show interfaces (detail | extensive)** operational mode command.
		• Use the **show policer** operational mode command.
		
		Firewall filter policer verification:
		
		• Use the **show interfaces (detail | extensive)** operational mode command.
		• Use the **show firewall filter *filter-name*** operational mode command.

Copyright © 2017, Juniper Networks, Inc.

Table 11: Two-Color Policer Configuration and Application Overview *(continued)*

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

Logical interface policer configuration:
[edit firewall]
policer *policer-name* {
logical-interface-policer;
if-exceeding {
bandwidth-limit *bps*;
burst-size-limit *bytes*;
}
then {
discard;
forwarding-class *class-name*;
loss-priority *supported-value*;
}
}
	Method A—Apply as an interface policer only:
	[edit interfaces]
	interface-name {
	unit *unit-number* {
	policer { # All protocols
	input *policer-name*;
	output *policer-name*;
	}
	family *family-name* {
	policer { # One protocol
	input *policer-name*;
	output *policer-name*;
	}
	}
	}
	}
	Method B—Apply as a firewall filter policer at the protocol family level:
	[edit firewall]
	family *family-name* {
	filter *filter-name* {
	interface-specific;
	term *term-name*{
	from {
	... *match-conditions* ...
	}
	}
	then {
	policer *policer-name*;
	}
	}
	}
		Policer configuration:
		• Include the **logical-interface-policer** statement.
		Two options for interface policer application:
		• To rate-limit all traffic types, regardless of the protocol family, apply the logical interface policer at the logical unit level.
		• To rate-limit traffic of a specific protocol family, apply the logical interface policer at the protocol family level.
		Interface policer verification:
		• Use the **show interfaces (detail | extensive)** operational mode command.
		• Use the **show policer** operational mode command.

Release History Table

Release	Description
15.1X49-D40	Starting in Junos OS Release 15.1X49-D40, you can configure and apply single-rate two-color policers to Layer 3 traffic.

Related Documentation

- logical-interface-policer on page 261
- Example: Configuring a Two-Color Logical Interface (Aggregate) Policer on page 40

Copyright © 2017, Juniper Networks, Inc.

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

Starting in Junos OS Release 15.1X49-D40, you can configure a single-rate two-color policer as a logical interface policer and apply it to incoming IPv4 traffic on a logical interface. This example shows how to do to so.

- Requirements on page 40
- Overview on page 40
- Configuration on page 40
- Verification on page 44

Requirements

Before you begin, make sure that the logical interface to which you apply the two-color logical interface policer is hosted on a Gigabit Ethernet interface (**ge-**) or a 10-Gigabit Ethernet interface (**xe-**).

Overview

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

Topology

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

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

Configuration

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

To configure this example, perform the following tasks:

- Configuring the Logical Interfaces on page 41
- Configuring the Single-Rate Two-Color Policer as a Logical Interface Policer on page 42
- Applying the Logical Interface Policer to Input IPv4 Traffic at a Logical Interface on page 43

Copyright © 2017, Juniper Networks, Inc.

CLI Quick Configuration

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

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

Configuring the Logical Interfaces

Step-by-Step Procedure

To configure the logical interfaces:

1. Enable configuration of the interface.

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

2. Configure single tagging.

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

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

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

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

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

Results

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

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

Copyright © 2017, Juniper Networks, Inc.

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

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

Step-by-Step Procedure

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

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

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

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

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

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

3. Specify the policer traffic limits.

   a. Specify the bandwidth limit.

      - To specify the bandwidth limit as an absolute rate, from 8,000 bits per second through 50,000,000,000 bits per second, include the **bandwidth-limit *bps*** statement.
      - To specify the bandwidth limit as a percentage of the physical port speed on the interface, include the **bandwidth-percent *percent*** statement.

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

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


Copyright © 2017, Juniper Networks, Inc.

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

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

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

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

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

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

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

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

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

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

1. Enable configuration of the logical interface.

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

Copyright © 2017, Juniper Networks, Inc.

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

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

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

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

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

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

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

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

Verification

Confirm that the configuration is working properly.

- Displaying Traffic Statistics and Policers for the Logical Interface on page 45
- Displaying Statistics for the Policer on page 45

 Copyright © 2017, Juniper Networks, Inc.

Displaying Traffic Statistics and Policers for the Logical Interface

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

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

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

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

Displaying Statistics for the Policer

Purpose Verify the number of packets evaluated by the policer.

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

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

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

Release History Table

Release	Description
15.1X49-D40	Starting in Junos OS Release 15.1X49-D40, you can configure a single-rate two-color policer as a logical interface policer and apply it to incoming IPv4 traffic on a logical interface.

Related Documentation

- logical-interface-policer on page 261
- Two-Color Policer Configuration Overview on page 37

Copyright © 2017, Juniper Networks, Inc.

Guidelines for Configuring Simple Filters

This topic covers the following information:

- Statement Hierarchy for Configuring Simple Filters on page 46
- Simple Filter Protocol Families on page 46
- Simple Filter Names on page 47
- Simple Filter Terms on page 47
- Simple Filter Match Conditions on page 47
- Simple Filter Terminating Actions on page 48
- Simple Filter Nonterminating Actions on page 49

Statement Hierarchy for Configuring Simple Filters

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

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

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

Simple Filter Protocol Families

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

NOTE: You can apply simple filters to the family inet only, and only in the input direction. Because of hardware limitations on the SRX1400, SRX3400, SRX3600, SRX5600 and SRX5800 devices, a maximum of 400 logical input interfaces and 2000 terms (in one Broadcom packet processor) can be applied with simple filters. (Platform support depends on the Junos OS release in your installation.)

Copyright © 2017, Juniper Networks, Inc.

Simple Filter Names

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

Simple Filter Terms

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

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

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

NOTE: In one Broadcom packet processor, a maximum of 2000 terms can be applied with simple filters on the SRX1400, SRX3400, SRX3600, SRX5600, SRX5600 and SRX5800 devices. (Platform support depends on the Junos OS release in your installation.)

Simple Filter Match Conditions

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

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

- On MX Series routers with the Enhanced Queuing DPC, simple filters do *not* support the **forwarding- class** match condition.
- Simple filters support only one **source-address** and one **destination-address** prefix for each filter term. If you configure multiple prefixes, only the last one is used.
- Simple filters do *not* support multiple source addresses and destination addresses in a single term. If you configure multiple addresses, only the last one is used.
- Simple filters do *not* support negated match conditions, such as the **protocol-except** match condition or the **exception** keyword.

Copyright © 2017, Juniper Networks, Inc.

- Simple filters support a range of values for **source-port** and **destination-port** match conditions only. For example, you can configure **source-port 400-500** or **destination-port 600-700**.
- Simple filters do *not* support noncontiguous mask values.

Table 12 on page 48 lists the simple filter match conditions.

Table 12: Simple Filter Match Conditions

Match Condition	Description
destination-address *destination-address*	Match IP destination address.
destination-port *number*	TCP or UDP destination port field. If you configure this match condition, we recommend that you also configure the **protocol** match statement to determine which protocol is being used on the port. In place of the numeric value, you can specify one of the following text aliases (the port numbers are also listed): **afs** (1483), **bgp** (179), **biff** (512), **bootpc** (68), **bootps** (67), **cmd** (514), **cvspserver** (2401), **dhcp** (67), **domain** (53), **eklogin** (2105), **ekshell** (2106), **exec** (512), **finger** (79), **ftp** (21), **ftp-data** (20), **http** (80), **https** (443), **ident** (113), **imap** (143), **kerberos-sec** (88), **klogin** (543), **kpasswd** (761), **krb-prop** (754), **krbupdate** (760), **kshell** (544), **ldap** (389), **login** (513), **mobileip-agent** (434), **mobilip-mn** (435), **msdp** (639), **netbios-dgm** (138), **netbios-ns** (137), **netbios-ssn** (139), **nfsd** (2049), **nntp** (119), **ntalk** (518), **ntp** (123), **pop3** (110), **pptp** (1723), **printer** (515), **radacct** (1813), **radius** (1812), **rip** (520), **rkinit** (2108), **smtp** (25), **snmp** (161), **snmptrap** (162), **snpp** (444), **socks** (1080), **ssh** (22), **sunrpc** (111), **syslog** (514), **tacacs-ds** (65), **talk** (517), **telnet** (23), **tftp** (69), **timed** (525), **who** (513), or **xdmcp** (177).
forwarding-class *class*	Match the forwarding class of the packet. Specify **assured-forwarding**, **best-effort**, **expedited-forwarding**, or **network-control**.
protocol *number*	IP protocol field. In place of the numeric value, you can specify one of the following text aliases (the field values are also listed): **ah** (51), **dstopts** (60), **egp** (8), **esp** (50), **fragment** (44), **gre** (47), **hop-by-hop** (0), **icmp** (1), **icmp6** (58), **icmpv6** (58), **igmp** (2), **ipip** (4), **ipv6** (41), **ospf** (89), **pim** (103), **rsvp** (46), **sctp** (132), **tcp** (6), **udp** (17), or **vrrp** (112).
source-address *ip-source-address*	Match the IP source address.
source-port *number*	Match the UDP or TCP source port field. If you configure this match condition, we recommend that you also configure the **protocol** match statement to determine which protocol is being used on the port. In place of the numeric field, you can specify one of the text aliases listed for **destination-port**.

Simple Filter Terminating Actions

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

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

Copyright © 2017, Juniper Networks, Inc.

Simple Filter Nonterminating Actions

Simple filters support only the following nonterminating actions:

- **forwarding-class (*forwarding-class* | assured-forwarding |best-effort | expedited-forwarding | network-control)**

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

- **loss-priority (high | low | medium-high | medium-low)**

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

Related Documentation

- Simple Filters and Policers Overview on page 29

Example: Configuring and Applying a Firewall Filter for a Multifield Classifier

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

- Requirements on page 49
- Overview on page 49
- Configuration on page 51
- Verification on page 53

Requirements

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

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

Overview

A classifier is a software operation that inspects a packet as it enters the router or switch. The packet header contents are examined, and this examination determines how the packet is treated when the network becomes too busy to handle all of the packets and you want your devices to drop packets intelligently, instead of dropping packets indiscriminately. One common way to detect packets of interest is by source port number.

Copyright © 2017, Juniper Networks, Inc.

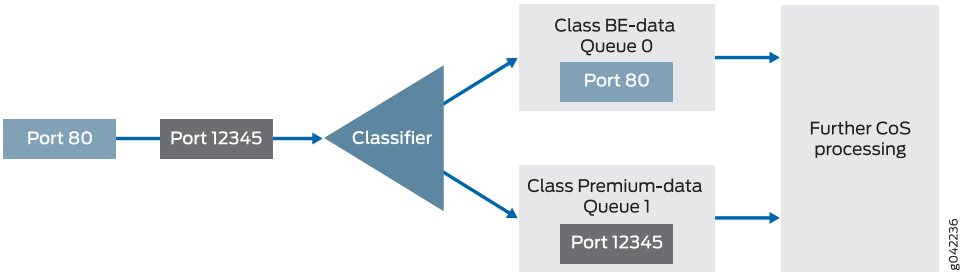
The TCP port numbers 80 and 12345 are used in this example, but many other matching criteria for packet detection are available to multifield classifiers, using firewall filter match conditions. The configuration in this example specifies that TCP packets with source port 80 are classified into the BE-data forwarding class and queue number 0. TCP packets with source port 12345 are classified into the Premium-data forwarding class and queue number 1.

Multifield classifiers are typically used at the network edge as packets enter an autonomous system (AS).

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

The classifier operation is shown in Figure 4 on page 50.

Figure 4: Multifield Classifier Based on TCP Source Ports

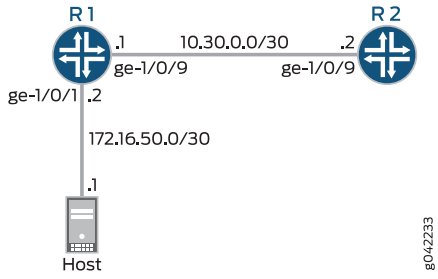


You apply the multifield classifier's firewall filter as an input filter on each customer-facing or host-facing interface that needs the filter. The incoming interface is ge-1/0/0 on Device R1. The classification and queue assignment is verified on the outgoing interface. The outgoing interface is Device R1's ge-1/0/2 interface.

Topology

Figure 5 on page 50 shows the sample network.

Figure 5: Multifield Classifier Scenario



"CLI Quick Configuration" on page 51 shows the configuration for all of the Juniper Networks devices in Figure 5 on page 50.

The section "Step-by-Step Procedure" on page 51 describes the steps on Device R1.

Copyright © 2017, Juniper Networks, Inc.

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

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

Configuration

CLI Quick Configuration

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

Device R1

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

Device R2

```
set interfaces ge-1/0/2 description to-R1
set interfaces ge-1/0/2 unit 0 family inet address 10.30.0.2/30
```

Step-by-Step Procedure

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

To configure Device R1:

1. Configure the device interfaces.

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

   user@R1# set ge-1/0/2 description to-R2
   user@R1# set ge-1/0/2 unit 0 family inet address 10.30.0.1/30
   ```

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

   ```
   [edit class-of-service forwarding-classes]
   ```

Copyright © 2017, Juniper Networks, Inc.

```
user@R1# set BE-data queue-num 0
user@R1# set Premium-data queue-num 1
user@R1# set Voice queue-num 2
user@R1# set NC queue-num 3
```

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

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

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

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

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

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

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

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

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

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

```
user@R1# show interfaces
ge-1/0/0 {
   description to-host;
   unit 0 {
      family inet {
         filter {
            input mf-classifier;
         }
         address 172.16.50.2/30;
      }
   }
}
ge-1/0/2 {
   description to-R2;
```

 Copyright © 2017, Juniper Networks, Inc.

```
    unit 0 {
      family inet {
        address 10.30.0.1/30;
      }
    }
  }
```

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

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

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

Verification

Confirm that the configuration is working properly.

- Checking the CoS Settings on page 53
- Sending TCP Traffic into the Network and Monitoring the Queue Placement on page 54

Checking the CoS Settings

Purpose Confirm that the forwarding classes are configured correctly.

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

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

Copyright © 2017, Juniper Networks, Inc.

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

Meaning The output shows the configured custom classifier settings.

Sending TCP Traffic into the Network and Monitoring the Queue Placement

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

Action

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

   ```
   user@R1> clear interfaces statistics ge-1/0/2
   ```

2. Use a traffic generator to send 50 TCP port 80 packets to Device R2 or to some other downstream device.

3. On Device R1, check the queue counters.

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

   ```
   user@R1> show interfaces extensive ge-1/0/2 | find "Queue counters"
     Queue counters:      Queued packets  Transmitted packets     Dropped packets

       0                                50                    50
   0
       1                                 0                    57
   0
       2                                 0                     0
   0
       3                                 0                     0
   0
   ```

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

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

5. On Device R1, check the queue counters.

   ```
   user@R1> show interfaces extensive ge-1/0/2 | find "Queue counters"
     Queue counters:      Queued packets  Transmitted packets     Dropped packets
   ```


Copyright © 2017, Juniper Networks, Inc.

0		50	50
0			
1		**50**	57
0			
2		0	0
0			
3		0	0
0			

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

Related Documentation

- Example: Configuring a Two-Rate Three-Color Policer on page 31

Copyright © 2017, Juniper Networks, Inc.

Copyright © 2017, Juniper Networks, Inc.

CHAPTER 4

Controlling Output Queues with Forwarding Classes

- Forwarding Classes Overview on page 57
- Example: Configuring Forwarding Classes on page 59
- Example: Assigning Forwarding Classes to Output Queues on page 64
- Example: Assigning a Forwarding Class to an Interface on page 66
- Understanding the SPC High-Priority Queue on page 67
- Example: Configuring the SPC High-Priority Queue on page 68
- Understanding Queuing and Marking of Host Outbound Traffic on page 70
- Default Routing Engine Protocol Queue Assignments on page 72

Forwarding Classes Overview

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

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

- Expedited forwarding (EF)—Provides a low-loss, low-latency, low-jitter, assured-bandwidth, end-to-end service.
- Assured forwarding (AF)—Provides a group of values you can define and includes four subclasses—AF1, AF2, AF3, and AF4—each with three drop probabilities (low, medium, and high).
- Best effort (BE)—Provides no service profile. For the BE forwarding class, loss priority is typically not carried in a class-of-service (CoS) value, and random early detection (RED) drop profiles are more aggressive.
- Network Control (NC)—This class is typically high priority because it supports protocol control.

In addition to behavior aggregate (BA) and mulitfield (MF) classification, the forwarding class (FC) of a packet can be directly determined by the logical interface that receives

Copyright © 2017, Juniper Networks, Inc.

the packet. The packet FC can be configured using CLI commands, and if configured, this FC overrides the FC from any BA classification that was previously configured on the logical interface.

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

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

This section contains the following topics:

- Forwarding Class Queue Assignments on page 58
- Forwarding Policy Options on page 59

Forwarding Class Queue Assignments

Juniper Networks devices have eight queues built into the hardware. By default, four queues are assigned to four FCs. Table 13 on page 58 shows the four default FCs and queues that Juniper Networks classifiers assign to packets, based on the class-of-service (CoS) values in the arriving packet headers.

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

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

Table 13: Default Forwarding Class Queue Assignments

Forwarding Queue	Forwarding Class	Forwarding Class Description
Queue 0	best-effort (BE)	The Juniper Networks device does not apply any special CoS handling to packets with 000000 in the DiffServ field, a backward compatibility feature. These packets are usually dropped under congested network conditions.
Queue 1	expedited-forwarding (EF)	The Juniper Networks device delivers assured bandwidth, low loss, low delay, and low delay variation (jitter) end-to-end for packets in this service class. Devices accept excess traffic in this class, but in contrast to assured forwarding, out-of-profile expedited-forwarding packets can be forwarded out of sequence or dropped.

Copyright © 2017, Juniper Networks, Inc.

Table 13: Default Forwarding Class Queue Assignments *(continued)*

Forwarding Queue	Forwarding Class	Forwarding Class Description
Queue 2	assured-forwarding (AF)	The Juniper Networks device offers a high level of assurance that the packets are delivered as long as the packet flow from the customer stays within a certain service profile that you define. The device accepts excess traffic, but applies a random early detection (RED) drop profile to determine whether the excess packets are dropped and not forwarded. Three drop probabilities (low, medium, and high) are defined for this service class.
Queue 3	network-control (NC)	The Juniper Networks device delivers packets in this service class with a low priority. (These packets are not delay sensitive.) Typically, these packets represent routing protocol hello or keepalive messages. Because loss of these packets jeopardizes proper network operation, delay is preferable to discard.

Forwarding Policy Options

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

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

Related Documentation

- Example: Assigning Forwarding Classes to Output Queues on page 64
- Example: Assigning a Forwarding Class to an Interface on page 66
- Example: Configuring Forwarding Classes on page 59

Example: Configuring Forwarding Classes

By default on all platforms, four output queues are mapped to four FCs as shown in "Forwarding Classes Overview" on page 57. On Juniper Networks devices, you can configure up to eight FCs and eight queues.

Copyright © 2017, Juniper Networks, Inc.

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

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

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

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

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

Defining Eight Classifiers

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


Copyright © 2017, Juniper Networks, Inc.

```
    }
  }
}
```

Adding Eight Schedulers to a Scheduler Map

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

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

Copyright © 2017, Juniper Networks, Inc.

```
    priority low;
    drop-profile-map loss-priority any protocol both drop-default;
  }
  Q7 {
    transmit-rate percent 5;
    priority high;
    drop-profile-map loss-priority any protocol both drop-default;
  }
}
```

Configuring an IP Precedence Classifier and Rewrite Tables

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


Copyright © 2017, Juniper Networks, Inc.

```
    forwarding-class af12 {
      loss-priority high code-point 101;
    }
    forwarding-class ef1 {
      loss-priority low code-point 110;
    }
    forwarding-class nc2 {
      loss-priority low code-point 111;
    }
  }
  inet-precedence inet-rw-table {
    forwarding-class be {
      loss-priority low code-point 000;
    }
    forwarding-class af11 {
      loss-priority high code-point 001;
    }
    forwarding-class ef1 {
      loss-priority low code-point 010;
    }
    forwarding-class nc1 {
      loss-priority low code-point 111;
    }
    forwarding-class be1 {
      loss-priority low code-point 100;
    }
    forwarding-class af12 {
      loss-priority high code-point 101;
    }
    forwarding-class ef1 {
      loss-priority low code-point 111;
    }
    forwarding-class nc2 {
      loss-priority low code-point 110;
    }
  }
```

Configuring an IDP Policy with a Forwarding Class

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

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

Related Documentation

- Forwarding Classes Overview on page 57

Copyright © 2017, Juniper Networks, Inc.

- Example: Assigning Forwarding Classes to Output Queues on page 64
- Example: Assigning a Forwarding Class to an Interface on page 66

Example: Assigning Forwarding Classes to Output Queues

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

- Requirements on page 64
- Overview on page 64
- Configuration on page 64
- Verification on page 66

Requirements

Before you begin, determine the MF classifier. See "Example: Configuring and Applying a Firewall Filter for a Multifield Classifier" on page 49.

Overview

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

You must assign the forwarding classes established by the MF classifier to output queues. Table 14 on page 64 shows how this example assigns output queues.

Table 14: Sample Output Queue Assignments for mf-classifier Forwarding Queues

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

Configuration

CLI Quick Configuration

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

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


Copyright © 2017, Juniper Networks, Inc.

Step-by-Step Procedure

The following example requires you to navigate various levels in the configuration hierarchy. For instructions on how to do that, see *Using the CLI Editor in Configuration Mode*.

To assign forwarding classes to output queues:

1. Configure class of service.

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

2. Assign best-effort traffic to queue 0.

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

3. Assign expedited forwarding traffic to queue 1.

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

4. Assign assured forwarding traffic to queue 2.

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

5. Assign network control traffic to queue 3.

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

Results

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

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

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

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

Copyright © 2017, Juniper Networks, Inc.

Verification

Verifying Forwarding Classes Are Assigned to Output Queues

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

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

Related Documentation

- Forwarding Classes Overview on page 57
- Example: Assigning a Forwarding Class to an Interface on page 66
- Example: Configuring Forwarding Classes on page 59

Example: Assigning a Forwarding Class to an Interface

This example shows how to assign a forwarding class to an interface.

- Requirements on page 66
- Overview on page 66
- Configuration on page 66
- Verification on page 67

Requirements

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

Overview

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

In this example, you configure class of service, create interface ge-3/0/0 unit 0 and then set the forwarding class to assured-forwarding.

All packets coming into the device from the ge-3/0/0 unit 0 interface are assigned to the assured-forwarding forwarding class.

Configuration

Step-by-Step Procedure The following example requires you to navigate various levels in the configuration hierarchy. For instructions on how to do that, see *Using the CLI Editor in Configuration Mode* in the *CLI User Guide*.

To assign a forwarding class to an interface:

1. Configure class of service and assign the interface.

Copyright © 2017, Juniper Networks, Inc.

```
[edit]
user@host# edit class-of-service interfaces ge-3/0/0 unit 0
```

2. Specify the forwarding class.

```
[edit class-of-service interfaces ge-3/0/0 unit 0]
user@host# set forwarding-class assured-forwarding
```

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

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

Verification

To verify the configuration is working properly, enter the **show class-of-service** command.

Related Documentation

- Forwarding Classes Overview on page 57
- Example: Assigning Forwarding Classes to Output Queues on page 64

Understanding the SPC High-Priority Queue

The Services Processing Card (SPC) on SRX1400, SRX3000 line, and SRX5000 line devices provides processing power to run integrated services such as firewall, IPsec, and IDP. All traffic traversing the SRX Series device is passed to an SPC to have service processing applied. Junos OS provides a configuration option to enable packets with specific Differentiated Services (DiffServ) code points (DSCP) precedence bits to enter a high-priority queue on the SPC. The Services Processing Unit (SPU) draws packets from the high-priority queue and only draws packets from a low-priority queue when the high-priority queue is empty. This feature can reduce overall latency for real-time traffic, such as voice traffic.

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

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

Related Documentation

- Example: Configuring the SPC High-Priority Queue on page 68
- Forwarding Classes Overview on page 57

Copyright © 2017, Juniper Networks, Inc.

Example: Configuring the SPC High-Priority Queue

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

- Requirements on page 68
- Overview on page 68
- Configuration on page 68
- Verification on page 69

Requirements

This example uses the following hardware and software components:

- SRX5000 line device
- Junos OS Release 11.4R2 or later

Overview

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

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

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

Configuration

CLI Quick Configuration

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

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


Copyright © 2017, Juniper Networks, Inc.

Step-by-Step Procedure

The following example requires you to navigate various levels in the configuration hierarchy. For instructions on how to do that, see *Using the CLI Editor in Configuration Mode* in the *CLI User Guide*.

To configure the high-priority queue on the SPC:

1. Define forwarding classes and assign queue numbers.

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

2. Configure the SPC high-priority queue.

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

Results

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

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

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

Verification

Verifying SPU High-Priority Queue Mapping

Purpose

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

Action

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

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

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

Copyright © 2017, Juniper Networks, Inc.

Related Documentation

- Understanding the SPC High-Priority Queue on page 67

Understanding Queuing and Marking of Host Outbound Traffic

This topic covers the following information:

- Host Outbound Traffic Overview on page 70
- Default Queuing and Marking of Host Outbound Traffic on page 70
- Configured Queuing and Marking of Host Outbound Traffic on page 71
- Configured Queuing and Marking of Outbound Routing Engine Traffic Only on page 71

Host Outbound Traffic Overview

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

Routing Engine Sourced Traffic

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

Distributed Protocol Handler Traffic

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

NOTE: **For interfaces on MX80 routers, LACP control traffic is sent through the Routing Engine rather than through the Packet Forwarding Engine.**

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

Default Queuing and Marking of Host Outbound Traffic

By default, the router assigns host outbound traffic to the **best-effort** forwarding class (which maps to queue 0) or to the **network-control** forwarding class (which maps to

Copyright © 2017, Juniper Networks, Inc.

queue 3) based on protocol. For more information, see "Default Routing Engine Protocol Queue Assignments" on page 72.

By default, the router marks the type of service (ToS) field of Layer 3 packets in the host outbound traffic flow with DifffServ code point (DSCP) bits 000000 (which correlate with the **best-effort** forwarding class). The router does not remark Layer 2 traffic such as LACP control traffic on aggregated Ethernet. For more information, see *Default DSCP and DSCP IPv6 Classifiers*.

Configured Queuing and Marking of Host Outbound Traffic

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

- Packets generated by the Routing Engine
- Distributed protocol handler traffic for egress interfaces hosted on MX Series routers, M120 routers, and Enhanced III FPCs in M320 routers.

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

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

Configured Queuing and Marking of Outbound Routing Engine Traffic Only

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

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

Related Documentation

- Default Routing Engine Protocol Queue Assignments on page 72
- *Default DSCP and DSCP IPv6 Classifiers*
- *Example: Configuring Different Queuing and Marking Defaults for Outbound Routing Engine and Distributed Protocol Handler Traffic*

Copyright © 2017, Juniper Networks, Inc.

Default Routing Engine Protocol Queue Assignments

Table 15 on page 72 lists the default output queues to which Routing Engine sourced traffic is mapped by protocol type. In general, control protocol packets are sent over queue 3 and management traffic is sent over queue 0. The following caveats apply to these default queue assignments:

- For all packets sent to queue 3 over a VLAN-tagged interface, the software sets the 802.1p bit to 110, except for VRRP packets, in which case the software sets the 802.1p bit to 111.
- For IPv4 and IPv6 packets, the software copies the IP type-of-service (ToS) value into the 802.1p field independently of which queue the packets are sent out.
- For MPLS packets, the software copies the EXP bits into the 802.1p field.

Table 15: Default Queue Assignments for Packets Generated by the Routing Engine

Routing Engine Protocol	Default Queue Assignment
Adaptive Services PIC TCP tickle (keepalive packets for idle session generated with stateful firewall to probe idle TCP sessions)	Queue 0
Address Resolution Protocol (ARP)	Queue 0
ATM Operation, Administration, and Maintenance (OAM)	Queue 3
Bidirectional Forwarding Detection (BFD) Protocol	Queue 3
BGP	Queue 0
BGP TCP Retransmission	Queue 3
Cisco High-Level Data Link Control (HDLC)	Queue 3
Distance Vector Multicast Routing Protocol (DVMRP)	Queue 3
Ethernet Operation, Administration, and Maintenance (OAM)	Queue 3
Frame Relay Local Management Interface (LMI)	Queue 3
Frame Relay Asynchronization permanent virtual circuit (PVC)/data link connection identifier (DLCI) status messages	Queue 3
FTP	Queue 0
IS-IS Open Systems Interconnection (OSI)	Queue 3
Internet Control Message Protocol (ICMP)	Queue 0

Copyright © 2017, Juniper Networks, Inc.

Table 15: Default Queue Assignments for Packets Generated by the Routing Engine *(continued)*

Routing Engine Protocol	Default Queue Assignment
Internet Group Management Protocol (IGMP) query	Queue 3
IGMP Report	Queue 0
IP version 6 (IPv6) Neighbor Solicitation	Queue 3
IPv6 Neighbor Advertisement	Queue 3
IPv6 Router Advertisement	Queue 0
Label Distribution Protocol (LDP) User Datagram Protocol (UDP) hello	Queue 3
LDP keepalive and Session data	Queue 0
LDP TCP Retransmission	Queue 3
Link Aggregation Control Protocol (LACP)	Queue 3
Link Services (LS) PIC	If link fragmentation and interleaving (LFI) is enabled, all routing protocol packets larger than 128 bytes are transmitted from queue 0. This ensures that VoIP traffic is not affected. Fragmentation is supported on queue 0 only.
Multicast listener discovery (MLD)	Queue 0
Multicast Source Discovery Protocol (MSDP)	Queue 0
MSDP TCP Retransmission	Queue 3
Multilink Frame Relay Link Integrity Protocol (LIP)	Queue 3
NETCONF	Queue 0
NetFlow	Queue 0
OSPF protocol data unit (PDU)	Queue 3
Point-to-Point Protocol (PPP)	Queue 3
Protocol Independent Multicast (PIM)	Queue 3
Real-time performance monitoring (RPM) probe packets	Queue 3
RSVP	Queue 3
Routing Information Protocol (RIP)	Queue 3

Copyright © 2017, Juniper Networks, Inc.

Table 15: Default Queue Assignments for Packets Generated by the Routing Engine *(continued)*

Routing Engine Protocol	Default Queue Assignment
SNMP	Queue 0
SSH	Queue 0
sFlow monitoring technology	Queue 0
Telnet	Queue 0
Virtual Router Redundancy Protocol (VRRP)	Queue 3
xnm-clear-text	Queue 0
xnm-ssl	Queue 0

Copyright © 2017, Juniper Networks, Inc.

CHAPTER 5

Altering Outgoing Packets Headers with Rewrite Rules

- Rewrite Rules Overview on page 75
- Rewriting Frame Relay Headers on page 75
- Example: Configuring and Applying Rewrite Rules on page 77

Rewrite Rules Overview

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

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

NOTE: You can configure up to 32 IEEE 802.1p rewrite rules on each SRX5K-MPC on the SRX5600 and SRX5800 devices.

Related Documentation

- Example: Configuring and Applying Rewrite Rules on page 77

Rewriting Frame Relay Headers

- Assigning the Default Frame Relay Rewrite Rule to an Interface on page 76
- Defining a Custom Frame Relay Rewrite Rule on page 76

Copyright © 2017, Juniper Networks, Inc.

Assigning the Default Frame Relay Rewrite Rule to an Interface

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

The default Frame Relay rewrite rule contains the following settings:

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

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

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

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

Defining a Custom Frame Relay Rewrite Rule

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

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

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

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

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


Copyright © 2017, Juniper Networks, Inc.

Related Documentation

- Rewrite Rules Overview on page 75
- Example: Configuring and Applying Rewrite Rules on page 77

Example: Configuring and Applying Rewrite Rules

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

- Requirements on page 77
- Overview on page 77
- Configuration on page 78
- Verification on page 80

Requirements

Before you begin, create and configure the forwarding classes.

Overview

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

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

NOTE: **You can apply one rewrite rule to each logical interface.**

Table 16 on page 77 shows how the rewrite rules replace the DSCPs on packets in the four forwarding classes.

Table 16: Sample rewrite-dscps Rewrite Rules to Replace DSCPs

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

Copyright © 2017, Juniper Networks, Inc.

Table 16: Sample rewrite-dscps Rewrite Rules to Replace DSCPs *(continued)*

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

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

Configuration

- [xref target has no title]

CLI Quick Configuration

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

```
set class-of-service rewrite-rules dscp rewrite-dscps forwarding-class be-class
    loss-priority low code-point 000000
set class-of-service rewrite-rules dscp rewrite-dscps forwarding-class be-class
    loss-priority high code-point 000001
set class-of-service rewrite-rules dscp rewrite-dscps forwarding-class ef-class loss-priority
    low code-point 101110
set class-of-service rewrite-rules dscp rewrite-dscps forwarding-class ef-class loss-priority
    high code-point 101111
set class-of-service rewrite-rules dscp rewrite-dscps forwarding-class af-class loss-priority
    low code-point 001010
set class-of-service rewrite-rules dscp rewrite-dscps forwarding-class af-class loss-priority
    high code-point 001100
set class-of-service rewrite-rules dscp rewrite-dscps forwarding-class nc-class loss-priority
    low code-point 110000
set class-of-service rewrite-rules dscp rewrite-dscps forwarding-class nc-class loss-priority
    high code-point 110001
set class-of-service interfaces irb unit 0 rewrite-rules dscp rewrite-dscps
```


Copyright © 2017, Juniper Networks, Inc.

Step-by-Step Procedure

The following example requires you to navigate various levels in the configuration hierarchy. For instructions on how to do that, see *Using the CLI Editor in Configuration Mode* in the *CLI User Guide*.

To configure and apply rewrite rules for a device:

1. Configure rewrite rules for DiffServ CoS.

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

2. Configure best-effort forwarding class rewrite rules.

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

3. Configure expedited forwarding class rewrite rules.

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

4. Configure an assured forwarding class rewrite rules.

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

5. Configure a network control class rewrite rules.

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

6. Apply rewrite rules to an IRB interface.

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

Results

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

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

Copyright © 2017, Juniper Networks, Inc.

```
    }
  }
  }
  rewrite-rules {
    dscp rewrite-dscps {
    forwarding-class be-class {
    loss-priority low code-point 000000;
      loss-priority high code-point 000001;
      }
      forwarding-class ef-class {
        loss-priority low code-point 101110;
      loss-priority high code-point 101111;
    }
    forwarding-class af-class {
      loss-priority low code-point 001010;
    loss-priority high code-point 001100;
  }
  forwarding-class nc-class {
    loss-priority low code-point 110000;
    loss-priority high code-point 110001;
    }
  }
}
```

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

Verification

Verifying Rewrite Rules Configuration

Purpose Verify that rewrite rules are configured properly.

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

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

Logical interface: irb.10, Index: 71
Object                 Name                  Type                  Index
Classifier             ipprec-compatibility  ip                       13
```

Meaning Rewrite rules are configured on IRB interface as expected.

Related Documentation

- Rewrite Rules Overview on page 75

Copyright © 2017, Juniper Networks, Inc.

CHAPTER 6

Defining Output Queue Properties with Schedulers

- Schedulers Overview on page 81
- Default Scheduler Settings on page 86
- Transmission Scheduling Overview on page 87
- Excess Bandwidth Sharing and Minimum Logical Interface Shaping on page 88
- Excess Bandwidth Sharing Proportional Rates on page 89
- Calculated Weights Mapped to Hardware Weights on page 90
- Weight Allocation with Only Shaping Rates or Unshaped Logical Interfaces on page 91
- Shared Bandwidth Among Logical Interfaces on page 92
- Example: Configuring Class-of-Service Schedulers on page 94
- Scheduler Buffer Size Overview on page 98
- Example: Configuring a Large Delay Buffer on a Channelized T1 Interface on page 101
- Configuring Large Delay Buffers in CoS on page 104
- Example: Configuring and Applying Scheduler Maps on page 109

Schedulers Overview

You use schedulers to define the properties of output queues. These properties include the amount of interface bandwidth assigned to the queue, the size of the memory buffer allocated for storing packets, the priority of the queue, and the random early detection (RED) drop profiles associated with the queue.

You associate the schedulers with forwarding classes by means of scheduler maps. You can then associate each scheduler map with an interface, thereby configuring the hardware queues, packet schedulers, and RED processes that operate according to this mapping.

An individual device interface has multiple queues assigned to store packets temporarily before transmission. To determine the order to service the queues, the device uses a round-robin scheduling method based on priority and the queue's weighted round-robin (WRR) credits. Junos OS schedulers allow you to define the priority, bandwidth, delay

Copyright © 2017, Juniper Networks, Inc.

buffer size, rate control status, and RED drop profiles to be applied to a particular queue for packet transmission.

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

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

This topic contains the following sections:

- Transmit Rate on page 82
- Delay Buffer Size on page 83
- Scheduling Priority on page 84
- Shaping Rate on page 85

Transmit Rate

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

This bandwidth amount can be a fixed value, such as 1 megabit per second (Mbps), a percentage of the total available bandwidth, or the rest of the available bandwidth. You can limit the transmission bandwidth to the exact value you configure, or allow it to exceed the configured rate if additional bandwidth is available from other queues (SRX5400, SRX5600, and SRX5800 devices do not support an exact value transmit rate). This property helps ensure that each queue receives the amount of bandwidth appropriate to its level of service.

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

NOTE: Interfaces with slower interface speeds, like T1, E1, or channelized T1/E1/ISDN PRI, cannot support minimum transmit rates because the minimum transmit rate supported on a device is 3,200 bps.

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

Copyright © 2017, Juniper Networks, Inc.

The transmit rate defines the transmission rate of a scheduler. The transmit rate determines the traffic bandwidth from each forwarding class you configure.

By default, queues 0 through 7 have the following percentage of transmission capacity:

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

To define a transmit rate, select the appropriate option:

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

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

Delay Buffer Size

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

On Juniper Networks devices, you can configure larger delay buffers on channelized T1/E1 interfaces. Larger delay buffers help these slower interfaces to avoid congestion and packet dropping when they receive large bursts of traffic.

By default, SRX300, SRX320, SRX340, SRX345, and SRX550M device interfaces support a delay buffer time of 100,000 microseconds. (Platform support depends on the Junos OS release in your implementation.)

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

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

Copyright © 2017, Juniper Networks, Inc.

- To specify buffer size as a percentage of the total buffer, select **Percent** and type an integer from 1 through 100.
- To specify buffer size as the remaining available buffer, select **Remainder**.

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

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

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

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

NOTE: For a Juniper Networks device, if the buffer size percentage is set to zero for T1 interfaces, traffic does not pass.

Scheduling Priority

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

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

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

The scheduling priority of the scheduler determines the order in which an output interface transmits traffic from the queues. You can set scheduling priority at different levels in an

Copyright © 2017, Juniper Networks, Inc.

order of increasing priority from low to high. A high-priority queue with a high transmission rate might lock out lower-priority traffic.

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

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

Shaping Rate

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

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

By default, the logical interface bandwidth is the average of unused bandwidth for the number of logical interfaces that require default bandwidth treatment. You can specify a peak bandwidth rate in bits per second (bps), either as a complete decimal number or as a decimal number followed by the abbreviation *k* (1000), *m* (1,000,000), or *g* (1,000,000,000). The range is from 1000 through 32,000,000,000 bps.

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

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

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

Related Documentation

- Default Scheduler Settings on page 86
- Example: Configuring Class-of-Service Schedulers on page 94
- Scheduler Buffer Size Overview on page 98

Copyright © 2017, Juniper Networks, Inc.

- Example: Configuring a Large Delay Buffer on a Channelized T1 Interface on page 101
- Example: Configuring and Applying Scheduler Maps on page 109
- Transmission Scheduling Overview on page 87

Default Scheduler Settings

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

By default, the best-effort forwarding class (queue 0) receives 95 percent, and the network-control (queue 3) receives 5 percent of the bandwidth and buffer space for the output link. The default drop profile causes the buffer to fill and then discard all packets until it again has space.

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

By default, each queue can exceed the assigned bandwidth if additional bandwidth is available from other queues. When a forwarding class does not fully use the allocated transmission bandwidth, the remaining bandwidth can be used by other forwarding classes if they receive a larger amount of offered load than the bandwidth allocated. If you do not want a queue to use any leftover bandwidth, you must configure it for strict allocation.

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

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


Copyright © 2017, Juniper Networks, Inc.

Related Documentation

- Schedulers Overview on page 81
- Example: Configuring Class-of-Service Schedulers on page 94
- Scheduler Buffer Size Overview on page 98
- Example: Configuring a Large Delay Buffer on a Channelized T1 Interface on page 101
- Example: Configuring and Applying Scheduler Maps on page 109
- Transmission Scheduling Overview on page 87

Transmission Scheduling Overview

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

By default, each queue can exceed the assigned bandwidth if additional bandwidth is available from other queues. When a forwarding class does not fully use the allocated transmission bandwidth, the remaining bandwidth can be used by other forwarding classes if they receive a larger amount of offered load than the bandwidth allocated. A queue receiving traffic within its bandwidth configuration is considered to have positive bandwidth credit, and a queue receiving traffic in excess of its bandwidth allocation is considered to have negative bandwidth credit.

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

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

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

Table 17 on page 87 shows a sample configuration of priority and transmit rate on six queues. The total available bandwidth on the interface is 100 Mbps.

Table 17: Sample Transmission Scheduling

Queue	Scheduling Priority	Transmit Rate	Incoming Traffic
0	Low	10%	20 Mbps

Copyright © 2017, Juniper Networks, Inc.

Table 17: Sample Transmission Scheduling *(continued)*

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

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

- High priority set—Consists of queue 1 and queue 2. Packets use 40 Mbps (20+20) of the available bandwidth (100 Mbps) and are transmitted first. Because of positive credit, the configured transmit rate is not considered.
- Medium-high priority set—Consists of queue 4 and queue 5. Packets use 30 Mbps (10+20) of the remaining 60 Mbps bandwidth. Because of positive credit, the transmit rate is not considered. If the queues had negative credit, they would receive an equal share of the leftover bandwidth because no transmit rate is configured.
- Low priority set—Consists of queue 0 and queue 3. Packets share the 20 Mbps of leftover bandwidth based on the configured transmit rate. The distribution of bandwidth is in proportion to the assigned percentages. Because the total assigned percentage is 40 (10 + 30), each queue receives a share of bandwidth accordingly. Thus queue 0 receives 5 Mbps (10/40 x 20), and queue 3 receives 15 Mbps (30/40 x 20).

Related Documentation

- Schedulers Overview on page 81
- Default Scheduler Settings on page 86
- Example: Configuring Class-of-Service Schedulers on page 94
- Scheduler Buffer Size Overview on page 98
- Example: Configuring a Large Delay Buffer on a Channelized T1 Interface on page 101
- Example: Configuring and Applying Scheduler Maps on page 109

Excess Bandwidth Sharing and Minimum Logical Interface Shaping

The default excess bandwidth sharing proportional rate is 32.65 Mbps (128 Kbps x 255). In order to have better weighed fair queuing (WFQ) accuracy among queues, the shaping rate configured should be larger than the excess bandwidth sharing proportional rate. Some examples are shown in Table 18 on page 89.

Copyright © 2017, Juniper Networks, Inc.

Table 18: Shaping Rates and WFQ Weights

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

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

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

Related Documentation

- Schedulers Overview on page 81
- Excess Bandwidth Sharing Proportional Rates on page 89
- Calculated Weights Mapped to Hardware Weights on page 90
- Weight Allocation with Only Shaping Rates or Unshaped Logical Interfaces on page 91
- Shared Bandwidth Among Logical Interfaces on page 92

Excess Bandwidth Sharing Proportional Rates

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

As an example, consider five logical interfaces on the same physical port, each with four queues, all with only PIRs configured and no CIRs. The WRR rate is the same as the PIR for the logical interface. The excess bandwidth is shared proportionally with a rate of 40 Mbps. The traffic control profiles for the logical interfaces are shown in Table 19 on page 89.

Table 19: Example Shaping Rates and WFQ Weights

Shaping Rate	Configured Queue Transmit Rate	WFQ Weight	Total Weights
(Unit 0) 10 Mbps	(95, 0, 0, 5)	(60, 0, 0, 3)	63

Copyright © 2017, Juniper Networks, Inc.

Table 19: Example Shaping Rates and WFQ Weights *(continued)*

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

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

Related Documentation

- Schedulers Overview on page 81
- Excess Bandwidth Sharing and Minimum Logical Interface Shaping on page 88
- Calculated Weights Mapped to Hardware Weights on page 90
- Weight Allocation with Only Shaping Rates or Unshaped Logical Interfaces on page 91
- Shared Bandwidth Among Logical Interfaces on page 92

Calculated Weights Mapped to Hardware Weights

The calculated weight in a traffic control profile is mapped to hardware weight, but the hardware only supports a limited WFQ profile. The weights are rounded to the nearest hardware weight according to the values in Table 20 on page 90.

Table 20: Rounding Configured Weights to Hardware Weights

Traffic Control Profile Number	Number of Traffic Control Profiles	Weights	Maximum Error
1–16	16	1–16 (interval of 1)	50.00%
17–29	13	18–42 (interval of 2)	6.25%
30–35	6	45–60 (interval of 3)	1.35%
36–43	8	64–92 (interval of 4)	2.25%
44–49	6	98–128 (interval of 6)	3.06%
50–56	7	136–184 (interval of 8)	3.13%
57–62	6	194–244 (interval of 10)	2.71%

Copyright © 2017, Juniper Networks, Inc.

Table 20: Rounding Configured Weights to Hardware Weights *(continued)*

Traffic Control Profile Number	Number of Traffic Control Profiles	Weights	Maximum Error
63–63	1	255–255 (interval of 11)	2.05%

As shown in Table 20 on page 90, the calculated weight of 18.9 is mapped to a hardware weight of 18, because 18 is closer to 18.9 than 20 (an interval of 2 applies in the range of 18 to 42).

Related Documentation

- Schedulers Overview on page 81
- Excess Bandwidth Sharing and Minimum Logical Interface Shaping on page 88
- Excess Bandwidth Sharing Proportional Rates on page 89
- Weight Allocation with Only Shaping Rates or Unshaped Logical Interfaces on page 91
- Shared Bandwidth Among Logical Interfaces on page 92

Weight Allocation with Only Shaping Rates or Unshaped Logical Interfaces

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

For example, a logical interface configuration with five units is shown in Table 21 on page 91.

Table 21: Allocating Weights with PIR and CIR on Logical Interfaces

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

Copyright © 2017, Juniper Networks, Inc.

The weights for these units are calculated as follows:

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

Related Documentation

- Schedulers Overview on page 81
- Excess Bandwidth Sharing and Minimum Logical Interface Shaping on page 88
- Excess Bandwidth Sharing Proportional Rates on page 89
- Calculated Weights Mapped to Hardware Weights on page 90
- Shared Bandwidth Among Logical Interfaces on page 92

Shared Bandwidth Among Logical Interfaces

As a simple example showing how bandwidth is shared among the logical interfaces, assume that all traffic is sent on queue 0. Assume also that there is a 40-Mbps load on all of the logical interfaces. Configuration details are shown in Table 22 on page 92.

Table 22: Example of Shared Bandwidth Among Logical Interfaces

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

Copyright © 2017, Juniper Networks, Inc.

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

When the port is shaped at 100 Mbps, because units 2 and 3 have a guaranteed rate (CIR) configured, each of them can transmit 20 Mbps. On units 1, 2, 3, and 4, the 60 Mbps of excess bandwidth is shaped according to the values shown in Table 23 on page 93.

Table 23: First Example of Bandwidth Sharing

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

However, unit 3 only has 20 Mbps extra (PIR and CIR) configured. This means that the leftover bandwidth of 16.22 Mbps (36.22 Mbps – 20 Mbps) is shared among units 1, 2, and 4. This is shown in Table 24 on page 93.

Table 24: Second Example of Bandwidth Sharing

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

Finally, Table 25 on page 93 shows the resulting allocation of bandwidth among the logical interfaces when the port is configured with a 100-Mbps shaping rate.

Table 25: Final Example of Bandwidth Sharing

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

Related Documentation

- Schedulers Overview on page 81
- Excess Bandwidth Sharing and Minimum Logical Interface Shaping on page 88

Copyright © 2017, Juniper Networks, Inc.

- Excess Bandwidth Sharing Proportional Rates on page 89
- Calculated Weights Mapped to Hardware Weights on page 90
- Weight Allocation with Only Shaping Rates or Unshaped Logical Interfaces on page 91

Example: Configuring Class-of-Service Schedulers

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

- Requirements on page 94
- Overview on page 94
- Configuration on page 95
- Verification on page 97

Requirements

Before you begin, determine the buffer size allocation method to use. See "Scheduler Buffer Size Overview" on page 98.

Overview

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

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

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

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

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

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

Copyright © 2017, Juniper Networks, Inc.

Table 26 on page 95 shows the schedulers created in this example.

Table 26: Sample Schedulers

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

Configuration

CLI Quick Configuration

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

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

Step-by-Step Procedure

The following example requires you to navigate various levels in the configuration hierarchy. For instructions on how to do that, see *Using the CLI Editor in Configuration Mode* in the *CLI User Guide*.

To configure CoS schedulers:

1. Configure a best-effort scheduler.

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

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

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

Copyright © 2017, Juniper Networks, Inc.

```
user@host# set buffer-size percent 40
```

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

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

4. Configure an expedited forwarding scheduler.

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

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

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

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

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

7. Configure an assured forwarding scheduler.

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

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

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

9. Configure an assured forwarding scheduler transmit rate.

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

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

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

11. Configure a network control scheduler.

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

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

Copyright © 2017, Juniper Networks, Inc.

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

13. Configure a network control scheduler transmit rate.

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

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

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

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

Verification

Verifying Schedulers Configuration

Purpose Verify that the schedulers are configured properly.

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

Copyright © 2017, Juniper Networks, Inc.

Related Documentation

- Schedulers Overview on page 81
- Default Scheduler Settings on page 86
- Example: Configuring a Large Delay Buffer on a Channelized T1 Interface on page 101
- Example: Configuring and Applying Scheduler Maps on page 109
- Transmission Scheduling Overview on page 87

Scheduler Buffer Size Overview

Large bursts of traffic from faster interfaces can cause congestion and dropped packets on slower interfaces that have small delay buffers. For example, a Juniper Networks device operating at the edge of the network can drop a portion of the burst traffic it receives on a channelized T1/E1 interface from a Fast Ethernet or Gigabit Ethernet interface on a router at the network core. On Juniper Networks devices, large delay buffers can be configured for both channelized T1/E1 and nonchannelized T1/E1 interfaces.

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

This section contains the following topics:

- Maximum Delay Buffer Sizes Available to Channelized T1/E1 Interfaces on page 98
- Maximum Delay Buffer Size for vSRX Interfaces on page 99
- Delay Buffer Size Allocation Methods on page 100
- Delay Buffer Sizes for Queues on page 100

Maximum Delay Buffer Sizes Available to Channelized T1/E1 Interfaces

When you enable the large delay buffer feature on interfaces, a larger buffer is available for allocation to scheduler queues. The maximum delay buffer size that is available for an interface depends on the maximum available delay buffer time and the speed of the interface as shown in Table 27 on page 98.

The default values are as follows:

- Clear-channel interface—The default delay buffer time is 500,000 microseconds (0.5 s).
- *N*xDS0 interface—The default delay buffer time is 1,200,000 microseconds (1.2 s).

Table 27: Maximum Available Delay Buffer Time by Channelized Interface and Rate

Effective Line Rate	Maximum Available Delay Buffer Time
< 4xDS0	4,000,000 microseconds (4 s)
< 8xDS0	2,000,000 microseconds (2 s)
< 16xDS0	1,000,000 microseconds (1 s)

Copyright © 2017, Juniper Networks, Inc.

Table 27: Maximum Available Delay Buffer Time by Channelized Interface and Rate *(continued)*

Effective Line Rate	Maximum Available Delay Buffer Time
<= 32xDS0	500,000 microseconds (0.5 s)
<= 10 mbps	400,000 microseconds (0.4 s)
<= 20 mbps	300,000 microseconds (0.3 s)
<= 30 mbps	200,000 microseconds (0.2 s)
<= 40 mbps	150,000 microseconds (0.15 s)

You can calculate the maximum delay buffer size available for an interface, with the following formula:

interface speed x *maximum delay buffer time* = *maximum available delay buffer size*

For example, the following maximum delay buffer sizes are available to 1xDS0 and 2xDS0 interfaces:

1xDS0—64 Kbps x 4 s = 256 Kb (32 KB)

2xDS0—128 Kbps x 4 s = 512 Kb (64 KB)

If you configure a delay buffer size larger than the maximum, the system allows you to commit the configuration but displays a system log warning message and uses the default buffer size setting instead of the configured maximum setting.

Maximum Delay Buffer Size for vSRX Interfaces

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

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

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

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

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

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

Copyright © 2017, Juniper Networks, Inc.

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

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

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

Delay Buffer Size Allocation Methods

You can specify delay buffer sizes for each queue using schedulers. The queue buffer can be specified as a period of time (microseconds) or as a percentage of the total buffer or as the remaining buffer. Table 28 on page 100 shows different methods that you can specify for buffer allocation in queues.

Table 28: Delay Buffer Size Allocation Methods

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

Delay Buffer Sizes for Queues

You specify delay buffer sizes for queues using schedulers. The system calculates the buffer size of a queue based on the buffer allocation method you specify for it in the

Copyright © 2017, Juniper Networks, Inc.

scheduler. See Table 28 on page 100 for different buffer allocation methods and Table 29 on page 101 for buffer size calculations.

Table 29: Delay Buffer Allocation Method and Queue Buffer

Buffer Size Allocation Method	Queue Buffer Calculation	Example
Percentage	*available interface bandwidth* x *configured buffer size percentage* x *maximum delay buffer time* = *queue buffer*	Suppose you configure a queue on a 1xDS0 interface to use 30 percent of the available delay buffer size. The system uses the maximum available delay buffer time (4 seconds) and allocates the queue 9600 bytes of delay buffer: 64 Kbps x 0.3 x 4 s=76,800 bits=9,600 bytes
Temporal	*available interface bandwidth* x *configured transmit rate percentage* x *configured temporal buffer size* = *queue buffer*	Suppose you configure a queue on a 1xDS0 interface to use 300,000,000 microseconds (3 seconds) of delay buffer, and you configure the transmission rate to be 20 percent. The queue receives 4800 bytes of delay buffer: 64 Kbps x 0.2 x 3 s=38,400 bits=4,800 bytes If you configure a temporal value that exceeds the maximum available delay buffer time, the queue is allocated the buffer remaining after buffers are allocated for the other queues. Suppose you configure a temporal value of 6,000,000 microseconds on a 1xDS0 interface. Because this value exceeds the maximum allowed value of 4,000,000 microseconds, the queue is allocated the remaining delay buffer.

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

Related Documentation

- Schedulers Overview on page 81
- Default Scheduler Settings on page 86
- Example: Configuring Class-of-Service Schedulers on page 94
- Example: Configuring a Large Delay Buffer on a Channelized T1 Interface on page 101
- Example: Configuring and Applying Scheduler Maps on page 109
- Transmission Scheduling Overview on page 87

Example: Configuring a Large Delay Buffer on a Channelized T1 Interface

This example shows how to configure a large delay buffer on a channelized T1 interface to help slower interfaces avoid congestion and packet dropping when they receive large bursts of traffic.

- Requirements on page 102
- Overview on page 102

Copyright © 2017, Juniper Networks, Inc.

- Configuration on page 102
- Verification on page 103

Requirements

Before you begin, enable the large buffer feature on the channelized T1/E1 PIM and then configure a buffer size for each queue in the CoS scheduler. See "Scheduler Buffer Size Overview" on page 98.

Overview

On devices, you can configure large delay buffers on channelized T1/E1 interfaces. Each channelized T1/E1 interface can be configured as a single clear channel, or for channelized (NxDS0) operation, where N denotes channels 1 to 24 for a T1 interface and channels 1 to 32 for an E1 interface.

In this example, you specify a queue buffer of 30 percent in scheduler **be-scheduler** and associate the scheduler to a defined forwarding class **be-class** using scheduler map **large-buf-sched-map**. Finally, you apply the scheduler map to channelized T1 interface **t1-3/0/0**.

Configuration

CLI Quick Configuration

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

```
set chassis fpc 3 pic 0 q-pic-large-buffer
set class-of-service schedulers be-scheduler buffer-size percent 30
set class-of-service scheduler-maps large-buf-sched-map forwarding-class be-class
  scheduler be-scheduler
set class-of-service interfaces t1-3/0/0 unit 0 scheduler-map large-buf-sched-map
```

Step-by-Step Procedure

The following example requires you to navigate various levels in the configuration hierarchy. For instructions on how to do that, see *Using the CLI Editor in Configuration Mode* in the *CLI User Guide*.

To configure a large delay buffer on a channelized T1 interface:

1. Enable the large buffer size feature on the channelized T1 interface.

   ```
   [edit]
   user@host# edit chassis
   user@host# set fpc 3 pic 0 q-pic-large-buffer
   ```

2. Create best-effort traffic and specify a buffer size.

   ```
   [edit]
   user@host# edit class-of-service
   user@host# set schedulers be-scheduler buffer-size percent 30
   ```


Copyright © 2017, Juniper Networks, Inc.

3. Configure the scheduler map to associate schedulers with defined forwarding classes.

```
[edit class-of-service]
user@host# set scheduler-maps large-buf-sched-map forwarding-class be-class
  scheduler be-scheduler
```

4. Apply the scheduler map to the channelized T1 interface.

```
[edit class-of-service]
user@host# set interfaces t1-3/0/0 unit 0 scheduler-map large-buf-sched-map
```

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

```
[edit]
  user@host# show class-of-service
  interfaces {
  t1-3/0/0 {
  unit 0 {
    scheduler-map large-buf-sched-map;
    }
  }
  }
  scheduler-maps {
    large-buf-sched-map {
    forwarding-class be-class scheduler be-scheduler;
  }
  }
  schedulers {
    be-scheduler {
    buffer-size percent 30;
  }
  }
  [edit]
    user@host# show chassis
    fpc 3 {
    pic 0 {
    q-pic-large-buffer;
  }
}
```

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

Verification

Verifying Large Delay Buffers Configuration

Purpose Verify that the large delay buffers are configured properly.

Action From configuration mode, enter the **show class-of-service** and **show chassis** commands.

Copyright © 2017, Juniper Networks, Inc.

Related Documentation

- Schedulers Overview on page 81
- Default Scheduler Settings on page 86
- Example: Configuring Class-of-Service Schedulers on page 94
- Example: Configuring and Applying Scheduler Maps on page 109
- Transmission Scheduling Overview on page 87

Configuring Large Delay Buffers in CoS

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

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

Configuring Large Delay Buffers

The following configuration applies to the examples that follow:

1. Configure two VLANs (one ingress, one egress) on one interface. No interface shaping rate is initially defined for this configuration.

```
[edit]
set interfaces ge-0/0/3 per-unit-scheduler
set interfaces ge-0/0/3 vlan-tagging
set interfaces ge-0/0/3 unit 102 vlan-id 102
set interfaces ge-0/0/3 unit 102 family inet address 203.0.113.2/24
set interfaces ge-0/0/3 unit 201 vlan-id 201
set interfaces ge-0/0/3 unit 201 family inet address 198.51.100.2/24
set routing-options static route 192.02.1/32 next-hop 198.51.100.3
```

2. Enable the **q-pic-large-buffer** option on the same PIC, in addition to the **buffer-size temporal** option on the queue, to create a large buffer on the queue:

```
[edit]
set chassis fpc 0 pic 0 q-pic-large-buffer
```

NOTE: The CLI does not provide a warning when you use buffer-size temporal without q-pic-large-buffer. When you use buffer-size temporal, verify that the configuration also includes the q-pic-large buffer command.

3. Define four forwarding-classes (queue names) for the four queues:

Copyright © 2017, Juniper Networks, Inc.

```
[edit]
set class-of-service forwarding-classes queue 0 be-Queue0
set class-of-service forwarding-classes queue 1 video-Queue1
set class-of-service forwarding-classes queue 2 voice-Queue2
set class-of-service forwarding-classes queue 3 nc-Queue3
```

4. Configure the forwarding classes (queue names) included in a scheduler map, applied to the egress VLAN:

```
[edit]
set class-of-service interfaces ge-0/0/3 unit 201 scheduler-map schedMapM
set class-of-service scheduler-maps schedMapM forwarding-class be-Queue0
  scheduler be-Scheduler0
set class-of-service scheduler-maps schedMapM forwarding-class video-Queue1
  scheduler video-Scheduler1
set class-of-service scheduler-maps schedMapM forwarding-class voice-Queue2
  scheduler voice-Scheduler2
set class-of-service scheduler-maps schedMapM forwarding-class nc-Queue3
  scheduler nc-Scheduler3
```

5. Set the queue priorities. Only queue priorities are initially defined, not transmit rates or buffer sizes.

```
[edit]
set class-of-service schedulers be-Scheduler0 priority low
set class-of-service schedulers video-Scheduler1 priority medium-low
set class-of-service schedulers voice-Scheduler2 priority medium-high
set class-of-service schedulers nc-Scheduler3 priority high
```

Example: Simple Configuration Using Four Queues

This configuration allocates 12,500,000 bytes of buffer for each of the four queues. To avoid exceeding the limits of the delay buffer calculation, this initial example has no interface shaping rate, scheduler transmit rate, or scheduler buffer size percent configuration.

1. Specify the maximum 4-second delay buffer on each of the four queues:

```
[edit]
set class-of-service schedulers be-Scheduler0 buffer-size temporal 4m
set class-of-service schedulers video-Scheduler1 buffer-size temporal 4m
set class-of-service schedulers voice-Scheduler2 buffer-size temporal 4m
set class-of-service schedulers nc-Scheduler3 buffer-size temporal 4m
```

   Specifying **buffer-size temporal** on some or all queues uses implicit (default) or explicit transmit rate percentages as the buffer-size percentages of the temporal values for those queues. Because there are no explicitly specified transmit rate percentages, divide 100 percent by the number of configured queues (queues with schedulers configured in the scheduler map) to get the implicit (default) per-queue transmit rate percentages. Each queue gets an implicit (default) transmit rate of 100% / 4 = 25%.

   In this example, specifying the maximum 4-second delay on each queue, with no shaping rate on the interface and implicit (default) per-queue transmit rates of 25 percent, the total buffer for all temporal 4m queues on an interface = 4 seconds *

Copyright © 2017, Juniper Networks, Inc.

100,000,000 maximum interface bps / 8 bits/byte = 4 seconds * 12,500,000 bytes = 50,000,000 bytes. Each queue specifying temporal 4m gets 25% * 50,000,000 = 12,500,000 bytes.

2. Add a shaping rate of 4 Mbps to the interface:

   ```
   [edit]
   set class-of-service interfaces ge-0/0/3 unit 201 shaping-rate 4m
   ```

   The total buffer for all temporal 4m queues on an interface = 4 sec * 4,000,000 bps shaping-rate / 8 bits/byte = 4 sec * 500,000 bytes = 2,000,000 bytes. Therefore, each queue specifying temporal 4m receives 25% * 2,000,000 = 500,000 bytes.

When using **buffer-size temporal** on any interface queues, if you also use the **transmit-rate percent** command, or the **buffer-size percent** command, or both commands, on any of the interface queues, the buffer size calculations become more complex and the limits of available queue depth might be reached. If the configuration attempts to exceed the available memory, then at commit time two system log messages appear in the **/var/log/messages** file, the interface class-of-service configuration is ignored, and the interface class-of-service configuration reverts to the two-queue defaults:

```
Mar 11 11:02:10.239  elma-n4 elma-n4 COSMAN_FWDD: queue mem underflow for ge-0/0/3
Mar 11 11:02:10.240  elma-n4 elma-n4 cosman_compute_install_sched_params: Failed
 to compute scheduler params for ge-0/0/3.Hence retaining defaults
```

When configuring **buffer-size temporal** along with **transmit-rate percent** or **buffer-size percent**, or both, you must monitor the system log to see whether the available queue depth limit has been reached.

Example: Using buffer-size temporal with Explicit transmit-rate percent Commands

To add explicit transmit rates to all four queues:

```
[edit]
set class-of-service schedulers be-Scheduler0 transmit-rate percent 10
set class-of-service schedulers video-Scheduler1 transmit-rate percent 25
set class-of-service schedulers voice-Scheduler2 transmit-rate percent 25
set class-of-service schedulers nc-Scheduler3 transmit-rate percent 40
```

For example, if an interface is shaped to 4 Mbps, the transmit rate percentage of 10 for a queue means that the bandwidth share for the specific queue is 0.4 Mbps. The queues are allocated portions of the 2,000,000 bytes of total buffer available for temporal queues on this interface, proportionally to their transmit rates. The four queues get 200,000, 500,000, 500,000, and 800,000 bytes of delay buffer, respectively.

To avoid exceeding the queue depth limits and triggering system log messages and default configuration behavior, when configuring queues with **buffer-size temporal** and **transmit rate percent** and other (non-temporal) queues with **buffer-size percent**, the following configuration rule must be followed: When one or more queues on an interface are configured with **buffer-size temporal**, the sum of the temporal queues explicitly

Copyright © 2017, Juniper Networks, Inc.

configured transmit rate percentages plus other non-temporal queues explicitly configured buffer size percentages must not exceed 100 percent.

If the total of the temporal queues transmit rate percentages and the non-temporal queues buffer-size percentages exceeds 100 percent, the **queue mem underflow** and **Failed to compute scheduler params** system log messages appear in the messages log, the explicitly configured CLI CoS configuration for the interface is ignored, and the interface reverts to a two-queue default CoS configuration.

When **buffer-size temporal** is specified on a queue, if **transmit-rate percent** is also configured on the same queue, the queue depth configured is based on the fractional bandwidth for the queue as obtained by the specified **transmit-rate percent**.

In addition to temporal delay times specified for one or more queues using buffer size temporal, there is another delay time automatically computed for the entire interface. This interface delay time is distributed across all non-temporal queues, proportionally to their implicit (default) or explicit transmit-rate percentages. If **q-pic-large-buffer** is not enabled, the interface delay time defaults to 100 ms. As shown in Table 30 on page 107, when **q-pic-large-buffer** is enabled, interface delay time is calculated according to configured shaping rate for the interface. Because the shaping-rate configured in the example above was 4 Mbps (> 2,048,000 bps), the interface delay time for the configuration is 100 msec.

Table 30: Interface Delay Times Enabled By q-pic-large-buffer

Configured Shaping Rate (bps)	Interface Delay Time (msec) Used for Non-Temporal Queues with q-pic-large-buffer Enabled	Default Delay Time Used (msec) Without q-pic-large-buffer
64,000-255,999	4000	100
256,000 - 511,999	2000	100
512,000 - 102,3999	1000	100
1,024,000 - 2,047,999	500	100
>= 2,048,000	100	100

This example properly computes the delay buffer limits on both temporal and non-temporal queues:

1. Substitute **buffer-size percent** for **buffer-size temporal** on queues 0 and 1:

   ```
   [edit]
   delete class-of-service schedulers be-Scheduler0 buffer-size temporal 4m
   delete class-of-service schedulers video-Scheduler1 buffer-size temporal 4m
   set class-of-service schedulers be-Scheduler0 buffer-size percent 10
   set class-of-service schedulers video-Scheduler1 buffer-size percent 25
   ```

   This deletes the requirement for hard-specified 4 seconds of buffering and replaces it with a proportional limit of 10 percent (or 25 percent) of the total interface delay time for the non-temporal queues. In both cases, the queue depth is calculated based

Copyright © 2017, Juniper Networks, Inc.

on the share of the interface bandwidth for the specific queues. Total Interface Non-Temporal Queue Memory = shaping-rate * Interface delay time (Table 1) = 4 Mbps * 0.1 seconds = 500,000 bytes per second * 0.1 seconds = 50,000 bytes, therefore queues 0 and 1 get 10% * 50,000 = 5000 bytes and 25% * 50,000 = 12,500 bytes of delay buffer, respectively.

2. Configure **buffer-size temporal** on queues 2 and 3:

```
[edit]
set class-of-service schedulers voice-Scheduler2 buffer-size temporal 4m
set class-of-service schedulers voice-Scheduler2 transmit-rate percent 25
set class-of-service schedulers nc-Scheduler3 buffer-size temporal 4m
set class-of-service schedulers nc-Scheduler3 transmit-rate percent 40
```

Queues 2 and 3 still get 500,000 and 800,000 bytes of delay buffer, respectively, as previously calculated. This configuration obeys the rule that the sum of the temporal queues transmit rate percentages (25% + 40% = 65%), plus the non-temporal queues buffer size percentages (10% + 25% = 35%) do not exceed 100% (65% + 35% <= 100%).

The following example exceeds the delay buffer limit, triggering the system log messages and the default, two-queue class-of-service behavior.

Increase the buffer-size percentage from 25 percent to 26 percent for non-temporal queue 1:

```
[edit]
set class-of-service schedulers video-Scheduler1 buffer-size percent 26
```

This violates the configuration rule that the sum of the non-temporal queues buffer-size percentages (10% + 26% = 36%), plus the temporal queues transmit rate percentages (25% + 40% = 65%) now exceed 100% (36% + 65% = 101%). Therefore, the following two system log messages appear in the **/var/log/messages** file:

```
Mar 23 18:08:23  elma-n4 elma-n4 COSMAN_FWDD: %PFE-3: queue mem underflow for
ge-0/0/3 q_num(3)
Mar 23 18:08:23  elma-n4 elma-n4 cosman_compute_install_sched_params: %PFE-3:
Failed to compute scheduler params for ge-0/0/3.Hence retaining defaults
```

When the delay buffer limits are exceeded, the CLI-configured class-of-service settings are not used and the default class-of-service configuration (the default scheduler-map) is assigned to the interface. This uses two queues: the forwarding-class best-effort (queue 0) has transmit rate percent 95 and buffer-size percent 95 and the forwarding-class network-control (queue 3) has the transmit rate percent 5 and buffer-size percent 5.

```
queue 0: 1,187,500 Bytes
queue 1:     9,192 Bytes
queue 2:     9,192 Bytes
queue 3:    62,500 Bytes
```


Copyright © 2017, Juniper Networks, Inc.

Related Documentation

- Example: Configuring and Applying Scheduler Maps on page 109
- Scheduler Buffer Size Overview on page 98

Example: Configuring and Applying Scheduler Maps

This example shows how to configure and apply a scheduler map to a device's interface.

- Requirements on page 109
- Overview on page 109
- Configuration on page 110
- Verification on page 111

Requirements

Before you begin:

- Create and configure the forwarding classes. See *Configuring a Custom Forwarding Class for Each Queue*.
- Create and configure the schedulers. See "Example: Configuring Class-of-Service Schedulers" on page 94.

Overview

After you define a scheduler, you can include it in a scheduler map, which maps a specified forwarding class to a scheduler configuration. You configure a scheduler map to assign a forwarding class to a scheduler, and then apply the scheduler map to any interface that must enforce DiffServ CoS.

After they are applied to an interface, the scheduler maps affect the hardware queues, packet schedulers, and RED drop profiles.

In this example, you create the scheduler map diffserv-cos-map and apply it to the device's Ethernet interface ge-0/0/0. The map associates the mf-classifier forwarding classes to the schedulers as shown in Table 31 on page 109.

Table 31: Sample diffserv-cos-map Scheduler Mapping

mf-classifier Forwarding Class	For CoS Traffic Type	diffserv-cos-map Scheduler
be-class	Best-effort traffic	be-scheduler
ef-class	Expedited forwarding traffic	ef-scheduler
af-class	Assured forwarding traffic	af-scheduler
nc-class	Network control traffic	nc-scheduler

Copyright © 2017, Juniper Networks, Inc.

Configuration

CLI Quick Configuration

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

```
set class-of-service scheduler-maps diffserv-cos-map forwarding-class be-class scheduler
  be-scheduler
set class-of-service scheduler-maps diffserv-cos-map forwarding-class ef-class scheduler
  ef-scheduler
set class-of-service scheduler-maps diffserv-cos-map forwarding-class af-class scheduler
  af-scheduler
set class-of-service scheduler-maps diffserv-cos-map forwarding-class nc-class scheduler
  nc-scheduler
set class-of-service interfaces ge-0/0/0 unit 0 scheduler-map diffserv-cos-map
```

Step-by-Step Procedure

The following example requires you to navigate various levels in the configuration hierarchy. For instructions on how to do that, see *Using the CLI Editor in Configuration Mode* in the *CLI User Guide*.

To configure and apply a scheduler map to a device's interface:

1. Configure a scheduler map for DiffServ CoS.

   ```
   [edit class-of-service]
   user@host# edit scheduler-maps diffserv-cos-map
   ```

2. Configure a best-effort forwarding class and scheduler.

   ```
   [edit class-of-service scheduler-maps diffserv-cos-map]
   user@host# set forwarding-class be-class scheduler be-scheduler
   ```

3. Configure an expedited forwarding class and scheduler.

   ```
   [edit class-of-service scheduler-maps diffserv-cos-map]
   user@host# set forwarding-class ef-class scheduler ef-scheduler
   ```

4. Configure an assured forwarding class and scheduler.

   ```
   [edit class-of-service scheduler-maps diffserv-cos-map]
   user@host# set forwarding-class af-class scheduler af-scheduler
   ```

5. Configure a network control class and scheduler.

   ```
   [edit class-of-service scheduler-maps diffserv-cos-map]
   user@host# set forwarding-class nc-class scheduler nc-scheduler
   ```

6. Apply the scheduler map to an interface.

   ```
   [edit class-of-service]
   user@host# set interfaces ge-0/0/0 unit 0 scheduler-map diffserv-cos-map
   ```


Copyright © 2017, Juniper Networks, Inc.

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

```
[edit]
user@host# show class-of-service
interfaces {
   ge-0/0/0 {
      unit 0 {
      scheduler-map diffserv-cos-map;
   }
   }
}
scheduler-maps {
   diffserv-cos-map {
      forwarding-class be-class scheduler be-scheduler;
      forwarding-class ef-class scheduler ef-scheduler;
      forwarding-class af-class scheduler af-scheduler;
      forwarding-class nc-class scheduler nc-scheduler;
   }
}
```

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

Verification

Verifying the Scheduler Map Configuration

Purpose Verify that scheduler maps are configured properly.

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

Related Documentation

- *Default Schedulers Overview*
- *Configuring Schedulers*
- *Configuring Scheduler Maps*

Copyright © 2017, Juniper Networks, Inc.

Copyright © 2017, Juniper Networks, Inc.

CHAPTER 7

Removing Delays with Strict-Priority Queues

- Strict-Priority Queue Overview on page 113
- Understanding Strict-Priority Queues on page 114
- Example: Configuring Priority Scheduling on page 115
- Example: Configuring Strict-Priority Queuing on page 117
- Example: Configuring CoS Non-Strict Priority Scheduling on page 126

Strict-Priority Queue Overview

You can configure one queue per interface to have strict-priority, which causes delay-sensitive traffic, such as voice traffic, to be removed and forwarded with minimum delay. Packets that are queued in a strict-priority queue are removed before packets in other queues, including high-priority queues.

The strict-high-priority queuing feature allows you to configure traffic policing that prevents lower priority queues from being starved. The strict-priority queue does not cause starvation of other queues because the configured policer allows the queue to exceed the configured bandwidth only when other queues are not congested. If the interface is congested, the software directs strict-priority queues to the configured bandwidth.

To prevent queue starvation of other queues, you must configure an output (egress) policer that defines a limit for the amount of traffic that the queue can service. The software services all traffic in the strict-priority queue that is under the defined limit. When strict-priority traffic exceeds the limit, the policer marks the traffic in excess of the limit as out-of-profile. If the output port is congested, the software drops out-of-profile traffic.

You can also configure a second policer with an upper limit. When strict-priority traffic exceeds the upper limit, the software drops the traffic in excess of the upper limit, regardless of whether the output port is congested. This upper-limit policer is not a requirement for preventing starvation of the lower priority queues. The policer for the lower limit, which marks the packets as out-of-profile, is sufficient to prevent starvation of other queues.

Copyright © 2017, Juniper Networks, Inc.

Related Documentation
- Understanding Strict-Priority Queues on page 114
- Example: Configuring Priority Scheduling on page 115
- Example: Configuring Strict-Priority Queuing on page 117

Understanding Strict-Priority Queues

You use strict-priority queuing and policing as follows:

- Identify delay-sensitive traffic by configuring a behavior aggregate (BA) or multifield (MF) classifier.
- Minimize delay by assigning all delay-sensitive packets to the strict-priority queue.
- Prevent starvation on other queues by configuring a policer that checks the data stream entering the strict-priority queue. The policer defines a lower bound, marks the packets that exceed the lower bound as out-of-profile, and drops the out-of-profile packets if the physical interface is congested. If there is no congestion, the software forwards all packets, including the out-of-profile packets.
- Optionally, configure another policer that defines an upper bound and drops the packets that exceed the upper bound, regardless of congestion on the physical interface.

To configure strict-priority queuing and prevent starvation of other queues, include the **priority strict-high** statement at the **[edit class-of-service schedulers *scheduler-name*]** hierarchy level and the **if-exceeding** and **then out-of-profile** statements at the **[edit firewall policer *policer-name*]** hierarchy level:

```
[edit class-of-service schedulers scheduler-name]
priority strict-high;

[edit firewall policer policer-name]
if-exceeding {
   bandwidth-limit bps;
   bandwidth-percent number;
   burst-size-limit bytes;
}
then out-of-profile;
```

Related Documentation
- Strict-Priority Queue Overview on page 113
- Example: Configuring Priority Scheduling on page 115
- Example: Configuring Strict-Priority Queuing on page 117

Copyright © 2017, Juniper Networks, Inc.

Example: Configuring Priority Scheduling

This example shows how to configure priority scheduling so important traffic receives better access to the outgoing interface.

- Requirements on page 115
- Overview on page 115
- Configuration on page 115
- Verification on page 116

Requirements

Before you begin, review how to assign forwarding classes. See "Example: Assigning Forwarding Classes to Output Queues" on page 64.

Overview

In this example, you configure CoS and a scheduler called be-sched with a medium-low priority. Then you configure scheduler map be-map to associate be-sched with the best-effort forwarding class. Finally, you apply be-map to interface ge-0/0/0.

Configuration

CLI Quick Configuration

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

```
set class-of-service schedulers be-sched priority medium-low
set class-of-service scheduler-maps be-map forwarding-class best-effort scheduler
  be-sched
set class-of-service interfaces ge-0/0/0 scheduler-map be-map
```

Step-by-Step Procedure

The following example requires you to navigate various levels in the configuration hierarchy. For instructions on how to do that, see *Using the CLI Editor in Configuration Mode* in the *CLI User Guide*.

To configure priority scheduling:

1. Configure CoS and a scheduler.

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

2. Set a priority.

   ```
   [edit class-of-service schedulers be-sched]
   user@host# set priority medium-low
   ```

3. Configure a scheduler map.

Copyright © 2017, Juniper Networks, Inc.

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

4. Specify the best-effort forwarding class.

```
[edit class-of-service scheduler-maps be-map]
user@host# set forwarding-class best-effort scheduler be-sched
```

5. Apply best-effort map to an interface.

```
[edit]
user@host# edit class-of-service
user@host# set interfaces ge-0/0/0 scheduler-map be-map
```

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

```
[edit]
  user@host# show class-of-service
  interfaces {
  ge-0/0/0 {
  scheduler-map be-map;
    }
    }
    scheduler-maps {
      be-map {
      forwarding-class best-effort scheduler be-sched;
    }
    }
    schedulers {
      be-sched {
    priority medium-low;
  }
}
```

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

Verification

Verifying Priority Scheduling

Purpose Verify that the priority scheduling is configured properly on a device.

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

Related Documentation

- Strict-Priority Queue Overview on page 113
- Understanding Strict-Priority Queues on page 114

Copyright © 2017, Juniper Networks, Inc.

- Example: Configuring Strict-Priority Queuing on page 117

Example: Configuring Strict-Priority Queuing

This example shows how to configure strict-priority queuing and prevent starvation of other queues.

- Requirements on page 117
- Overview on page 117
- Configuration on page 117
- Verification on page 125

Requirements

Before you begin, review how to create and configure forwarding classes. See "Forwarding Classes Overview" on page 57.

Overview

In this example, you create a BA classifier to classify traffic based on the IP precedence of the packet. The classifier defines IP precedence value 101 as voice traffic and 000 as data traffic. You assign forwarding-class priority queue 0 to voice traffic and queue 1 as data traffic. You then configure the scheduler map as corp-map and voice scheduler as voice-sched.

Then you set the priority for the voice traffic scheduler as strict-high and for the data traffic scheduler as strict-low. You apply the BA classifier to input interface ge-0/0/0 and apply the scheduler map to output interface e1-1/0/0. You then configure two policers called voice-drop and voice-excess. You set the burst size limit and bandwidth limit for voice-drop policer and for voice-excess policer. You then create a firewall filter that includes the new policers and add the policer to the term.

Finally, you apply the filter to output interface e1-1/0/1 and set the IP address as 203.0.113.1/24.

Configuration

- Configuring a BA Classifier on page 118
- Configuring Forwarding Classes on page 118
- Configuring a Scheduler Map on page 119
- Configuring a Scheduler on page 120
- Applying a BA Classifier to an Input Interface on page 121
- Applying a Scheduler Map to an Output Interface on page 122
- Configuring Two Policers on page 122
- Applying a Filter to an Output Interface on page 124

Copyright © 2017, Juniper Networks, Inc.

Configuring a BA Classifier

CLI Quick Configuration

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

```
set class-of-service classifiers inet-precedence corp-traffic forwarding-class voice-class
  loss-priority low code-points 101
set class-of-service classifiers inet-precedence corp-traffic forwarding-class data-class
  loss-priority high code-points 000
```

Step-by-Step Procedure

To configure a BA classifier:

1. Create a BA classifier and set the IP precedence value for voice traffic.

```
[edit]
user@host# edit class-of-service classifiers inet-precedence corp-traffic
  forwarding-class voice-class loss-priority low
user@host# set code-points 101
```

2. Create a BA classifier and set the IP precedence value for data traffic.

```
[edit]
user@host# edit class-of-service classifiers inet-precedence corp-traffic
  forwarding-class data-class loss-priority high
user@host# set code-points 000
```

Results

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

```
[edit]
  user@host# show class-of-service
  classifiers {
  inet-precedence corp-traffic {
  forwarding-class voice-class {
    loss-priority low code-points 101;
    }
    forwarding-class data-class {
      loss-priority high code-points 000;
    }
  }
}
```

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

Configuring Forwarding Classes

CLI Quick Configuration

To quickly configure this section of the example, copy the following commands, paste them into a text file, remove any line breaks, change any details necessary to match your

Copyright © 2017, Juniper Networks, Inc.

network configuration, copy and paste the commands into the CLI at the **[edit]** hierarchy level, and then enter **commit** from configuration mode.

```
set class-of-service forwarding-classes queue 0 voice-class
set class-of-service forwarding-classes queue 1 data-class
```

Step-by-Step Procedure

To configure forwarding classes:

1. Assign priority queuing to voice traffic.

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

2. Assign priority queuing to data traffic.

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

Results

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

```
[edit]
user@host# show class-of-service
forwarding-classes {
   queue 0 voice-class;
   queue 1 data-class;
}
```

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

Configuring a Scheduler Map

CLI Quick Configuration

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

```
set class-of-service scheduler-maps corp-map forwarding-class voice-class scheduler
   voice-sched
set class-of-service scheduler-maps corp-map forwarding-class data-class scheduler
   data-sched
```

Step-by-Step Procedure

To configure a scheduler map:

1. Configure a scheduler map and voice scheduler.

```
[edit]
user@host# edit class-of-service scheduler-maps corp-map forwarding-class
   voice-class
user@host# set scheduler voice-sched
```

Copyright © 2017, Juniper Networks, Inc.

2. Configure a scheduler map and data scheduler.

```
[edit]
user@host# edit class-of-service scheduler-maps corp-map forwarding-class
  data-class
user@host# set scheduler data-sched
```

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

```
[edit]
user@host# show class-of-service
scheduler-maps {
  corp-map {
  forwarding-class voice-class scheduler voice-sched;
  forwarding-class data-class scheduler data-sched;
  }
}
```

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

Configuring a Scheduler

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

```
set class-of-service schedulers voice-sched priority strict-high
set class-of-service schedulers data-sched priority lowset xxx
```

Step-by-Step Procedure To configure schedulers:

1. Configure a voice traffic scheduler and set the priority.

```
[edit]
user@host# edit class-of-service schedulers voice-sched
user@host# set priority strict-high
```

2. Configure a data traffic scheduler and set the priority.

```
[edit]
user@host# edit class-of-service schedulers data-sched
user@host# set priority low
```

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

```
[edit]
```


Copyright © 2017, Juniper Networks, Inc.

```
user@host# show class-of-service
schedulers {
  voice-sched {
  priority strict-high;
  }
  data-sched {
    priority low;
  }
}
```

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

Applying a BA Classifier to an Input Interface

CLI Quick Configuration

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

```
set class-of-service interfaces ge-0/0/0 unit 0 classifiers inet-precedence corp-traffic
```

Step-by-Step Procedure

The following example requires you to navigate various levels in the configuration hierarchy. For instructions on how to do that, see *Using the CLI Editor in Configuration Mode* in the *CLI User Guide*.

To apply a BA classifier to an input interface:

1. Configure an interface.

```
[edit]
user@host# edit class-of-service interfaces ge-0/0/0 unit 0
```

2. Apply a BA classifier to an input interface.

```
[edit class-of-service interfaces ge-0/0/0 unit 0]
user@host# set classifiers inet-precedence corp-traffic
```

Results

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

```
[edit]
  user@host# show class-of-service interfaces
  ge-0/0/0 {
  unit 0 {
  classifiers {
    inet-precedence corp-traffic;
    }
  }
}
```

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

Copyright © 2017, Juniper Networks, Inc.

Applying a Scheduler Map to an Output Interface

CLI Quick Configuration

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

```
set class-of-service interfaces e1-1/0/0 unit 0 scheduler-map corp-map
```

Step-by-Step Procedure

The following example requires you to navigate various levels in the configuration hierarchy. For instructions on how to do that, see *Using the CLI Editor in Configuration Mode* in the *CLI User Guide*.

To apply the scheduler map to an output interface:

1. Configure an interface.

```
[edit]
user@host# edit class-of-service interfaces e1-1/0/0 unit 0
```

2. Apply a scheduler map to an output interface.

```
[edit class-of-service interfaces e1-1/0/0 unit 0]
user@host# set scheduler-map corp-map
```

Results

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

```
[edit]
  user@host# show class-of-service
  interfaces {
  e1-1/0/0 {
  unit 0 {
    scheduler-map corp-map;
    }
  }
}
```

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

Configuring Two Policers

CLI Quick Configuration

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

```
set firewall policer voice-drop if-exceeding burst-size-limit 200000 bandwidth-limit
  2000000
set firewall policer voice-drop then discard
```


Copyright © 2017, Juniper Networks, Inc.

```
set firewall policer voice-excess if-exceeding burst-size-limit 200000 bandwidth-limit
  1000000
set firewall policer voice-excess then out-of-profile
set firewall filter voice-term term 01 from forwarding-class voice-class
set firewall filter voice-term term 01 then policer voice-drop next term
set firewall filter voice-term term 02 from forwarding-class voice-class
set firewall filter voice-term term 02 then policer voice-excess accept
```

Step-by-Step Procedure

To configure two policers:

1. Configure a policer voice drop.

```
[edit]
user@host# edit firewall policer voice-drop
user@host# set if-exceeding burst-size-limit 200000 bandwidth-limit 2000000
user@host# set then discard
```

2. Configure a policer voice excess.

```
[edit]
user@host# edit firewall policer voice-excess
user@host# set if-exceeding burst-size-limit 200000 bandwidth-limit 1000000
user@host# set then out-of-profile
```

3. Create a firewall filter that includes the new policers.

```
[edit]
user@host# edit firewall filter voice-term term 01
user@host# set from forwarding-class voice-class
user@host# set then policer voice-drop next term
```

4. Add the policer to the term.

```
[edit]
user@host# edit firewall filter voice-term term 02
user@host# set from forwarding-class voice-class
user@host# set then policer voice-excess accept
```

Results

From configuration mode, confirm your configuration by entering the **show firewall** command. If the output does not display the intended configuration, repeat the configuration instructions in this example to correct it.

```
[edit]
  user@host# show firewall
  policer voice-drop {
  if-exceeding {
  bandwidth-limit 2m;
    burst-size-limit 200k;
    }
    then discard;
    }
    policer voice-excess {
      if-exceeding {
```

Copyright © 2017, Juniper Networks, Inc.

```
        bandwidth-limit 1m;
        burst-size-limit 200k;
        }
          then out-of-profile;
        }
        filter voice-term {
          term 01 {
        from {
        forwarding-class voice-class;
        }
        then {
      policer voice-drop;
      next term;
        }
      }
      term 02 {
        from {
        forwarding-class voice-class;
        }
        then {
          policer voice-excess;
        accept;
      }
    }
  }
```

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

Applying a Filter to an Output Interface

CLI Quick Configuration

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

```
set interfaces e1-1/0/1 unit 0 family inet filter output voice-term
set interfaces e1-1/0/1 unit 0 family inet address 203.0.113.1/24
```

Step-by-Step Procedure

To apply a filter to an output interface:

1. Apply a filter to an interface.

```
[edit]
user@host# edit interfaces e1-1/0/1 unit 0 family inet filter output
user@host# set voice-term
```

2. Set an IP address.

```
[edit]
user@host# set interfaces e1-1/0/1 unit 0 family inet address 203.0.113.1/24
```


Copyright © 2017, Juniper Networks, Inc.

Results From configuration mode, confirm your configuration by entering the **show interfaces** command. If the output does not display the intended configuration, repeat the configuration instructions in this example to correct it.

```
[edit]
  user@host# show interfaces
  e1-1/0/1 {
  unit 0 {
  family inet {
    filter {
    output voice-term;
    }
    address 203.0.113.1/24;
    }
  }
}
```

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

Verification

Confirm that the configuration is working properly.

- Verifying the Scheduler Map on page 125
- Verifying the Interfaces on page 125
- Verifying the Interface Queues on page 125

Verifying the Scheduler Map

Purpose Verify that the scheduler map is configured properly.

Action From operational mode, enter the **show class-of-service scheduler-map corp-map** command.

Verifying the Interfaces

Purpose Verify that the interfaces are configured properly.

Action From configuration mode, enter the **show interfaces** command.

Verifying the Interface Queues

Purpose Verify that the interface queues are configured properly.

Action From configuration mode, enter the **show interfaces queue** command.

Copyright © 2017, Juniper Networks, Inc.

Related Documentation

- Strict-Priority Queue Overview on page 113
- Understanding Strict-Priority Queues on page 114
- Example: Configuring Priority Scheduling on page 115

Example: Configuring CoS Non-Strict Priority Scheduling

Starting in Junos OS Release 15.1X49-D80, you can configure non-strict priority scheduling to avoid starvation of lower priority queues on SRX300, SRX320, SRX340, SRX345, SRX550M, SRX1500, and vSRX 2.0 devices.

This example shows how to assign non-strict priority scheduling to CoS queues.

- Requirements on page 126
- Overview on page 126
- Configuration on page 127
- Verification on page 129

Requirements

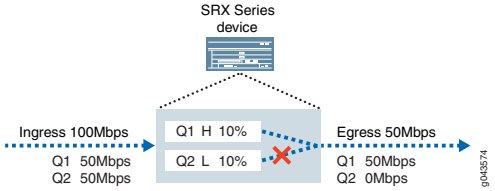
Before you begin, determine the shaping rate, schedulers, and forwarding classes for the CoS traffic. See **shaping-rate (CoS Interfaces)**, "Example: Configuring Class-of-Service Schedulers" on page 94, and "Example: Assigning Forwarding Classes to Output Queues" on page 64.

Overview

Traffic shaping bandwidth allocation is based on the egress (outgoing) interface that the packet traverses. If you have several traffic streams with CoS prioritized, all traffic streams across the network are sent with more bandwidth than the bandwidth on the egress interface. This can sometimes result in higher-priority queues getting all of the bandwidth and lower priority queues not getting any bandwidth, and thus being starved.

This example demonstrates how the non-strict priority feature can resolve the starvation of strict priority scheduling problem. For this scenario, you initialize two traffic streams (50 Mbps each) with CoS classifiers configured. Interface ge-0/0/1 is configured for ingress traffic, and ge-0/0/2 is configured for egress traffic with shaping enabled at 50 million. For traffic stream Q2, you set the queue priority as high and the shaping rate at 10%. For the other traffic stream Q1, you set the queue priority as low and the shaping rate at 10%. See Figure 6 on page 126.

Figure 6: CoS Traffic with HIgh and Low Priority Queues



Copyright © 2017, Juniper Networks, Inc.

NOTE: Since CoS is strict priority scheduling, please keep in mind that higher priority queues can starve lower priority queues.

Configuration

CLI Quick Configuration

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

```
set class-of-service interfaces ge-0/0/2 unit 0 shaping-rate 50m
set interfaces ge-0/0/2 per-unit-scheduler
set class-of-service interfaces ge-0/0/1 unit 0 classifiers dscp dscp_custom
set class-of-service classifiers dscp dscp_custom forwarding-class HIGH loss-priority
  low code-points 100011
set class-of-service classifiers dscp dscp_custom forwarding-class LOW loss-priority low
  code-points 100100
set class-of-service forwarding-classes queue 1 HIGH
set class-of-service forwarding-classes queue 0 LOW
set class-of-service scheduler-maps sched forwarding-class HIGH scheduler Q1
set class-of-service scheduler-maps sched forwarding-class LOW scheduler Q2
set class-of-service schedulers Q2 transmit-rate percent 10
set class-of-service schedulers Q2 priority high
set class-of-service schedulers Q1 transmit-rate percent 10
set class-of-service schedulers Q1 priority low
set-class-of-service non-strict-priority-scheduling
```

Step-by-Step Procedure

The following example requires you to navigate various levels in the configuration hierarchy. For instructions on how to do that, see *Using the CLI Editor in Configuration Mode* in the *CLI User Guide*.

To configure non-strict priority scheduling:

1. Configure shaping rate of 50 Mbps on the egress interface.

   ```
   [edit]
   user@host# set class-of-service interfaces ge-0/0/2 unit 0 shaping-rate 50m
   set interfaces ge-0/0/2 per-unit-scheduler
   ```

2. Configure classifiers on the ingress interface.

   ```
   [edit]
   user@host# set class-of-service interfaces ge-0/0/1 unit 0 classifiers dscp
     dscp_custom
   ```

3. Define the DSCP value to be assigned to the forwarding class.

   ```
   [edit]
   user@host# set class-of-service classifiers dscp dscp_custom forwarding-class
     HIGH loss-priority low code-points 100011
   user@host# set class-of-service classifiers dscp dscp_custom forwarding-class
     LOW loss-priority low code-points 100100
   ```

Copyright © 2017, Juniper Networks, Inc.

4. Define the forwarding class to a queue number.

```
[edit]
user@host# set class-of-service forwarding-classes queue 1 HIGH
user@host# set class-of-service forwarding-classes queue 0 LOW
```

5. Map the forwarding classes to a scheduler to control prioritized queueing.

```
[edit]
user@host# set class-of-service scheduler-maps sched forwarding-class HIGH
  scheduler Q1
user@host# set class-of-service scheduler-maps sched forwarding-class LOW
  scheduler Q2
```

6. Define the schedulers with priority and transmit rates. The example uses the same ratio for transmit rate but defines different priorities.

```
[edit]
user@host# set class-of-service schedulers Q2 transmit-rate percent 10
user@host# set class-of-service schedulers Q2 priority high
user@host# set class-of-service schedulers Q1 transmit-rate percent 10
user@host# set class-of-service schedulers Q1 priority low
```

7. Configure the new non-strict-priority-scheduling option.

```
[edit]
user@host# set-class-of-service non-strict-priority-scheduling
```

Results From configuration mode, confirm your configuration by entering the **show interfaces queue** command. If the output does not display the intended configuration, repeat the configuration instructions in this example to correct it.

```
user@host> show interface queue ge-0/0/2
Queue: 0, Forwarding classes: LOW
  Queued:
    Packets              :              18085500                 8571 pps
    Bytes                :           18013158000              68297136 bps
  Transmitted:
    Packets              :               3800910                 2030 pps
    Bytes                :            3785706360              16178104 bps
    Tail-dropped packets :              14284525                 6534 pps
Queue: 1, Forwarding classes: HIGH
  Queued:
    Packets              :              18085556                 8541 pps
    Bytes                :           18013213776              68062256 bps
  Transmitted:
    Packets              :              11432620                 6107 pps
    Bytes                :           11386889520              48660808 bps
    Tail-dropped packets :               6652859                 2436 pps
```

You will notice that the LOW priority queue got some traffic.

Copyright © 2017, Juniper Networks, Inc.

NOTE: Traffic on the low priority queue is still less than the high priority queue, as the non-priority scheduling option still works to control traffic..

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

Verification

Verifying Non-Strict Priority Configuration

Purpose Verify that non-strict priority scheduling is configured properly.

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

Related Documentation

- non-strict-priority-scheduling on page 265
- Example: Configuring and Applying Scheduler Maps on page 109
- Transmission Scheduling Overview on page 87

Copyright © 2017, Juniper Networks, Inc.

Copyright © 2017, Juniper Networks, Inc.

CHAPTER 8

Controlling Congestion with Drop Profiles

- RED Drop Profiles Overview on page 131
- RED Drop Profiles and Congestion Control on page 132
- Configuring RED Drop Profiles on page 134
- Example: Configuring RED Drop Profiles on page 135
- Example: Configuring Segmented and Interpolated Style Profiles on page 137

RED Drop Profiles Overview

A drop profile is a feature of the random early detection (RED) process that allows packets to be dropped before queues are full. Drop profiles are composed of two main values—the queue fullness and the drop probability. The queue fullness represents percentage of memory used to store packets in relation to the total amount that has been allocated for that queue. The drop probability is a percentage value that correlates to the likelihood that an individual packet is dropped from the network. These two variables are combined in a graph-like format.

A random number between 0 and 100 is calculated for each packet. This random number is plotted against the drop profile having the current queue fullness of that particular queue. When the random number falls above the graph line, the packet is transmitted onto the physical media. When the number falls below the graph line, the packet is dropped from the network.

Randomly dropped packets are counted as RED-dropped, while packets dropped for other reasons (100% probability) are counted as tail-dropped.

You specify drop probabilities in the drop profile section of the class-of-service (CoS) configuration hierarchy and reference them in each scheduler configuration. For each scheduler, you can configure multiple separate drop profiles, one for each combination of loss priority (low, medium-low, medium-high, or high) and IP transport protocol (TCP or non-TCP or any).

NOTE: For some SRX Series devices, tcp and non-tcp values are not supported; only the value “any” is supported. Actual platform support depends on the Junos OS release in your implementation.

Copyright © 2017, Juniper Networks, Inc.

You can configure a maximum of 32 different drop profiles.

To configure RED drop profiles, include the following statements at the **[edit class-of-service]** hierarchy level of the configuration:

```
[edit class-of-service]
drop-profiles {
  profile-name {
    fill-level percentage drop-probability percentage;
    interpolate {
      drop-probability [ values ];
      fill-level [ values ];
    }
  }
}
```

Default Drop Profiles

By default, if you configure no drop profiles, RED is still in effect and functions as the primary mechanism for managing congestion. In the default RED drop profile, when the fill level is 0 percent, the drop probability is 0 percent. When the fill level is 100 percent, the drop probability is 100 percent.

Related Documentation

- Example: Configuring RED Drop Profiles on page 135

RED Drop Profiles and Congestion Control

If the device must support assured forwarding, you can control congestion by configuring random early detection (RED) drop profiles. RED drop profiles use drop probabilities for different levels of buffer fullness to determine which scheduling queue on the device is likely to drop assured forwarding packets under congested conditions. The device can drop packets when the queue buffer becomes filled to the configured percentage.

Assured forwarding traffic with the PLP (packet loss priority) bit set is more likely to be discarded than traffic without the PLP bit set. This example shows how to configure a drop probability and a queue fill level for both PLP and non-PLP assured forwarding traffic. It is only one example of how to use RED drop profiles.

The example shows how to configure the RED drop profiles listed in Table 32 on page 132.

Table 32: Sample RED Drop Profiles

Drop Profile	Drop Probability	Queue Fill Level
af-normal—For non-PLP (normal) assured forwarding traffic	Between 0 (never dropped) and 100 percent (always dropped)	Between 95 and 100 percent
af-with-plp—For PLP (aggressive packet dropping) assured forwarding traffic	Between 95 and 100 percent (always dropped)	Between 80 and 95 percent

To configure RED drop profiles for assured forwarding congestion control on the device:

Copyright © 2017, Juniper Networks, Inc.

1. Navigate to the top of the configuration hierarchy in either the J-Web or CLI configuration editor.
2. Perform the configuration tasks described in Table 33 on page 133.
3. If you are finished configuring the device, commit the configuration.
4. Go on to one of the following tasks:
   - To assign resources, priorities, and profiles to output queues, see "Example: Configuring Class-of-Service Schedulers" on page 94.
   - To apply rules to logical interfaces, see "Example: Configuring Virtual Channels" on page 149.
   - To use adaptive shapers to limit bandwidth for Frame Relay, see "Example: Configuring and Applying an Adaptive Shaper" on page 145.

Table 33: Configuring RED Drop Profiles for Assured Forwarding Congestion Control

Task	CLI Configuration Editor
Navigate to the **Class of service** level in the configuration hierarchy.	From the **[edit]** hierarchy level, enter **edit class-of-service**
Configure the lower drop probability for normal, non-PLP traffic.	Enter **edit drop-profiles af-normal interpolate** **set drop-probability 0** **set drop-probability 100**
Configure a queue fill level for the lower non-PLP drop probability.	Enter **set fill-level 95** **set fill-level 100**
Configure the higher drop probability for PLP traffic.	From the **[edit class of service]** hierarchy level, enter **edit drop-profiles af-with-PLP interpolate** **set drop-probability 95** **set drop-probability 100**
Configure a queue fill level for the higher PLP drop probability.	Enter **set fill-level 80** **set fill-level 95**

Copyright © 2017, Juniper Networks, Inc.

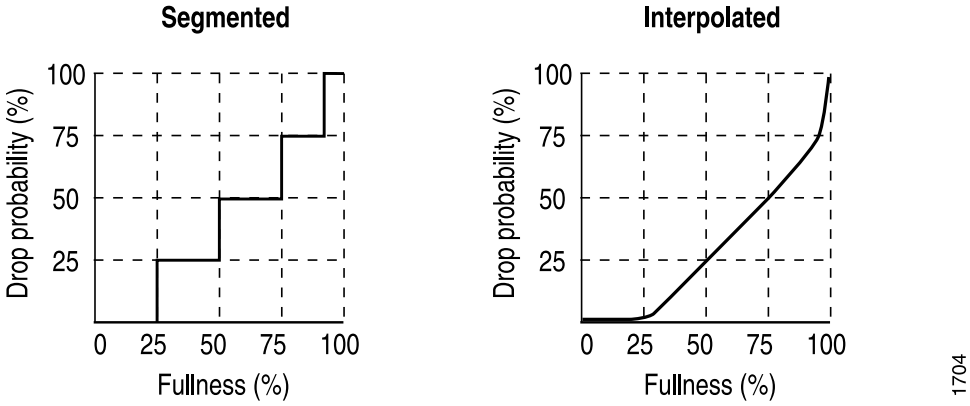
Related Documentation

- Example: Configuring RED Drop Profiles on page 135

Configuring RED Drop Profiles

Create a segmented configuration and an interpolated configuration that correspond to the graphs in Figure 7 on page 134. The values defined in the configuration are matched to represent the data points in the graph line. In this example, the drop probability is 25 percent when the queue is 50 percent full. The drop probability increases to 50 percent when the queue is 75 percent full.

Figure 7: Segmented and Interpolated Drop Profiles



Segmented

```
class-of-service {
  drop-profiles {
    segmented-style-profile {
      fill-level 25 drop-probability 25;
      fill-level 50 drop-probability 50;
      fill-level 75 drop-probability 75;
      fill-level 95 drop-probability 100;
    }
  }
}
```

To create the profile's graph line, the software begins at the bottom-left corner, representing a 0 percent fill level and a 0 percent drop probability. This configuration draws a line directly to the right until it reaches the first defined fill level, 25 percent for this configuration. The software then continues the line vertically until the first drop probability is reached. This process is repeated for all of the defined levels and probabilities until the top-right corner of the graph is reached.

Create a smoother graph line by configuring the profile with the **interpolate** statement. This allows the software to automatically generate 64 data points on the graph beginning at (0, 0) and ending at (100, 100). Along the way, the graph line intersects specific data points, which you define as follows:

Interpolated

```
class-of-service {
  drop-profiles {
    interpolated-style-profile {
```


Copyright © 2017, Juniper Networks, Inc.

```
            interpolate {
              fill-level [ 50 75 ];
              drop-probability [ 25 50 ];
            }
          }
        }
      }
```

Related Documentation

- *Understanding RED Drop Profiles*

Example: Configuring RED Drop Profiles

This example shows how to configure RED drop profiles.

- Requirements on page 135
- Overview on page 135
- Configuration on page 136
- Verification on page 137

Requirements

Before you begin, determine which type of profile you want to configure. See "Example: Configuring Segmented and Interpolated Style Profiles" on page 137.

Overview

A drop profile is a feature of the RED process that allows packets to be dropped before queues are full. Drop profiles are composed of two main values the queue fullness and the drop probability.

You can control congestion by configuring RED drop profiles, if the device supports assured forwarding. RED drop profiles use drop probabilities for different levels of buffer fullness to determine which scheduling queue on the device is likely to drop assured forwarding packets under congested conditions. The device can drop packets when the queue buffer becomes filled to the configured percentage. Assured forwarding traffic with the PLP bit set is more likely to be discarded than traffic without the PLP bit set.

In this example, you configure a drop probability and a queue fill level for both PLP and non-PLP assured forwarding traffic.

Table 34 on page 135 shows how to configure the RED drop profiles listed.

Table 34: Sample RED Drop Profiles

Drop Profile	Drop Probability	Queue Fill Level
af-normal—For non-PLP (normal) assured forwarding traffic	Between 0 (never dropped) and 100 percent (always dropped)	Between 95 and 100 percent
af-with-plp—For PLP (aggressive packet dropping) assured forwarding traffic	Between 95 and 100 percent (always dropped)	Between 80 and 95 percent

Copyright © 2017, Juniper Networks, Inc.

Configuration

CLI Quick Configuration

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

```
[edit]
set class-of-service drop-profiles af-normal interpolate drop-probability 0
set class-of-service drop-profiles af-normal interpolate drop-probability 100
set class-of-service drop-profiles af-normal interpolate fill-level 95
set class-of-service drop-profiles af-normal interpolate fill-level 100
set class-of-service drop-profiles af-with-PLP interpolate drop-probability 95
set class-of-service drop-profiles af-with-PLP interpolate drop-probability 100
set class-of-service drop-profiles af-with-PLP interpolate fill-level 80
set class-of-service drop-profiles af-with-PLP interpolate fill-level 95
```

Step-by-Step Procedure

The following example requires you to navigate various levels in the configuration hierarchy. For instructions on how to do that, see *Using the CLI Editor in Configuration Mode*.

To configure RED drop profiles:

1. Configure the lower drop probability for normal, non-PLP traffic.

   ```
   [edit]
   user@host# edit class-of-service
   user@host# edit drop-profiles af-normal interpolate
   user@host# set drop-probability 0
   user@host# set drop-probability 100
   ```

2. Configure a queue fill level for the lower non-PLP drop probability.

   ```
   [edit class-of-service drop-profiles af-normal interpolate]
   user@host# set fill-level 95
   user@host# set fill-level 100
   ```

3. Configure the higher drop probability for PLP traffic.

   ```
   [edit]
   user@host# edit class-of-service
   user@host# edit drop-profiles af-with-PLP interpolate
   user@host# set drop-probability 95
   user@host# set drop-probability 100
   ```

4. Configure a queue fill level for the higher PLP drop probability.

   ```
   [edit class-of-service drop-profiles af-with-PLP interpolate]
   user@host# set fill-level 80
   user@host# set fill-level 95
   ```


Copyright © 2017, Juniper Networks, Inc.

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

```
user@host# show class-of-service
  drop-profiles {
  af-normal {
    interpolate {
      fill-level [ 95 100 ];
      drop-probability [ 0 100 ];
    }
  }
  af-with-PLP {
    interpolate {
      fill-level [ 80 95 ];
      drop-probability [ 95 100 ];
    }
  }
}
```

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

Verification

Verifying RED Drop Profiles Configuration

Purpose Verify that the RED drop profiles are configured properly.

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

Related Documentation

- RED Drop Profiles Overview on page 131
- *Understanding RED Drop Profiles*

Example: Configuring Segmented and Interpolated Style Profiles

This example shows how to configure segmented and interpolated style profiles.

- Requirements on page 137
- Overview on page 138
- Configuration on page 138
- Verification on page 140

Requirements

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

Copyright © 2017, Juniper Networks, Inc.

Overview

In this example, you configure the segmented style profile by setting the drop probability to 25 percent when the queue is 25 percent full. The drop probability increases to 50 percent when the queue is 50 percent full. You set the drop probability to 75 percent when the queue is 75 percent full and finally the drop probability is set to 95 percent when the queue is 100 percent full.

Then you configure the interpolated style profile and set the fill level to 50 percent and 75 percent. Finally you set the drop probability to 25 percent and later to 50 percent.

Configuration

- Configuring Segmented Style Profiles on page 138
- Configuring Interpolated Style Profiles on page 139

Configuring Segmented Style Profiles

CLI Quick Configuration

To quickly configure segmented style profiles, copy the following commands and paste them into the CLI:

```
[edit]
set class-of-service drop-profiles segmented-style-profile fill-level 25 drop-probability
  25
set class-of-service drop-profiles segmented-style-profile fill-level 50 drop-probability
  50
set class-of-service drop-profiles segmented-style-profile fill-level 75 drop-probability
  75
set class-of-service drop-profiles segmented-style-profile fill-level 95 drop-probability
  100
```

Step-by-Step Procedure

The following example requires you to navigate various levels in the configuration hierarchy. For instructions on how to do that, see *Using the CLI Editor in Configuration Mode*.

To configure segmented style profiles:

1. Configure class of service.

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

2. Configure segmented style profile.

   ```
   [edit class-of-service]
   user@host# edit drop-profiles segmented-style-profile
   ```

3. Specify fill levels and drop probabilities.

   ```
   [edit class-of-service drop-profiles segmented-style-profile]
   user@host# set fill-level 25 drop-probability 25
   user@host# set fill-level 50 drop-probability 50
   user@host# set fill-level 75 drop-probability 75
   ```


Copyright © 2017, Juniper Networks, Inc.

```
user@host# set fill-level 95 drop-probability 100
```

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

```
[edit]
user@host# show class-of-service
  drop-profiles {
  segmented-style-profile {
    fill-level 25 drop-probability 25;
    fill-level 50 drop-probability 50;
    fill-level 75 drop-probability 75;
    fill-level 95 drop-probability 100;
  }
}
```

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

Configuring Interpolated Style Profiles

CLI Quick Configuration To quickly configure interpolated style profiles, copy the following commands and paste them into the CLI:

```
[edit]
set class-of-service drop-profiles interpolated-style-profile interpolate fill-level 50
set class-of-service drop-profiles interpolated-style-profile interpolate fill-level 75
set class-of-service drop-profiles interpolated-style-profile interpolate drop-probability
  25
set class-of-service drop-profiles interpolated-style-profile interpolate drop-probability
  50
```

Step-by-Step Procedure The following example requires you to navigate various levels in the configuration hierarchy. For instructions on how to do that, see *Using the CLI Editor in Configuration Mode*.

To configure interpolated style profile:

1. Configure class of service.

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

2. Configure interpolated style profile.

   ```
   [edit class-of-service]
   user@host# edit drop-profiles interpolated-style-profile interpolate
   ```

3. Specify fill levels.

   ```
   [edit class-of-service drop-profiles interpolated-style-profile interpolate]
   user@host# set fill-level 50
   ```

Copyright © 2017, Juniper Networks, Inc.

```
user@host# set fill-level 75
```

4. Specify drop probabilities.

```
[edit class-of-service drop-profiles interpolated-style-profile interpolate]
user@host# set drop-probability 25
user@host# set drop-probability 50
```

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

```
[edit]
user@host# show class-of-service
  drop-profiles {
  interpolated-style-profile {
    fill-level [ 50 75 ];
    drop-probability [ 25 50 ];
  }
  }
}
```

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

Verification

To confirm that the configuration is working properly, perform these tasks:

- Verifying Segmented Style Profile Configuration on page 140
- Verifying Interpolated Style Profile Configuration on page 140

Verifying Segmented Style Profile Configuration

Purpose Verify that the segmented style profile is configured properly.

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

Verifying Interpolated Style Profile Configuration

Purpose Verify that the interpolated style profile is configured properly.

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

Related Documentation

- *Junos OS Feature Support Reference for SRX Series and J Series Devices*
- RED Drop Profiles Overview on page 131
- *Understanding RED Drop Profiles*

Copyright © 2017, Juniper Networks, Inc.

- Example: Configuring RED Drop Profiles on page 135

Copyright © 2017, Juniper Networks, Inc.

Copyright © 2017, Juniper Networks, Inc.

CHAPTER 9

Controlling Congestion with Adaptive Shapers

- Adaptive Shaping Overview on page 143
- Assigning the Default Frame Relay Loss Priority Map to an Interface on page 144
- Defining a Custom Frame Relay Loss Priority Map on page 144
- Example: Configuring and Applying an Adaptive Shaper on page 145

Adaptive Shaping Overview

You can use adaptive shaping to limit the bandwidth of traffic flowing on a Frame Relay logical interface. If you configure and apply adaptive shaping, the device checks the backward explicit congestion notification (BECN) bit within the last inbound (ingress) packet received on the interface.

NOTE: Adaptive shaping is not available on SRX5600 and SRX5800 devices.

To configure an adaptive shaper, include the **adaptive-shaper** statement at the [edit class-of-service] hierarchy level:

```
[edit class-of-service]
adaptive-shaper {
   adaptive-shaper-name {
      trigger type shaping-rate (percent percentage | rate);
   }
}
```

The trigger type can be **BECN** only. If the last ingress packet on the logical interface has its BECN bit set to 1, the output queues on the logical interface are shaped according to the associated shaping rate.

The associated shaping rate can be a percentage of the available interface bandwidth from 0 through 100 percent. Alternatively, you can configure the shaping rate to be an absolute peak rate, in bits per second (bps) from 3200 through 32,000,000,000 bps. You can specify the value either as a complete decimal number or as a decimal number followed by the abbreviation **K** (1000), **M** (1,000,000), or **G** (1,000,000,000).

Copyright © 2017, Juniper Networks, Inc.

Related Documentation

- Assigning the Default Frame Relay Loss Priority Map to an Interface on page 144
- Defining a Custom Frame Relay Loss Priority Map on page 144
- Example: Configuring and Applying an Adaptive Shaper on page 145

Assigning the Default Frame Relay Loss Priority Map to an Interface

For SRX210, SRX240, and SRX650 device interfaces with Frame Relay encapsulation, you can set the loss priority of Frame Relay traffic based on the discard eligibility (DE) bit. (Platform support depends on the Junos OS release in your installation.) For each incoming frame with the DE bit containing the CoS value **0** or **1**, you can configure a Frame Relay loss priority value of low, medium-low, medium-high, or high.

The default Frame Relay loss priority map contains the following settings:

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

This default map sets the loss priority to low for each incoming frame with the DE bit containing the **0** CoS value. The map sets the loss priority to high for each incoming frame with the DE bit containing the **1** CoS value.

To assign the default map to an interface, include the **frame-relay-de default** statement at the [edit class-of-service interfaces *interface-name* unit *logical-unit-number* loss-priority-maps] hierarchy level:

```
[edit class-of-service interfaces interface-name unit logical-unit-number loss-priority-maps]
frame-relay-de default;
```

Defining a Custom Frame Relay Loss Priority Map

You can apply a classifier to the same interface on which you configure a Frame Relay loss priority value. The Frame Relay loss priority map is applied first, followed by the classifier. The classifier can change the loss priority to a higher value only (for example, from low to high). If the classifier specifies a loss priority with a lower value than the current loss priority of a particular packet, the classifier does not change the loss priority of that packet.

To define a custom Frame Relay loss priority map, include the following statements at the [edit class-of-service] hierarchy level:

```
[edit class-of-service]
loss-priority-maps {
  frame-relay-de map-name {
    loss-priority (low | medium-low | medium-high | high) code-point (0 | 1);
  }
}
```

A custom loss priority map sets the loss priority to low, medium-low, medium-high, or high for each incoming frame with the DE bit containing the specified **0** or **1** CoS value.

Copyright © 2017, Juniper Networks, Inc.

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

```
[edit class-of-service interfaces interface-name unit logical-unit-number loss-priority-maps]
frame-relay-de map-name;
```

Example: Configuring and Applying an Adaptive Shaper

This example shows how to configure and apply an adaptive shaper to limit the bandwidth of traffic on a Frame Relay logical interface.

- Requirements on page 145
- Overview on page 145
- Configuration on page 145
- Verification on page 146

Requirements

Before you begin, review how to create and apply scheduler maps. See "Example: Configuring and Applying Scheduler Maps" on page 109

Overview

In this example, you create adaptive shaper fr-shaper and apply it to T1 interface t1-0/0/2. The adapter shaper limits the transmit bandwidth on the interface to 64 Kbps.

Configuration

Step-by-Step Procedure

The following example requires you to navigate various levels in the configuration hierarchy. For instructions on how to do that, see *Using the CLI Editor in Configuration Mode* in the *CLI User Guide*.

To configure and apply an adaptive shaper to a logical interface:

1. Specify the name and the maximum transmit rate of the adaptive shaper.

   ```
   [edit]
   user@host# edit class-of-service
   user@host# set adaptive-shapers fr-shaper trigger becn shaping-rate 64k
   ```

2. Apply the adaptive shaper to the logical interface.

   ```
   [edit class-of-service]
   user@host# set interfaces t1-0/0/2 unit 0 adaptive-shaper fr-shaper
   ```

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

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

Copyright © 2017, Juniper Networks, Inc.

Verification

To verify the configuration is working properly, enter the **show class-of-service** command.

Related Documentation

- Adaptive Shaping Overview on page 143
- Assigning the Default Frame Relay Loss Priority Map to an Interface on page 144
- Defining a Custom Frame Relay Loss Priority Map on page 144

Copyright © 2017, Juniper Networks, Inc.

CHAPTER 10

Limiting Traffic Using Virtual Channels

- Virtual Channels Overview on page 147
- Understanding Virtual Channels on page 148
- Example: Configuring Virtual Channels on page 149

Virtual Channels Overview

You can configure virtual channels to limit traffic sent from a corporate headquarters to its branch offices. Virtual channels might be required when the headquarters site has an expected aggregate bandwidth higher than that of the individual branch offices. The headquarters router must limit the traffic sent to each branch office router to avoid oversubscribing their links. For instance, if branch 1 has a 1.5 Mbps link and the headquarters router attempts to send 6 Mbps to branch 1, all of the traffic in excess of 1.5-Mbps is dropped in the ISP network.

You configure virtual channels on a logical interface. Each virtual channel has a set of eight queues with a scheduler and an optional shaper. You can use an output firewall filter to direct traffic to a particular virtual channel. For example, a filter can direct all traffic with a destination address for branch office 1 to virtual channel 1, and all traffic with a destination address for branch office 2 to virtual channel 2.

Although a virtual channel group is assigned to a logical interface, a virtual channel is quite different from a logical interface. The only features supported on a virtual channel are queuing, packet scheduling, and accounting. Rewrite rules and routing protocols apply to the entire logical interface.

When you configure virtual channels on an interface, the virtual channel group uses the same scheduler and shaper you configure at the **[edit interfaces *interface-name* unit *logical-unit-number*]** hierarchy level. In this way, virtual channels are an extension of regular scheduling and shaping and are not independent entities.

Related Documentation

- Understanding Virtual Channels on page 148
- *Example: Configuring Virtual Channels*

Copyright © 2017, Juniper Networks, Inc.

Understanding Virtual Channels

You configure a virtual channel to set up queuing, packet scheduling, and accounting rules to be applied to one or more logical interfaces. You must apply then the virtual channel to a particular logical interface.

You also create a list of virtual channels that you can assign to a virtual channel group. To define a virtual channel group that you can assign to a logical interface, include the **virtual-channel-groups** statement at the **[edit class-of-service]** hierarchy level.

The *virtual-channel-group-name* can be any name that you want. The *virtual-channel-name* must be one of the names that you define at the **[edit class-of-service virtual-channels]** hierarchy level. You can include multiple virtual channel names in a group.

The scheduler map is required. The *map-name* must be one of the scheduler maps that you configure at the **[edit class-of-service scheduler-maps]** hierarchy level. For more information, see "Example: Configuring Class-of-Service Schedulers" on page 94.

The shaping rate is optional. If you configure the shaping rate as a percentage, when the virtual channel is applied to a logical interface, the shaping rate is set to the specified percentage of the interface bandwidth. If you configure a shaper on a virtual channel, the shaper limits the maximum bandwidth transmitted by that virtual channel. Virtual channels without a shaper can use the full logical interface bandwidth. If there are multiple unshaped virtual channels, they share the available logical interface bandwidth equally.

When you apply the virtual channel group to a logical interface, a set of eight queues is created for each of the virtual channels in the group. The **scheduler-map** statement applies a scheduler to these queues. If you include the **shaping-rate** statement, a shaper is applied to the entire virtual channel.

You must configure one of the virtual channels in the group to be the default channel. Therefore, the **default** statement is required in the configuration of one virtual channel per channel group. Any traffic not explicitly directed to a particular channel is transmitted by this default virtual channel.

For the corresponding physical interface, you must also include the **per-unit-scheduler** statement at the **[edit interfaces *interface-name*]** hierarchy level as follows:

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

The **per-unit-scheduler** statement enables one set of output queues for each logical interface configured under the physical interface.

When you apply a virtual channel group to a logical interface, the software creates a set of eight queues for each of the virtual channels in the group.

If you apply a virtual channel group to multiple logical interfaces, the software creates a set of eight queues on each logical interface. The virtual channel names listed in the group are used on all the logical interfaces. We recommend specifying the scheduler and shaping rates in the virtual channel configuration in terms of percentages, rather than

Copyright © 2017, Juniper Networks, Inc.

absolute rates. This allows you to apply the same virtual channel group to logical interfaces that have different bandwidths.

When you apply a virtual channel group to a logical interface, you cannot include the **scheduler-map** and **shaping-rate** statements at the [edit class-of-service interfaces *interface-name* unit *logical-unit-number*] hierarchy level. In other words, you can configure a scheduler map and a shaping rate on a logical interface, or you can configure virtual channels on the logical interface, but not both.

If you configure multiple logical interfaces on a single physical interface, each logical interface is guaranteed an equal fraction of the physical interface bandwidth as follows:

```
logical-interface-bandwidth =
physical-interface-bandwidth / number-of-logical-interfaces
```

If one or more logical interfaces do not completely use their allocation, the other logical interfaces share the excess bandwidth equally.

If you configure multiple virtual channels on a logical interface, they are each guaranteed an equal fraction of the logical interface bandwidth as follows:

```
virtual-channel-bandwidth =
logical-interface-bandwidth / number-of-virtual-channels
```

If you configure a shaper on a virtual channel, the shaper limits the maximum bandwidth transmitted by that virtual channel. Virtual channels without a shaper can use the full logical interface bandwidth. If there are multiple unshaped virtual channels, they share the available logical interface bandwidth equally.

Related Documentation

- Virtual Channels Overview on page 147
- *Example: Configuring Virtual Channels*

Example: Configuring Virtual Channels

This example shows how to create virtual channels between a headquarters and its branch office.

- Requirements on page 149
- Overview on page 150
- Configuration on page 150
- Verification on page 154

Requirements

Before you begin, ensure that your headquarters and branch office have a network connection where the expected aggregate bandwidth is higher for your headquarters than for your branch office. The devices at your headquarters will then be set up to limit the traffic sent to the branch office to avoid oversubscribing the link.

Copyright © 2017, Juniper Networks, Inc.

Overview

In this example, you create the virtual channels as branch1–vc, branch2–vc, branch3–vc, and default-vc. You then define the virtual channel group as wan-vc-group to include the four virtual channels and assign the scheduler map as bestscheduler to each virtual channel. Three of the virtual channels are shaped to 1.5 Mbps. The fourth virtual channel is default-vc, and it is not shaped so it can use the full interface bandwidth.

Then you apply them in the firewall filter as choose-vc to the device's interface t3-1/0/0. The output filter on the interface sends all traffic with a destination address matching 192.168.10.0/24 to branch1-vc, and similar configurations are set for branch2-vc and branch3-vc. Traffic not matching any of the addresses goes to the default, unshaped virtual channel.

Configuration

CLI Quick Configuration

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

```
set class-of-service virtual-channels branch1-vc
set class-of-service virtual-channels branch2-vc
set class-of-service virtual-channels branch3-vc
set class-of-service virtual-channels default-vc
set class-of-service virtual-channel-groups wan-vc-group branch1-vc scheduler-map
  bestscheduler
set class-of-service virtual-channel-groups wan-vc-group branch2-vc scheduler-map
  bestscheduler
set class-of-service virtual-channel-groups wan-vc-group branch3-vc scheduler-map
  bestscheduler
set class-of-service virtual-channel-groups wan-vc-group default-vc scheduler-map
  bestscheduler
set class-of-service virtual-channel-groups wan-vc-group default-vc default
set class-of-service virtual-channel-groups wan-vc-group branch1-vc shaping-rate
  1500000
set class-of-service virtual-channel-groups wan-vc-group branch2-vc shaping-rate
  1500000
set class-of-service virtual-channel-groups wan-vc-group branch3-vc shaping-rate
  1500000
set class-of-service interfaces t3-1/0/0 unit 0 virtual-channel-group wan-vc-group
set firewall family inet filter choose-vc term branch1 from destination-address
  192.168.10.0/24
set firewall family inet filter choose-vc term branch1 then virtual-channel branch1-vc
set firewall family inet filter choose-vc term branch1 then accept
set firewall family inet filter choose-vc term branch2 from destination-address
  192.168.20.0/24
set firewall family inet filter choose-vc term branch2 then virtual-channel branch2-vc
set firewall family inet filter choose-vc term branch2 then accept
set firewall family inet filter choose-vc term branch3 from destination-address
  192.168.30.0/24
set firewall family inet filter choose-vc term branch3 then virtual-channel branch3-vc
set firewall family inet filter choose-vc term branch3 then accept
```


Copyright © 2017, Juniper Networks, Inc.

```
set firewall family inet filter choose-vc term default then virtual-channel default-vc
set firewall family inet filter choose-vc term default then accept
set interfaces t3-1/0/0 unit 0 family inet filter output choose-vc
```

Step-by-Step Procedure

The following example requires you to navigate various levels in the configuration hierarchy. For instructions on how to do that, see *Using the CLI Editor in Configuration Mode* in the *CLI User Guide*.

To configure virtual channels:

1. Define the virtual channels and the default virtual channel.

```
[edit]
user@host# edit class-of-service
user@host# set virtual-channels branch1-vc
user@host# set virtual-channels branch2-vc
user@host# set virtual-channels branch3-vc
user@host# set virtual-channels default-vc
```

2. Define the virtual channel group and assign each virtual channel a scheduler map.

```
[edit class-of-service]
user@host# set virtual-channel-groups wan-vc-group branch1-vc scheduler-map
  bestscheduler
user@host# set virtual-channel-groups wan-vc-group branch2-vc scheduler-map
  bestscheduler
user@host# set virtual-channel-groups wan-vc-group branch3-vc scheduler-map
  bestscheduler
user@host# set virtual-channel-groups wan-vc-group default-vc scheduler-map
  bestscheduler
user@host# set virtual-channel-groups wan-vc-group default-vc default
```

3. Specify a shaping rate.

```
[edit class-of-service]
user@host# set virtual-channel-groups wan-vc-group branch1-vc shaping-rate 1.5m
user@host# set virtual-channel-groups wan-vc-group branch2-vc shaping-rate
  1.5m
user@host# set virtual-channel-groups wan-vc-group branch3-vc shaping-rate
  1.5m
```

4. Apply the virtual channel group to the logical interface.

```
[edit class-of-service]
user@host# set interfaces t3–1/0/0 unit 0 virtual-channel-group wan-vc-group
```

5. Create the firewall filter to select the traffic.

```
[edit firewall]
user@host# set firewall family inet filter choose-vc term branch1 from
  destination-address 192.168.10.0/24
user@host# set firewall family inet filter choose-vc term branch1 then virtual-channel
  branch1-vc
user@host# set firewall family inet filter choose-vc term branch1 then accept
```

Copyright © 2017, Juniper Networks, Inc.

```
user@host# set firewall family inet filter choose-vc term branch2 from
  destination-address 192.168.20.0/24
user@host# set firewall family inet filter choose-vc term branch2 then virtual-channel
  branch2-vc
user@host# set firewall family inet filter choose-vc term branch2 then accept
user@host# set firewall family inet filter choose-vc term branch3 from
  destination-address 192.168.30.0/24
user@host# set firewall family inet filter choose-vc term branch3 then virtual-channel
  branch3-vc
user@host# set firewall family inet filter choose-vc term branch3 then accept
user@host# set firewall family inet filter choose-vc term default then virtual-channel
  default-vc
user@host# set firewall family inet filter choose-vc term default then accept
```

6. Apply the firewall filter to output traffic.

```
[edit interfaces]
user@host# set t3–1/0/0 unit 0 family inet filter output choose-vc
```

Results From configuration mode, confirm your configuration by entering the **show class-of-service**, **show firewall**, and **show interfaces t3-1/0/0** commands. If the output does not display the intended configuration, repeat the configuration instructions in this example to correct it.

```
user@host# show class-of-service
virtual-channels {
  branch1-vc;
  branch2-vc;
  branch3-vc;
  default-vc;
}
virtual-channel-groups {
  wan-vc-group {
    branch1-vc {
    scheduler-map bestscheduler;
    shaping-rate 1500000;
  }
  branch2-vc {
    scheduler-map bestscheduler;
    shaping-rate 1500000;
  }
  branch3-vc {
    scheduler-map bestscheduler;
    shaping-rate 1500000;
  }
  default-vc {
    scheduler-map bestscheduler;
    default;
  }
}
}
interfaces {
  t3-1/0/0 {
```

 Copyright © 2017, Juniper Networks, Inc.

```
      unit 0 {
        virtual-channel-group wan-vc-group;
      }
    }
}
[edit]
user@host# show firewall
family inet {
    filter choose-vc {
      term branch1 {
        from {
          destination-address {
            192.168.10.0/24;
          }
        }
        then {
          virtual-channel branch1-vc;
          accept;
        }
      }
      term branch2 {
        from {
          destination-address {
            192.168.20.0/24;
          }
        }
        then {
          virtual-channel branch2-vc;
          accept;
        }
      }
      term branch3 {
        from {
          destination-address {
            192.168.30.0/24;
          }
        }
        then {
          virtual-channel branch3-vc;
          accept;
        }
      }
      term default {
        then {
          virtual-channel default-vc;
          accept;
        }
      }
    }
}
[edit]
user@host# show interfaces t3-1/0/0
unit 0 {
    family inet {
      filter {
        output choose-vc;
```

Copyright © 2017, Juniper Networks, Inc.

```
    }
  }
}
```

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

Verification

Verifying Virtual Channel Configuration

Purpose Verify that the virtual channels are properly configured.

Action From configuration mode, enter the **show class-of-service**, **show firewall**, and **show interfaces t3-1/0/0** commands.

Related Documentation

- Virtual Channels Overview on page 147
- Understanding Virtual Channels on page 148

Copyright © 2017, Juniper Networks, Inc.

CHAPTER 11

Enabling Queuing for Tunnel Interfaces

- CoS Queuing for Tunnels Overview on page 155
- Understanding the ToS Value of a Tunnel Packet on page 159
- Example: Configuring CoS Queuing for GRE or IP-IP Tunnels on page 160
- Copying Outer IP Header DSCP and ECN to Inner IP Header on page 164

CoS Queuing for Tunnels Overview

On an SRX Series device running Junos OS, a tunnel interface is an internal interface and supports many of the same CoS features as a physical interface. The tunnel interface creates a virtual point-to-point link between two SRX Series devices at remote points over an IP network.

For example, you can configure CoS features for generic routing encapsulation (GRE) and IP-IP tunnel interfaces. Tunneling protocols encapsulate packets inside a transport protocol.

GRE and IP-IP tunnels are used with services like IPsec and NAT to set up point-to-point VPNs. Junos OS allows you to enable CoS queuing, scheduling, and shaping for traffic exiting through these tunnel interfaces. For an example of configuring CoS Queuing for GRE tunnels, see "Example: Configuring CoS Queuing for GRE or IP-IP Tunnels" on page 160.

Starting with Junos OS Release 15.1X49-D100, you can configure CoS on logical tunnels for SRX 300, SRX320, SRX340, SRX 345, and SRX 550M devices.

This topic includes the following sections:

- Benefits of CoS Queuing for Tunnel Interfaces on page 156
- Configuring CoS on Logical Tunnels on page 156
- How CoS Queuing Works on page 158
- Limitations on CoS Shapers for Tunnel Interfaces on page 159

Copyright © 2017, Juniper Networks, Inc.

Benefits of CoS Queuing for Tunnel Interfaces

CoS queuing enabled for tunnel interfaces has the following benefits:

- Segregates tunnel traffic.

  Each tunnel can be shaped so that a tunnel with low-priority traffic cannot flood other tunnels that carry high-priority traffic.

  Traffic for one tunnel does not impact traffic on other tunnels.

- Controls tunnel bandwidth.

  Traffic through various tunnels is limited to not exceed a certain bandwidth.

  For example, suppose you have three tunnels to three remote sites through a single WAN interface. You can select CoS parameters for each tunnel such that traffic to some sites gets more bandwidth than traffic to other sites.

- Customizes CoS policies.

  You can apply different queuing, scheduling, and shaping policies to different tunnels based on user requirements. Each tunnel can be configured with different scheduler maps, different queue depths, and so on. Customization allows you to configure granular CoS policy providing for better control over tunnel traffic.

- Prioritizes traffic before it enters a tunnel.

  For example, CoS queuing avoids having low-priority packets scheduled ahead of high-priority packets when the interface speed is higher than the tunnel traffic speed. This feature is most useful when combined with IPsec. Typically, IPsec processes packets in a FIFO manner. However, with CoS queuing each tunnel can prioritize high-priority packets over low-priority packets. Also, each tunnel can be shaped so that a tunnel with low-priority traffic does not flood tunnels carrying high-priority traffic.

Configuring CoS on Logical Tunnels

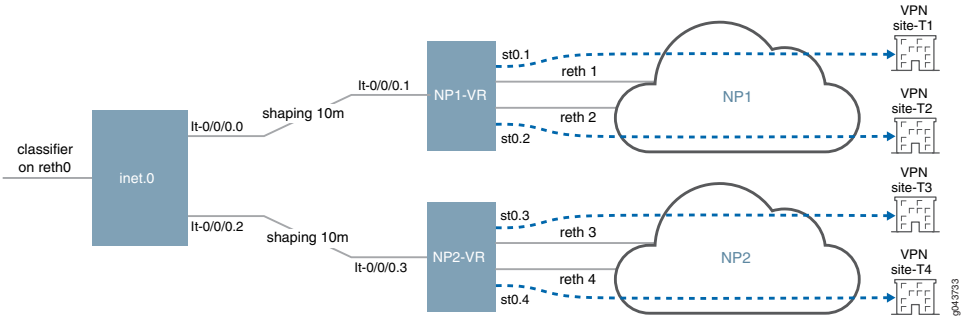
CoS has four typical scenarios that allow connection with remote sites using secure tunnels. However different secure tunnels may connect using different reth interfaces to different Network Providers (NP). For a specific NP, limited uplink bandwidth may be used to prioritize high-priority business and to avoid blindly dropping traffic at the NP side. Currently CoS does not support queuing across physical interfaes (IFD). Having a shared policer does not work as well as queuing, the policer may drop high-priority traffic regardless of priority. To support queuing on an IFD to enable CoS features to prioritize queuing and shaping requires logical tunnel (LT) and NP configuration.

You must define a pair of logical tunnels that are one-to-one mapped to NPs and redirect traffic with routing to the LT interface before encrypting the traffic through a secure tunnel.

For example, configure lt-0/0/0.0 and lt-0/0/0.1 to connect inet 0 and NP1 (virtual router) and configure a static route to redirect traffic to NP1 as lt-0/0/0.0 next-hop. Because NP1 has 10mbps bandwidth for upstream traffic, lt-0/00.0 can be configured with 10mbps of bandwidth shaping. See Figure 8 on page 157.

 Copyright © 2017, Juniper Networks, Inc.

Figure 8: CoS Solutions Using Logical Tunnels




```
routing-instances {
    NP1 {
        instance-type virtual-router;
        interface lt-0/0/0.1;
        interface lo0.0;
        interface reth1.0;
        interface reth2.0;
        interface st0.1;
        interface st0.2;
        routing-options {
            static {
                route 59.200.200.1/32 next-hop <next-hop addr of ipsec tunnel
st0.1>;
                route 59.200.200.2/32 next-hop <next-hop addr of ipsec tunnel
st0.2>;
                route 60.60.60.1/32 next-hop st0.1;
                route 60.60.60.2/32 next-hop st0.2;
                route 58.58.58.1/32 next-hop lt-0/0/0.1;
                route 58.58.58.2/32 next-hop lt-0/0/0.1;
            }
        }
    }
    NP2 {
        instance-type virtual-router;
        interface lt-0/0/0.3;
        interface lo0.1;
        interface reth3.0;
        interface reth4.0;
        interface st0.3;
        interface st0.4;
        routing-options {
            static {
                route 59.200.200.3/32 next-hop <next-hop addr of ipsec tunnel
st0.3>;
                route 59.200.200.4/32 next-hop <next-hop addr of ipsec tunnel
st0.4>;
                route 60.60.60.3/32 next-hop st0.3;
                route 60.60.60.4/32 next-hop st0.4;
                route 58.58.58.3/32 next-hop lt-0/0/0.3;
                route 58.58.58.4/32 next-hop lt-0/0/0.3;
            }
        }
    }
}
```

Copyright © 2017, Juniper Networks, Inc.

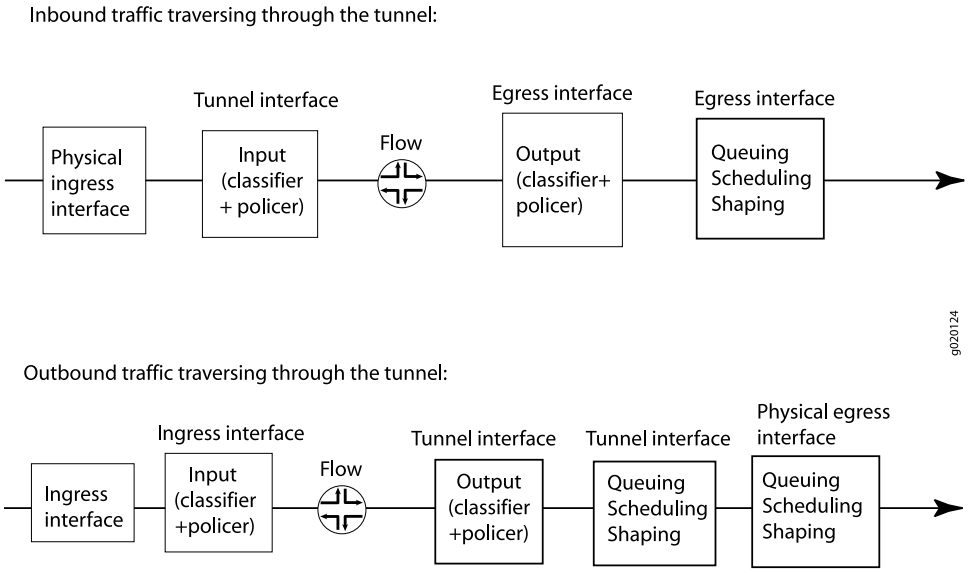
```
routing-options {
    static {
        route 60.60.60.1/32 next-hop lt-0/0/0.0;
        route 60.60.60.2/32 next-hop lt-0/0/0.0;
        route 60.60.60.3/32 next-hop lt-0/0/0.2;
        route 60.60.60.4/32 next-hop lt-0/0/0.2;
    }
}

class-of-service {
    interfaces {
        lt-0/0/0 {
            unit 0 {
                shaping-rate 10m;
            }
            unit 2 {
                shaping-rate 10m;
            }
}
```

How CoS Queuing Works

Figure 9 on page 158 shows CoS-related processing that occurs for traffic entering and exiting a tunnel. For information on flow-based packet processing, see the *Flow-Based and Packet-Based Processing Feature Guide for Security Devices*.

Figure 9: CoS Processing for Tunnel Traffic



Copyright © 2017, Juniper Networks, Inc.

Limitations on CoS Shapers for Tunnel Interfaces

When defining a CoS shaping rate on a tunnel interface, be aware of the following restrictions:

- The shaping rate on the tunnel interface must be less than that of the physical egress interface.
- The shaping rate only measures the packet size that includes the Layer 3 packet with GRE or IP-IP encapsulation. The Layer 2 encapsulation added by the physical interface is not factored into the shaping rate measurement.
- The CoS behavior works as expected only when the physical interface carries the shaped GRE or IP-IP tunnel traffic alone. If the physical interface carries other traffic, thereby lowering the available bandwidth for tunnel interface traffic, the CoS features do not work as expected.
- You cannot configure a logical interface shaper and a virtual circuit shaper simultaneously on the router. If virtual circuit shaping is desired, do not define a logical interface shaper. Instead, define a shaping rate for all the virtual circuits.

Understanding the ToS Value of a Tunnel Packet

To ensure that the tunneled packet continues to have the same CoS treatment even in the physical interface, you must preserve the type-of-service (ToS) value from the inner IP header to the outer IP header.

For transit traffic, Junos OS preserves the CoS value of the tunnel packet for both GRE and IP-IP tunnel interfaces. The inner IPv4 or IPv6 ToS bits are copied to the outer IPv4 ToS header for both types of tunnel interfaces.

For Routing Engine traffic, however, the router handles GRE tunnel interface traffic differently from IP-IP tunnel interface traffic. Unlike for IP-IP tunnels, the IPv4 ToS bits are not copied to the outer IPv4 header by default. You have a configuration option to copy the ToS value from the packet's inner IPv4 header to the outer IPv4 header.

To copy the inner ToS bits to the outer IP header (which is required for some tunneled routing protocols) on packets sent by the Routing Engine, include the **copy-tos-to-outer-ip-header** statement at the logical unit hierarchy level of a GRE interface.

NOTE: For IPv6 traffic, the inner ToS value is not copied to the outer IPv4 header for both GRE and IP-IP tunnel interfaces even if the **copy-tos-to-outer-ip-header** statement is specified.

This example copies the inner ToS bits to the outer IP header on a GRE tunnel:

```
[edit interfaces]
gr-0/0/0 {
   unit 0 {
      copy-tos-to-outer-ip-header;
```

Copyright © 2017, Juniper Networks, Inc.

```
        family inet;
    }
}
```

Related Documentation

- CoS Queuing for Tunnels Overview on page 155
- Example: Configuring CoS Queuing for GRE or IP-IP Tunnels on page 160

Example: Configuring CoS Queuing for GRE or IP-IP Tunnels

This example shows how to configure CoS queuing for GRE or IP-IP tunnels.

- Requirements on page 160
- Overview on page 160
- Configuration on page 161
- Verification on page 162

Requirements

Before you begin:

- Establish a main office and a branch office connected by a VPN using GRE or IP-IP tunneled interfaces.
- Configure forwarding classes and schedulers. See "Example: Assigning Forwarding Classes to Output Queues" on page 64 and "Example: Configuring Class-of-Service Schedulers" on page 94.
- Configure a scheduler map and apply the scheduler map to the tunnel interface. See "Example: Configuring and Applying Scheduler Maps" on page 109.
- Configure classifiers and apply them to the tunnel interface. See "Example: Configuring Behavior Aggregate Classifiers" on page 19.
- Create rewrite rules and apply them to the tunnel interface. See "Example: Configuring and Applying Rewrite Rules" on page 77.

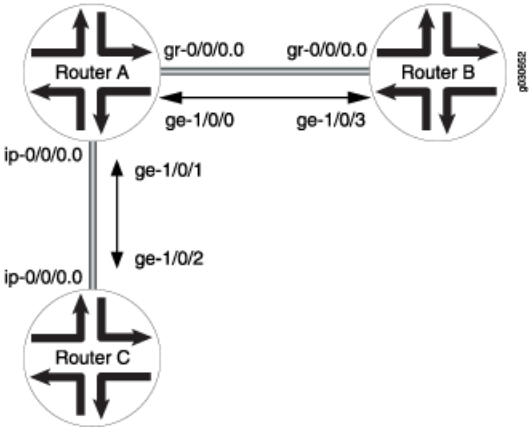
Overview

In this example, you enable tunnel queuing, define the GRE tunnel interface as gr-0/0/0, (Alternatively, you could define the IP-IP tunnel interface as ip-0/0/0.) and set the per unit scheduler. You then set the GRE tunnel's line rate as 100 Mbps by using the shaper definition.

In Figure 10 on page 161, Router A has a GRE tunnel established with Router B through interface **ge-1/0/0**. Router A also has an IP-IP tunnel established with Router C through interface **ge-1/0/1**. Router A is configured so that tunnel-queuing is enabled. Router B and Router C do not have tunnel-queuing configured.

Copyright © 2017, Juniper Networks, Inc.

Figure 10: Configuring CoS Queuing for GRE Tunnels



Configuration

CLI Quick Configuration

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

```
set chassis fpc 0 pic 0 tunnel-queuing
set interfaces gr-0/0/0 unit 0
set interfaces gr-0/0/0 per-unit-scheduler
set class-of-services interfaces gr-0/0/0 unit 0 shaping-rate 100m
```

Step-by-Step Procedure

To configure CoS queuing for GRE tunnels:

1. Enable tunnel queuing on the device.

   ```
   [edit]
   user@host# set chassis fpc 0 pic 0 tunnel-queuing
   ```

2. Define the GRE tunnel interface.

   ```
   [edit]
   user@host# set interfaces gr-0/0/0 unit 0
   ```

3. Define the per-unit scheduler for the GRE tunnel interface.

   ```
   [edit]
   user@host# set interfaces gr-0/0/0 per-unit-scheduler
   ```

4. Define the GRE tunnel's line rate by using the shaper definition.

   ```
   [edit]
   user@host# set class-of-services interfaces gr-0/0/0 unit 0 shaping-rate 100m
   ```

Copyright © 2017, Juniper Networks, Inc.

Results From configuration mode, confirm your configuration by entering the **show class-of-service interfaces gr-0/0/0**, **show interfaces gr-0/0/0** , and **show chassis** commands. If the output does not display the intended configuration, repeat the configuration instructions in this example to correct it.

```
[edit]
  user@host# show class-of-service interfaces gr-0/0/0
  unit 0 {
  shaping-rate 100m;
  }
  [edit]
    user@host# show interfaces gr-0/0/0
    per-unit-scheduler;
  unit 0;
  [edit]
    user@host# show chassis
    fpc 0 {
    pic 0 {
    tunnel-queuing;
  }
}
```

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

Verification

Confirm that the configuration is working properly.

- Verifying a CoS Queuing for GRE Tunnel Configuration on page 162
- Verifying a CoS Queuing for IP-IP Tunnel Configuration on page 164

Verifying a CoS Queuing for GRE Tunnel Configuration

Purpose Verify that the device is configured properly for tunnel configuration.

Action From configuration mode, enter the **show interfaces queue gr-0/0/0.0** command.

NOTE: If you enter gr-0/0/0.0 only, queue information for all tunnels is displayed. If you enter gr-0/0/0.0, queue information for the specific tunnel is displayed.

```
user@host> show interfaces queue gr-0/0/0.0
Logical interface gr-0/0/0.0 (Index 68) (SNMP ifIndex 112)
Forwarding classes: 8 supported, 4 in use
Egress queues: 8 supported, 4 in use Burst size: 0
Queue: 0, Forwarding classes: VOICE
  Queued:
    Packets              :              7117734                 7998 pps
    Bytes                :            512476848              4606848 bps
  Transmitted:
    Packets              :              4548146                 3459 pps
```


Copyright © 2017, Juniper Networks, Inc.

```
    Bytes                :            327466512              1992912 bps
    Tail-dropped packets :                    0                    0 pps
    RED-dropped packets  :              2569421                 4537 pps
     Low                 :                    0                    0 pps
     Medium-low          :                    0                    0 pps
     Medium-high         :                    0                    0 pps
     High                :              2569421                 4537 pps
    RED-dropped bytes    :            184998312              2613640 bps
     Low                 :                    0                    0 pps
     Medium-low          :                    0                    0 pps
     Medium-high         :                    0                    0 pps
     High                :            184998312              2613640 bps
Queue: 1, Forwarding classes: GOLD
  Queued:
    Packets              :               117600                    0 pps
    Bytes                :              8467200                    0 bps
  Transmitted:
    Packets              :               102435                    0 pps
    Bytes                :              7375320                    0 bps
    Tail-dropped packets :                    0                    0 pps
    RED-dropped packets  :               15165                    0 pps
     Low                 :                    0                    0 pps
     Medium-low          :                    0                    0 pps
     Medium-high         :                    0                    0 pps
     High                :               15165                    0 pps
    RED-dropped bytes    :             1091880                    0 bps
     Low                 :                    0                    0 pps
     Medium-low          :                    0                    0 pps
     Medium-high         :                    0                    0 pps
     High                :             1091880                    0 bps
Queue: 2, Forwarding classes: SILVER
  Queued:
    Packets              :                    0                    0 pps
    Bytes                :                    0                    0 bps
  Transmitted:
    Packets              :                    0                    0 pps
    Bytes                :                    0                    0 bps
    Tail-dropped packets :                    0                    0 pps
    RED-dropped packets  :                    0                    0 pps
     Low                 :                    0                    0 pps
     Medium-low          :                    0                    0 pps
     Medium-high         :                    0                    0 pps
     High                :                    0                    0 pps
    RED-dropped bytes    :                    0                    0 bps
     Low                 :                    0                    0 pps
     Medium-low          :                    0                    0 pps
     Medium-high         :                    0                    0 pps
     High                :                    0                    0 pps
Queue: 3, Forwarding classes: BRONZE
  Queued:
    Packets              :                    0                    0 pps
    Bytes                :                    0                    0 bps
  Transmitted:
    Packets              :                    0                    0 pps
    Bytes                :                    0                    0 bps
    Tail-dropped packets :                    0                    0 pps
    RED-dropped packets  :                    0                    0 pps
     Low                 :                    0                    0 pps
     Medium-low          :                    0                    0 pps
     Medium-high         :                    0                    0 pps
     High                :                    0                    0 pps
```

```
RED-dropped bytes     :                0                0 bps
 Low                  :                0                0 bps
 Medium-low           :                0                0 bps
 Medium-high          :                0                0 bps
 High                 :                0                0 bps
```

Verifying a CoS Queuing for IP-IP Tunnel Configuration

Purpose Verify that the device is configured properly for tunnel configuration.

Action From configuration mode, enter the **show interfaces queue ip-0/0/0.0** command.

NOTE: If you enter ip-0/0/0.0 only, queue information for all tunnels is displayed. If you enter ip-0/0/0.0, queue information for the specific tunnel is displayed.

Related Documentation

- CoS Queuing for Tunnels Overview on page 155
- Understanding the ToS Value of a Tunnel Packet on page 159

Copying Outer IP Header DSCP and ECN to Inner IP Header

Starting in Junos OS Release 15.1X49-D30, copying of a Differentiated Services Code Point (DSCP) (outer DSCP+ECN) field from the outer IP header encrypted packet to the inner IP header plain text message on the decryption path is supported.

The benefit in enabling this feature is that after IPsec decryption, clear text packets can follow the inner CoS (DSCP+ECN) rules.

This feature supports chassis cluster and also supports IPv6 and IPv4. The following are supported:

- Copying outer IPv4 DSCP and Explicit Congestion Notification (ECN) field to inner IPv4 DSCP and ECN field
- Copying outer IPv6 DSCP and ECN field to inner IPv6 DSCP and ECN field
- Copying outer IPv4 DSCP and ECN field to inner IPv6 DSCP and ECN field
- Copying outer IPv6 DSCP and ECN field to inner IPv4 DSCP and ECN field

By default this feature is disabled. When you enable this feature on a VPN object, the corresponding IPsec security Association (SA) is cleared and reestablished.

- To enable the feature:

  set security ipsec vpn *vpn-name* copy-outer-dscp

- To disable the feature:

Copyright © 2017, Juniper Networks, Inc.

delete security ipsec vpn *vpn-name* copy-outer-dscp

- To verify whether the feature is enabled or not:

  show security ipsec security-associations detail

Release History Table

Release	Description
15.1X49-D30	Starting in Junos OS Release 15.1X49-D30, copying of a Differentiated Services Code Point (DSCP) (outer DSCP+ECN) field from the outer IP header encrypted packet to the inner IP header plain text message on the decryption path is supported.

Related Documentation

- *VPN Feature Guide for Security Devices*
- *show security ipsec security-associations*

Copyright © 2017, Juniper Networks, Inc.

Copyright © 2017, Juniper Networks, Inc.

CHAPTER 12

Naming Components with Code-Point Aliases

- Code-Point Aliases Overview on page 167
- Default CoS Values and Aliases on page 168
- Example: Defining Code-Point Aliases for Bits on page 171

Code-Point Aliases Overview

A code-point alias assigns a name to a pattern of code-point bits. You can use this name instead of the bit pattern when you configure other class-of-service (CoS) components, such as classifiers, drop-profile maps, and rewrite rules.

When you configure classes and define classifiers, you can refer to the markers by alias names. You can configure user-defined classifiers in terms of alias names. If the value of an alias changes, it alters the behavior of any classifier that references it.

The following types of code points are supported by Junos operating system (OS):

- DSCP—Defines aliases for DiffServ code point (DSCP) IPv4 values.

  You can refer to these aliases when you configure classes and define classifiers.
- DSCP-IPv6—Defines aliases for DSCP IPv6 values.

  You can refer to these aliases when you configure classes and define classifiers.
- EXP—Defines aliases for MPLS EXP bits.

  You can map MPLS EXP bits to the device forwarding classes.
- inet-precedence—Defines aliases for IPv4 precedence values.

  Precedence values are modified in the IPv4 type-of-service (ToS) field and mapped to values that correspond to levels of service.

Related Documentation

- Default CoS Values and Aliases on page 168
- Example: Defining Code-Point Aliases for Bits on page 171

Copyright © 2017, Juniper Networks, Inc.

Default CoS Values and Aliases

Table 35 on page 169 shows the default mapping between the standard aliases and the bit values.

Copyright © 2017, Juniper Networks, Inc.

Table 35: Standard CoS Aliases and Bit Values

CoS Value Type	Alias	Bit Value
MPLS EXP	be	000
	be1	001
	ef	010
	ef1	011
	af11	100
	af12	101
	nc1/cs6	110
	nc2/cs7	111

Copyright © 2017, Juniper Networks, Inc.

Table 35: Standard CoS Aliases and Bit Values *(continued)*

CoS Value Type	Alias	Bit Value
DSCP and DSCP IPv6	ef	101110
	af11	001010
	af12	001100
	af13	001110
	af21	010010
	af22	010100
	af23	010110
	af31	011010
	af32	011100
	af33	011110
	af41	100010
	af42	100100
	af43	100110
	be	000000
	cs1	001000
	cs2	010000
	cs3	011000
	cs4	100000
	cs5	101000
	nc1/cs6	110000
	nc2/cs7	111000

Copyright © 2017, Juniper Networks, Inc.

Table 35: Standard CoS Aliases and Bit Values *(continued)*

CoS Value Type	Alias	Bit Value
IEEE 802.1	be	000
	be1	001
	ef	010
	ef1	011
	af11	100
	af12	101
	nc1/cs6	110
	nc2/cs7	111
IP precedence	be	000
	be1	001
	ef	010
	ef1	011
	af11	100
	af12	101
	nc1/cs6	110
	nc2/cs7	111

Related Documentation

- Code-Point Aliases Overview on page 167
- Example: Defining Code-Point Aliases for Bits on page 171

Example: Defining Code-Point Aliases for Bits

This example shows how to define code-point aliases for bits on a device.

- Requirements on page 172
- Overview on page 172
- Configuration on page 172
- Verification on page 172

Copyright © 2017, Juniper Networks, Inc.

Requirements

Before you begin, determine which default mapping to use. See "Default CoS Values and Aliases" on page 168.

Overview

In this example, you configure class of service and specify names and values for the CoS code-point aliases that you want to configure. Finally, you specify CoS value using the appropriate formats.

Configuration

Step-by-Step Procedure

The following example requires you to navigate various levels in the configuration hierarchy. For instructions on how to do that, see *Using the CLI Editor in Configuration Mode* in the *CLI User Guide*.

To define code-point aliases for bits on a device:

1. Configure class of service.

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

2. Specify CoS values.

```
[edit class-of-service]
user@host# set code-point-aliases dscp my1 110001
user@host# set code-point-aliases dscp my2 101110
user@host# set code-point-aliases dscp be 000001
user@host# set code-point-aliases dscp cs7 110000
```

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

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

Verification

To verify the configuration is working properly, enter the **show class-of-service code-point-aliases dscp** command.

Related Documentation

- Code-Point Aliases Overview on page 167

Copyright © 2017, Juniper Networks, Inc.

PART 3

Configuring Class of Service Scheduler Hierarchy

- Controlling Traffic by Configuring Scheduler Hierarchy on page 175

Copyright © 2017, Juniper Networks, Inc.

Copyright © 2017, Juniper Networks, Inc.

CHAPTER 13

Controlling Traffic by Configuring Scheduler Hierarchy

- Understanding Hierarchical Schedulers on page 175
- Understanding Internal Scheduler Nodes on page 178
- SRX1400, SRX3400, and SRX3600 Device Hardware Capabilities and Limitations on page 179
- Example: Configuring a Four-Level Scheduler Hierarchy on page 181
- Example: Controlling Remaining Traffic on page 193

Understanding Hierarchical Schedulers

Hierarchical schedules consist of nodes and queues. Queues terminate the CLI hierarchy. Nodes can be either root nodes, leaf nodes, or internal (non-leaf) nodes. Internal nodes are nodes that have other nodes as "children" in the hierarchy. For example, if an **interface-set** statement is configured with a logical interface (such as unit 0) and queue, then the **interface-set** is an internal node at level 2 of the hierarchy. However, if there are no traffic control profiles configured on logical interfaces, then the interface set is at level 3 of the hierarchy.

Table 36 on page 175 shows how the configuration of an interface set or logical interface affects the terminology of hierarchical scheduler nodes.

Table 36: Hierarchical Scheduler Nodes

Root Node (Level 1)	Internal Node (Level 2)	Leaf Node (Level 3)	Queue (Level 4)
Physical interface	Interface set	Logical interfaces	One or more queues
Physical interface	–	Interface set	One or more queues
Physical interface	–	Logical interfaces	One or more queues

When used, the interface set level of the hierarchy falls between the physical interface level (level 1) and the logical interface (level 3). Queues are always level 4 of the hierarchy. The schedulers hold the information about the queues, the last level of the hierarchy. In

Copyright © 2017, Juniper Networks, Inc.

all cases, the properties for level 4 of the hierarchical schedulers are determined by the scheduler map.

Hierarchical schedulers add CoS parameters to the new interface set level of the configuration. They use traffic control profiles to set values for parameters such as shaping rate (the peak information rate [PIR]), guaranteed rate (the committed information rate [CIR] on these interfaces), and scheduler maps (the queues and resources assigned to traffic).

The following CoS configuration places the following parameters in traffic control profiles at various levels:

- Traffic control profile at the port level (**tcp-port-level1**):
  - A shaping rate (PIR) of 100 Mbps
  - A delay buffer rate of 100 Mbps
- Traffic control profile at the interface set level (**tcp-interface-level2**):
  - A shaping rate (PIR) of 60 Mbps
  - A guaranteed rate (CIR) of 40 Mbps
- Traffic control profile at the logical interface level (**tcp-unit-level3**):
  - A shaping rate (PIR) of 50 Mbps
  - A guaranteed rate (CIR) of 30 Mbps
  - A scheduler map called smap1 to hold various queue properties (level 4)
  - A delay buffer rate of 40 Mbps

In this case, the traffic control profiles look like this:

```
[edit class-of-service traffic-control-profiles]
tcp-port-level1 { # This is the physical port level
   shaping-rate 100m;
   delay-buffer-rate 100m;
}
tcp-interface-level2 { # This is the interface set level
   shaping-rate 60m;
   guaranteed-rate 40m;
}
tcp-unit-level3 { # This is the logical interface level
   shaping-rate 50m;
   guaranteed-rate 30m;
   scheduler-map smap1;
   delay-buffer-rate 40m;
}
```

Once configured, the traffic control profiles must be applied to the proper places in the CoS interfaces hierarchy.

```
[edit class-of-service interfaces]
interface-set level-2 {
   output-traffic-control-profile tcp-interface-level-2;
```


Copyright © 2017, Juniper Networks, Inc.

```
}
ge-0/1/0 {
  output-traffic-control-profile tcp-port-level-1;
  unit 0 {
    output-traffic-control-profile tcp-unit-level-3;
  }
}
```

Interface sets can be defined as a list of logical interfaces, for example, unit 100, unit 200, and so on. Service providers can use these statements to group interfaces to apply scheduling parameters such as guaranteed rate and shaping rate to the traffic in the groups. Interface sets are currently only used by CoS, but they are applied at the **[edit interfaces]** hierarchy level so that they might be available to other services.

All traffic heading downstream must be gathered into an interface set with the **interface-set** statement at the [edit class-of-service interfaces] hierarchy level.

NOTE: Ranges are not supported; you must list each logical interface separately.

Although the interface set is applied at the [edit interfaces] hierarchy level, the CoS parameters for the interface set are defined at the [edit class-of-service interfaces] hierarchy level, usually with the **output-traffic-control-profile** ***profile-name*** statement.

You cannot specify an interface set mixing the logical interface, S-VLAN, or VLAN outer tag list forms of the **interface-set** statement. A logical interface can only belong to one interface set. If you try to add the same logical interface to different interface sets, the commit will fail.

This example will generate a commit error:

```
[edit interfaces]
interface-set set-one {
  ge-2/0/0 {
    unit 0;
    unit 2;
  }
}
interface-set set-two {
  ge-2/0/0 {
    unit 1;
    unit 3;
    unit 0; # COMMIT ERROR! Unit 0 already belongs to -set-one.
  }
}
```

Members of an interface set cannot span multiple physical interfaces. Only one physical interface is allowed to appear in an interface set.

This configuration is not supported:

```
[edit interfaces]
interface-set set-group {
```

Copyright © 2017, Juniper Networks, Inc.

```
    ge-0/0/1 {
      unit 0;
      unit 1;
    }
    ge-0/0/2 { # This type of configuration is NOT supported in the same interface set!
      unit 0;
      unit 1;
    }
  }
```

You can configure many logical interfaces under an interface. However, only a subset of them might have a traffic control profile attached. For example, you can configure three logical interfaces (units) over the same service VLAN, but you can apply a traffic control profile specifying best-effort and voice queues to only one of the logical interface units. Traffic from the two remaining logical interfaces is considered remaining traffic.

The scheduler map configured at individual interfaces (Level 3), interface sets (Level 2), or physical ports (Level 1), defines packet scheduling behavior at different levels. You can group logical interfaces in an interface set and configure the interfaces with scheduler maps. Any egress packet arriving at the physical or logical interfaces will be handled by the interface specific scheduler. If the scheduler map is not configured at the interface level, the packet will be handled by the scheduler configured at the interface set level or the port level.

Related Documentation

- Example: Configuring a Four-Level Scheduler Hierarchy on page 181
- Example: Controlling Remaining Traffic on page 193
- Understanding Internal Scheduler Nodes on page 178

Understanding Internal Scheduler Nodes

A node in the hierarchy is considered internal if either of the following conditions apply:

- One of its children nodes has a traffic control profile configured and applied.
- You configure the **internal-node** statement.

There are more resources available at the logical interface (unit) level than at the interface set level. It might be desirable to configure all resources at a single level, rather than spread over several levels. The **internal-node** statement provides this flexibility. This can be a helpful configuration device when interface-set queuing without logical interfaces is used exclusively on the interface.

You can use the **internal-node** statement to raise the interface set without children to the same level as the other configured interface sets with children, allowing them to compete for the same set of resources.

Using the **internal-node** statement allows statements to all be scheduled at the same level with or without children.

Copyright © 2017, Juniper Networks, Inc.

The following example makes the interface sets **if-set-1** and **if-set-2** internal:

```
[edit class-of-service interfaces ]
interface-set {
  if-set-1 {
    internal-node;
    output-traffic-control-profile tcp-200m-no-smap;
  }
  if-set-2 {
    internal-node;
    output-traffic-control-profile tcp-100m-no-smap;
  }
}
```

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

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

Related Documentation

- Understanding Hierarchical Schedulers on page 175
- Example: Configuring a Four-Level Scheduler Hierarchy on page 181
- Example: Controlling Remaining Traffic on page 193

SRX1400, SRX3400, and SRX3600 Device Hardware Capabilities and Limitations

For SRX1400, SRX3400, and SRX3600 devices, each Input/Output Card (IOC), Flexible PIC Concentrator (FPC), or IOC slot has only one Physical Interface Card (PIC), which contains either two 10-Gigabit Ethernet ports or sixteen 1-Gigabit Ethernet ports. Table 37 on page 179 shows the maximum number of cards and ports allowed in SRX1400, SRX3400, and SRX3600 devices.

NOTE: The number of ports the Network Processing Unit (NPU) needs to handle might be different from the fixed 10:1 port to NPU ratio for 1-Gigabit IOC, or the 1:1 ratio for the 10-Gigabit IOC that is needed on the SRX5600 and SRX5800 devices, leading to oversubscription on the SRX1400, SRX3400, and SRX3600 devices.

Platform support depends on the Junos OS release in your installation.

Table 37: Available NPCs and IO Ports for SRX1400, SRX3400, and SRX3600 Devices

System	IOCs	IO Ports	NPCs
SRX3600	7	108 (16 x 6 + 12)	3
SRX3400	5	76 (16 x 4 + 12)	2
SRX1400	2	28 (16 x 1 + 12)	1

Copyright © 2017, Juniper Networks, Inc.

SRX3400 and SRX3600 devices allow you to install up to three Network Processing Cards (NPCs). In a single NPC configuration, the NPC has to process all of the packets to and from each IOC. However, when there is more than one NPC available, an IOC will only exchange packets with a preassigned NPC. You can use the **set chassis ioc-npc-connectivity** CLI statement to configure the IOC-to-NPC mapping. By default, the mapping is assigned so that the load is shared equally among all NPCs. When the mapping is changed, for example, an IOC or NPC is removed, or you have mapped a specific NPC to an IOC, then the device has to be restarted.

NOTE: **SRX1400 devices support a single NPC or an NSPC combo card.**

For SRX1400, SRX3400, and SRX3600 devices, the IOC supports the following hierarchical scheduler characteristics:

- Level 1- Shaping at the physical interface (ifd)
- Level 2- Shaping and scheduling at the logical interface level (ifl)
- Level 3- Scheduling at the queue level

NOTE: **Interface set (iflset) is not supported for SRX1400, SRX3400, and SRX3600 devices.**

In SRX5600 and SRX5800 devices, an NPC supports 32 port-level shaping profiles at level 1, such that each front port can have its own shaping profile.

In SRX1400, SRX3400, and SRX3600 devices, an NPC supports only 16 port-level shaping profiles in the hardware, including two profiles that are predefined for 10-GB and 1-GB shaping rates. The user can configure up to 14 different levels of shaping rates. If more levels are configured, then the closest match found in the 16 profiles will be used instead.

For example, assume that a system is already configured with the following rates for ifd:

10 Mbps, 20 Mbps, 40 Mbps, 60 Mbps, 80 Mbps, 100 Mbps, 200 Mbps, 300 Mbps, 400 Mbps, 500 Mbps, 600 Mbps, 700 Mbps, 800 Mbps, 900 Mbps, 1 GB (predefined), 10 GB (predefined)

Each of these 16 rates is programmed into one of the 16 profiles in the hardware; then consider the following two scenarios:

- Scenario 1: If the user changes one port's shaping rate from 1 GB to 100 Mbps, which is already programmed in one of the 16 profiles, the profile with a 100 Mbps shaping rate will be used by the port.
- Scenario 2: If the user changes another port's shaping rate from 1 GB to 50 Mbps, which is not in the shaping profiles, the closest matching profile with a 60 Mbps shaping rate will be used instead.

Copyright © 2017, Juniper Networks, Inc.

When scenario 2 occurs, not all of the user-configured rates can be supported by the hardware. Even if more than 14 different rates are specified, only 14 will be programmed in the hardware. Which 14 rates are programmed in the hardware depends on many factors. For this reason, we recommend that you plan carefully and use no more than 14 levels of port-level shaping rates.

Each device supports Weighed Random Early Discard (WRED) at the port level, and each NPU has 512 MB of frame memory. Also, 10-Gigabit Ethernet ports get more buffers than the 1-Gigabit Ethernet ports. Buffer availability depends on how much bandwidth (number of NPCs, ports, 1 GB or 10 GB, and so on) the device has to support. The more bandwidth that the device has to support, the less buffer is available. When two NPCs are available, the amount of frame buffer available is doubled.

Related Documentation

- Understanding Hierarchical Schedulers on page 175
- Example: Configuring a Four-Level Scheduler Hierarchy on page 181
- Example: Controlling Remaining Traffic on page 193
- Understanding Internal Scheduler Nodes on page 178

Example: Configuring a Four-Level Scheduler Hierarchy

This example shows how to configure a 4-level hierarchy of schedulers.

- Requirements on page 181
- Overview on page 181
- Configuration on page 182
- Verification on page 193

Requirements

Before you begin:

- Review how to configure schedulers. See "Example: Configuring Class-of-Service Schedulers" on page 94.
- Review RED drop profiles. See *Understanding RED Drop Profiles*.
- Review how to configure and apply scheduler maps. See "Example: Configuring and Applying Scheduler Maps" on page 109.

Overview

The configuration parameters for this example are shown in Figure 11 on page 182. The queues are shown at the top of the figure with the other three levels of the hierarchy below.

Copyright © 2017, Juniper Networks, Inc.

Figure 11: Building a Scheduler Hierarchy

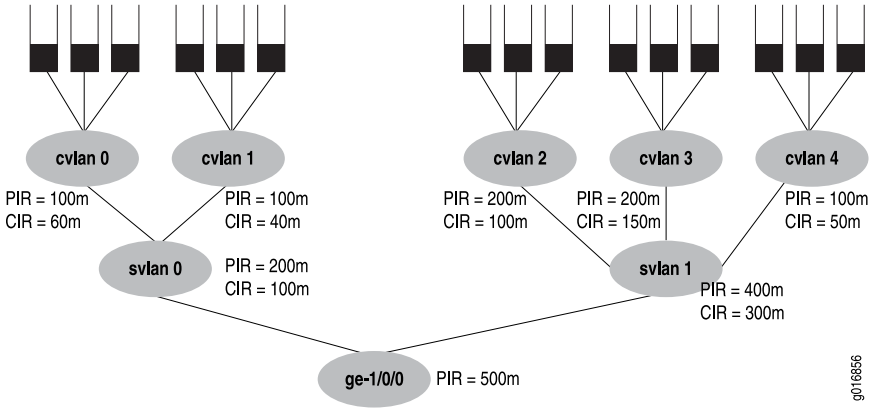


Figure 11 on page 182's PIR values will be configured as the shaping rates, and the CIRs will be configured as the guaranteed rate on the Ethernet interface **ge-1/0/0**. The PIR can be oversubscribed (that is, the sum of the children PIRs can exceed the parent's, as in **svlan 1**, where 200 + 200 + 100 exceeds the parent rate of 400). However, the sum of the children node level's CIRs must never exceed the parent node's CIR, as shown in all the service VLANs (otherwise, the guaranteed rate could never be provided in all cases).

NOTE: **Although a shaping rate can be applied directly to the physical interface, hierarchical schedulers must use a traffic control profile to hold the shaping rate parameter.**

The keyword to configure hierarchical schedulers is at the physical interface level, as are VLAN tagging and the VLAN IDs. In this example, the interface sets are defined by logical interfaces (units) and not outer VLAN tags. All VLAN tags in this example are customer VLAN tags.

The traffic control profiles in this example are for both the service VLAN level (logical interfaces) and the customer VLAN (VLAN tag) level.

This example shows all details of the CoS configuration for the **ge-1/0/0** interface in Figure 11 on page 182.

Configuration

This section contains the following topics:

- Configuring the Logical Interfaces on page 183
- Configuring the Interface Sets on page 184
- Applying an Interface Set on page 185
- Configuring the Traffic Control Profiles on page 185
- Configuring the Schedulers on page 187
- Configuring the Drop Profiles on page 189

Copyright © 2017, Juniper Networks, Inc.

- Configuring the Scheduler Maps on page 190
- Applying Traffic Control Profiles on page 191

Configuring the Logical Interfaces

CLI Quick Configuration

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

```
edit interface ge-1/0/0
set hierarchical-scheduler vlan-tagging unit 1 vlan-id 101
set hierarchical-scheduler vlan-tagging unit 1 vlan-id 101
set hierarchical-scheduler vlan-tagging unit 2 vlan-id 102
set hierarchical-scheduler vlan-tagging unit 3 vlan-id 103
set hierarchical-scheduler vlan-tagging unit 4 vlan-id 104
```

Step-by-Step Procedure

To configure the logical interfaces:

1. Create the logical interface.

   ```
   [edit]
   user@host# edit interface ge-1/0/0
   ```

2. Create the interface sets by defining the VLAN tagging and the VLAN IDs for each level.

   ```
   [edit interface ge-1/0/0]
   user@host# set hierarchical-scheduler vlan-tagging unit 0 vlan-id 100
   user@host# set hierarchical-scheduler vlan-tagging unit 1 vlan-id 101
   user@host# set hierarchical-scheduler vlan-tagging unit 2 vlan-id 102
   user@host# set hierarchical-scheduler vlan-tagging unit 3 vlan-id 103
   user@host# set hierarchical-scheduler vlan-tagging unit 4 vlan-id 104
   ```

Results

From configuration mode, confirm your configuration by entering the **show interface ge-1/0/0** command. If the output does not display the intended configuration, repeat the configuration instructions in this example to correct it.

```
[edit]
user@host# show interface ge-1/0/0
hierarchical-scheduler;
vlan-tagging;
unit 0 {
   vlan-id 100;
}
unit 1 {
   vlan-id 101;
}
unit 2 {
   vlan-id 102;
}
unit 3 {
```

Copyright © 2017, Juniper Networks, Inc.

```
    vlan-id 103;
  }
  unit 4 {
    vlan-id 104;
  }
```

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

Configuring the Interface Sets

CLI Quick Configuration

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

```
set class-of-service interfaces interface-set svlan-0 interface ge-1/0/0 unit 0
set class-of-service interfaces interface-set svlan-0 interface ge-1/0/0 unit 1
set class-of-service interfaces interface-set svlan-1 interface ge-1/0/0 unit 2
set class-of-service interfaces interface-set svlan-1 interface ge-1/0/0 unit 3
set class-of-service interfaces interface-set svlan-1 interface ge-1/0/0 unit 4
```

Step-by-Step Procedure

To configure the interface sets:

1. Create the first logical interface and its CoS parameters.

```
[edit class-of-service interfaces]
user@host# set interface-set svlan-0 interface ge-1/0/0 unit 0
user@host# set interface-set svlan-0 interface ge-1/0/0 unit 1
```

2. Create the second logical interface and its CoS parameters.

```
[edit class-of-service interfaces]
user@host# set interface-set svlan-1 interface ge-1/0/0 unit 2
user@host# set interface-set svlan-1 interface ge-1/0/0 unit 3
user@host# set interface-set svlan-1 interface ge-1/0/0 unit 4
```

Results

From configuration mode, confirm your configuration by entering the **show interfaces** command. If the output does not display the intended configuration, repeat the configuration instructions in this example to correct it.

```
[edit]
user@host# show interfaces
interface-set svlan-0 {
  interface ge-1/0/0 {
    unit 0;
    unit 1;
  }
}
interface-set svlan-1 {
  interface ge-1/0/0 {
    unit 2;
    unit 3;
```


Copyright © 2017, Juniper Networks, Inc.

```
    unit 4;
  }
}
```

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

Applying an Interface Set

CLI Quick Configuration

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

```
set class-of-service interfaces interface-set set-ge-0 output-traffic-control-profile tcp-set1
```

Step-by-Step Procedure

To apply an interface set:

1. Create the Ethernet interface set.

```
[edit class-of-service interfaces]
user@host# set interface-set set-ge-0
```

2. Apply a traffic control parameter to the Ethernet interface set.

```
[edit class-of-service interfaces interface-set set-ge-0]
user@host# set output-traffic-control-profile tcp-set1
```

Results

From configuration mode, confirm your configuration by entering the **show interfaces** command. If the output does not display the intended configuration, repeat the configuration instructions in this example to correct it.

```
[edit]
user@host# show interfaces
interface-set set-ge-0 {
  output-traffic-control-profile tcp-set1;
}
```

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

Configuring the Traffic Control Profiles

CLI Quick Configuration

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

```
set class-of-service traffic-control-profiles tcp-500m-shaping-rate shaping-rate 500m
set class-of-service traffic-control-profiles tcp-svlan0 shaping-rate 200m guaranteed-rate
  100m delay-buffer-rate 300m
set class-of-service traffic-control-profiles tcp-svlan1 shaping-rate 400m guaranteed-rate
  30100m delay-buffer-rate 100m
```

Copyright © 2017, Juniper Networks, Inc.

```
set class-of-service traffic-control-profiles tcp-cvlan0 shaping-rate 100m guaranteed-rate
  60m scheduler-map tcp-map-cvlan0
set class-of-service traffic-control-profiles tcp-cvlan1 shaping-rate 100m guaranteed-rate
  40m scheduler-map tcp-map-cvlan1
set class-of-service traffic-control-profiles tcp-cvlan2 shaping-rate 200m guaranteed-rate
  100m scheduler-map tcp-map-cvlanx
set class-of-service traffic-control-profiles tcp-cvlan3 shaping-rate 200m guaranteed-rate
  150m scheduler-map tcp-map-cvlanx
set class-of-service traffic-control-profiles tcp-cvlan4 shaping-rate 100m guaranteed-rate
  50m scheduler-map tcp-map-cvlanx
```

Step-by-Step Procedure

To configure the traffic control profiles:

1. Create the traffic profile parameters.

```
[edit class-of-service traffic-control-profiles]
user@host# tcp-500m-shaping-rate shaping-rate 500m
```

2. Create the traffic control profiles and parameters for the S-VLAN (logical interfaces) level.

```
[edit class-of-service traffic-control-profiles]
user@host# set tcp-svlan0 shaping-rate 200m guaranteed-rate 100m
  delay-buffer-rate 300m
user@host# set tcp-svlan1 shaping-rate 400m guaranteed-rate 30100m
  delay-buffer-rate 100m
```

3. Create the traffic control profiles and parameters for the C-VLAN (VLAN tags) level.

```
[edit class-of-service traffic-control-profiles]
user@host# set tcp-cvlan0 shaping-rate 100m guaranteed-rate 60m scheduler-map
  tcp-map-cvlan0
user@host# set tcp-cvlan1 shaping-rate 100m guaranteed-rate 40m scheduler-map
  tcp-map-cvlan1
user@host# set tcp-cvlan2 shaping-rate 200m guaranteed-rate 100m
  scheduler-map tcp-map-cvlanx
user@host# set tcp-cvlan3 shaping-rate 200m guaranteed-rate 150m scheduler-map
  tcp-map-cvlanx
user@host# set tcp-cvlan4 shaping-rate 100m guaranteed-rate 50m scheduler-map
  tcp-map-cvlanx
```

Results

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

```
[edit]
user@host# show class-of-service traffic-control-profiles
tcp-500m-shaping-rate {
  shaping-rate 500m;
}
tcp-svlan0 {
  shaping-rate 200m;
  guaranteed-rate 100m;
```


Copyright © 2017, Juniper Networks, Inc.

```
    delay-buffer-rate 300m; # This parameter is not shown in the figure
}
tcp-svlan1 {
    shaping-rate 400m;
    guaranteed-rate 300m;
    delay-buffer-rate 100m; # This parameter is not shown in the figure
}
tcp-cvlan0 {
    shaping-rate 100m;
    guaranteed-rate 60m;
    scheduler-map tcp-map-cvlan0; # This example applies scheduler maps to customer
        VLANs
}
tcp-cvlan1 {
    shaping-rate 100m;
    guaranteed-rate 40m;
    scheduler-map tcp-map-cvlan1; # This example applies scheduler maps to customer
        VLANs
}
tcp-cvlan2 {
    shaping-rate 200m;
    guaranteed-rate 100m;
    scheduler-map tcp-map-cvlanx; # This example applies scheduler maps to customer
        VLANs
}
tcp-cvlan3 {
    shaping-rate 200m;
    guaranteed-rate 150m;
    scheduler-map tcp-map-cvlanx; # This example applies scheduler maps to customer
        VLANs
}
tcp-cvlan4 {
    shaping-rate 100m;
    guaranteed-rate 50m;
    scheduler-map tcp-map-cvlanx; # This example applies scheduler maps to customer
        VLANs
}
```

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

Configuring the Schedulers

CLI Quick Configuration

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

```
set class-of-service schedulers sched-cvlan0-qx priority low transmit-rate 20m buffer-size
    temporal 100ms drop-profile-map loss-priority low dp-low drop-profile-map loss-priority
    high dp-high
set class-of-service schedulers sched-cvlan1-q0 priority high transmit-rate 20m buffer-size
    percent 40 drop-profile-map loss-priority low dp-low drop-profile-map loss-priority
    high dp-high
```

Copyright © 2017, Juniper Networks, Inc.

```
set class-of-service schedulers sched-cvlanx-qx transmit-rate percent 30 buffer-size
  percent 30 drop-profile-map loss-priority low dp-low drop-profile-map loss-priority
  high dp-high
set class-of-service schedulers sched-cvlan1-qx transmit-rate 10m buffer-size temporal
  100ms drop-profile-map loss-priority low dp-low drop-profile-map loss-priority high
  dp-high
```

Step-by-Step Procedure

To configure the schedulers:

1. Create the schedulers and their parameters.

```
[edit class-of-service schedulers]
user@host# set sched-cvlan0-qx priority low transmit-rate 20m buffer-size temporal
  100ms drop-profile-map loss-priority low dp-low drop-profile-map loss-priority
  high dp-high
user@host# set sched-cvlan1-q0 priority high transmit-rate 20m buffer-size percent
  40 drop-profile-map loss-priority low dp-low drop-profile-map loss-priority high
  dp-high
user@host# set sched-cvlanx-qx transmit-rate percent 30 buffer-size percent 30
  drop-profile-map loss-priority low dp-low drop-profile-map loss-priority high
  dp-high
user@host# set sched-cvlan1-qx transmit-rate 10m buffer-size temporal 100ms
  drop-profile-map loss-priority low dp-low drop-profile-map loss-priority high
  dp-high
```

Results

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

```
[edit]
user@host# show class-of-service schedulers
sched-cvlan0-qx {
  priority low;
  transmit-rate 20m;
  buffer-size temporal 100ms;
  drop-profile-map loss-priority low dp-low;
  drop-profile-map loss-priority high dp-high;
}
sched-cvlan1-q0 {
  priority high;
  transmit-rate 20m;
  buffer-size percent 40;
  drop-profile-map loss-priority low dp-low;
  drop-profile-map loss-priority high dp-high;
}
sched-cvlanx-qx {
  transmit-rate percent 30;
  buffer-size percent 30;
  drop-profile-map loss-priority low dp-low;
  drop-profile-map loss-priority high dp-high;
}
sched-cvlan1-qx {
  transmit-rate 10m;
  buffer-size temporal 100ms;
```


Copyright © 2017, Juniper Networks, Inc.

```
    drop-profile-map loss-priority low dp-low;
    drop-profile-map loss-priority high dp-high;
}
```

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

Configuring the Drop Profiles

CLI Quick Configuration

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

```
set class-of-service drop-profiles dp-low interpolate fill-level 80 drop-probability 80
set class-of-service drop-profiles dp-low interpolate fill-level 100 drop-probability 100
set class-of-service drop-profiles dp-high interpolate fill-level 60 drop-probability 80
set class-of-service drop-profiles dp-high interpolate fill-level 80 drop-probability 100
```

Step-by-Step Procedure

To configure the drop profiles:

1. Create the low drop profile.

```
[edit class-of-service drop-profiles]
user@host# set dp-low interpolate fill-level 80 drop-probability 80
user@host# set dp-low interpolate fill-level 100 drop-probability 100
```

2. Create the high drop profile.

```
[edit class-of-service drop-profiles]
user@host# set dp-high interpolate fill-level 60 drop-probability 80
user@host# set dp-high interpolate fill-level 80 drop-probability 100
```

Results

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

```
[edit]
user@host# show class-of-service drop-profiles
dp-low {
    interpolate fill-level 80 drop-probability 80;
    interpolate fill-level 100 drop-probability 100;
}
dp-high {
    interpolate fill-level 60 drop-probability 80;
    interpolate fill-level 80 drop-probability 100;
}
```

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

Copyright © 2017, Juniper Networks, Inc.

Configuring the Scheduler Maps

CLI Quick Configuration

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

```
set class-of-service scheduler-maps tcp-map-cvlan0 forwarding-class voice scheduler
  sched-cvlan0-qx
set class-of-service scheduler-maps tcp-map-cvlan0 forwarding-class video scheduler
  sched-cvlan0-qx
set class-of-service scheduler-maps tcp-map-cvlan0 forwarding-class data scheduler
  sched-cvlan0-qx
set class-of-service scheduler-maps tcp-map-cvlan1 forwarding-class voice scheduler
  sched-cvlan1-q0
set class-of-service scheduler-maps tcp-map-cvlan1 forwarding-class video scheduler
  sched-cvlan1-qx
set class-of-service scheduler-maps tcp-map-cvlan1 forwarding-class data scheduler
  sched-cvlan1-qx
set class-of-service scheduler-maps tcp-map-cvlanx forwarding-class voice scheduler
  sched-cvlanx-qx
set class-of-service scheduler-maps tcp-map-cvlanx forwarding-class video scheduler
  sched-cvlanx-qx
set class-of-service scheduler-maps tcp-map-cvlanx forwarding-class data scheduler
  sched-cvlanx-qx
```

Step-by-Step Procedure

To configure three scheduler maps:

1. Create the first scheduler map.

```
[edit class-of-service scheduler-maps]
user@host# set tcp-map-cvlan0 forwarding-class voice scheduler sched-cvlan0-qx
user@host# set tcp-map-cvlan0 forwarding-class video scheduler sched-cvlan0-qx
user@host# set tcp-map-cvlan0 forwarding-class data scheduler sched-cvlan0-qx
```

2. Create the second scheduler map.

```
[edit class-of-service scheduler-maps]
user@host# set tcp-map-cvlan1 forwarding-class voice scheduler sched-cvlan1-q0
user@host# set tcp-map-cvlan1 forwarding-class video scheduler sched-cvlan1-qx
user@host# set tcp-map-cvlan1 forwarding-class data scheduler sched-cvlan1-qx
```

3. Create the third scheduler map.

```
[edit class-of-service scheduler-maps]
user@host# set tcp-map-cvlanx forwarding-class voice scheduler sched-cvlanx-qx
user@host# set tcp-map-cvlanx forwarding-class video scheduler sched-cvlanx-qx
user@host# set tcp-map-cvlanx forwarding-class data scheduler sched-cvlanx-qx
```

Results

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

Copyright © 2017, Juniper Networks, Inc.

```
[edit]
user@host# show class-of-service scheduler-maps
tcp-map-cvlan0 {
   forwarding-class voice scheduler sched-cvlan0-qx;
   forwarding-class video scheduler sched-cvlan0-qx;
   forwarding-class data scheduler sched-cvlan0-qx;
}
tcp-map-cvlan1 {
   forwarding-class voice scheduler sched-cvlan1-q0;
   forwarding-class video scheduler sched-cvlan1-qx;
   forwarding-class data scheduler sched-cvlan1-qx;
}
tcp-map-cvlanx {
   forwarding-class voice scheduler sched-cvlanx-qx;
   forwarding-class video scheduler sched-cvlanx-qx;
   forwarding-class data scheduler sched-cvlanx-qx;
}
```

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

Applying Traffic Control Profiles

CLI Quick Configuration

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

```
set class-of-service interfaces ge-1/0/0 output-traffic-control-profile
   tcp-500m-shaping-rate unit 0 output-control-traffic-control-profile tcp-cvlan0
set class-of-service interfaces ge-1/0/0 output-traffic-control-profile
   tcp-500m-shaping-rate unit 1 output-control-traffic-control-profile tcp-cvlan1
set class-of-service interfaces ge-1/0/0 output-traffic-control-profile
   tcp-500m-shaping-rate unit 2 output-control-traffic-control-profile tcp-cvlan2
set class-of-service interfaces ge-1/0/0 output-traffic-control-profile
   tcp-500m-shaping-rate unit 3 output-control-traffic-control-profile tcp-cvlan3
set class-of-service interfaces ge-1/0/0 output-traffic-control-profile
   tcp-500m-shaping-rate unit 4 output-control-traffic-control-profile tcp-cvlan4
set class-of-service interfaces ge-1/0/0 output-traffic-control-profile
   tcp-500m-shaping-rate interface-set-svlan0 output-control-traffic-control-profile
   tcp-svlan0
set class-of-service interfaces ge-1/0/0 output-traffic-control-profile
   tcp-500m-shaping-rate interface-set-svlan1 output-control-traffic-control-profile
   tcp-svlan1
```

Step-by-Step Procedure

To apply traffic control profiles:

1. Set the interface.

   ```
   [edit class-of-service]
   user@host# set interfaces ge-1/0/0
   ```

2. Set the traffic control profiles for the C-VLANs.

   ```
   [edit class-of-service interfaces ge-1/0/0 ]
   ```

Copyright © 2017, Juniper Networks, Inc.

```
user@host# set output-traffic-control-profile tcp-500m-shaping-rate unit 0
  output-control-traffic-control-profile tcp-cvlan0
user@host# set output-traffic-control-profile tcp-500m-shaping-rate unit 1
  output-control-traffic-control-profile tcp-cvlan1
user@host# set output-traffic-control-profile tcp-500m-shaping-rate unit 2
  output-control-traffic-control-profile tcp-cvlan2
user@host# set output-traffic-control-profile tcp-500m-shaping-rate unit 3
  output-control-traffic-control-profile tcp-cvlan3
user@host# set output-traffic-control-profile tcp-500m-shaping-rate unit 4
  output-control-traffic-control-profile tcp-cvlan4
```

3. Set the traffic control profiles for the S-VLANs.

```
[edit class-of-service interfaces ge-1/0/0 ]
user@host# set output-traffic-control-profile tcp-500m-shaping-rate
  interface-set-svlan0 output-control-traffic-control-profile tcp-svlan0
user@host# set output-traffic-control-profile tcp-500m-shaping-rate
  interface-set-svlan1 output-control-traffic-control-profile tcp-svlan1
```

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

```
[edit]
user@host# show class-of-service interfaces
ge-1/0/0 {
  output-traffic-control-profile tcp-500m-shaping-rate;
  unit 0 {
    output-traffic-control-profile tcp-cvlan0;
  }
  unit 1 {
    output-traffic-control-profile tcp-cvlan1;
  }
  unit 2 {
    output-traffic-control-profile tcp-cvlan2;
  }
  unit 3 {
    output-traffic-control-profile tcp-cvlan3;
  }
  unit 4 {
    output-traffic-control-profile tcp-cvlan4;
  }
}
interface-set svlan0 {
  output-traffic-control-profile tcp-svlan0;
}
interface-set svlan1 {
  output-traffic-control-profile tcp-svlan1;
}
```

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

Copyright © 2017, Juniper Networks, Inc.

Verification

Verifying Scheduler Hierarchy Configuration

Purpose Verify that the scheduler hierarchy is configured properly.

Action From operational mode, enter the following commands:

- **show interface ge-1/0/0**
- **show class-of-service interfaces**
- **show class-of-service traffic-control-profiles**
- **show class-of-service schedulers**
- **show class-of-service drop-profiles**
- **show class-of-service scheduler-maps**

Related Documentation

- Understanding Hierarchical Schedulers on page 175
- Example: Controlling Remaining Traffic on page 193
- Understanding Internal Scheduler Nodes on page 178

Example: Controlling Remaining Traffic

This example shows how to control remaining traffic from the remaining logical interfaces.

- Requirements on page 193
- Overview on page 193
- Configuration on page 195
- Verification on page 198

Requirements

Before you begin:

- Review how to configure schedulers. See "Example: Configuring Class-of-Service Schedulers" on page 94.
- Review how to configure and apply scheduler maps. See "Example: Configuring and Applying Scheduler Maps" on page 109.

Overview

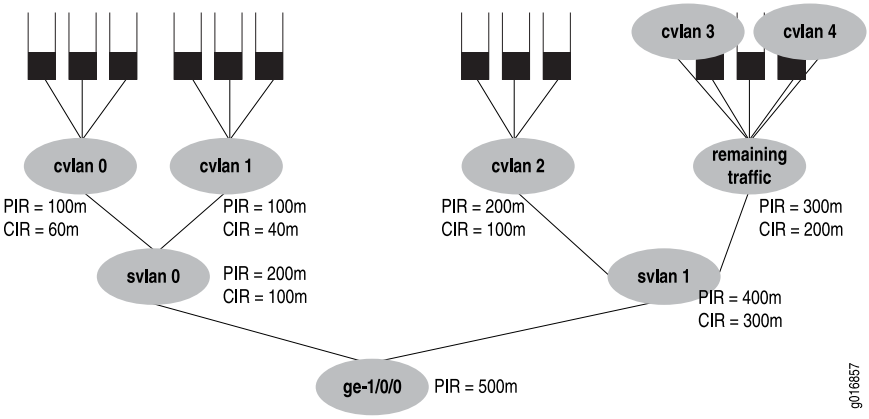
To configure transmit rate guarantees for the remaining traffic, you configure the **output-traffic-control-profile-remaining** statement specifying a guaranteed rate for the remaining traffic. Without this statement, the remaining traffic gets a default, minimal bandwidth. Similarly, you can specify the **shaping-rate** and **delay-buffer-rate** statements

Copyright © 2017, Juniper Networks, Inc.

in the traffic control profile referenced with the **output-traffic-control-profile-remaining** statement to shape and provide buffering for remaining traffic.

In the interface shown in Figure 12 on page 194, customer VLANs 3 and 4 have no explicit traffic control profile. However, the service provider might want to establish a shaping and guaranteed transmit rate for aggregate traffic heading for those C-VLANs. The solution is to configure and apply a traffic control profile for all remaining traffic on the interface.

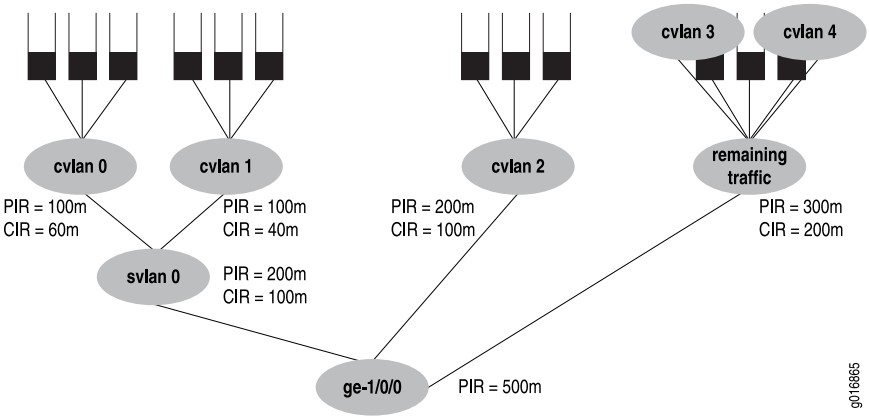
Figure 12: Example 1 Handling Remaining Traffic with no Explicit Traffic Control Profile



Example 1 considers the case where C-VLANs 3 and 4 have no explicit traffic control profile, yet need to establish a shaping and guaranteed transmit rate for traffic heading for those C-VLANs. The solution is to add a traffic control profile to the **svlan1** interface set. This example builds on the example used in "Example: Configuring a Four-Level Scheduler Hierarchy" on page 181 and does not repeat all configuration details, only those at the S-VLAN level.

Next, consider Example 2 shown in Figure 13 on page 194.

Figure 13: Example 2 Handling Remaining Traffic with an Interface Set



Copyright © 2017, Juniper Networks, Inc.

In Example 2, **ge-1/0/0** has five logical interfaces (C-VLAN 0, 1, 2, 3 and 4), and S-VLAN 0, which are covered by the interface set:

- Scheduling for the interface set **svlan0** is specified by referencing an **output-traffic-control-profile** statement, which specifies the **guaranteed-rate**, **shaping-rate**, and **delay-buffer-rate** statement values for the interface set. In this example, the output traffic control profile called **tcp-svlan0** guarantees 100 Mbps and shapes the interface set **svlan0** to 200 Mbps.
- Scheduling and queuing for remaining traffic of **svlan0** is specified by referencing an **output-traffic-control-profile-remaining** statement, which references a **scheduler-map** statement that establishes queues for the remaining traffic. The specified traffic control profile can also configure guaranteed, shaping, and delay-buffer rates for the remaining traffic. In Example 2, **output-traffic-control-profile-remaining tcp-svlan0-rem** references **scheduler-map smap-svlan0-rem**, which calls for a best-effort queue for remaining traffic (that is, traffic on unit 3 and unit 4, which is not classified by the **svlan0** interface set). The example also specifies a **guaranteed-rate** of 200 Mbps and a **shaping-rate** of 300 Mbps for all remaining traffic.
- Scheduling and queuing for logical interface **ge-1/0/0 unit 1** is configured "traditionally" and uses an **output-traffic-control-profile** specified for that unit. In this example, **output-traffic-control-profile tcp-ifl1** specifies scheduling and queuing for **ge-1/0/0 unit 1**.

Configuration

This section contains the following topics:

- Controlling Remaining Traffic With No Explicit Traffic Control Profile on page 195
- Controlling Remaining Traffic With An Interface Set on page 196

Controlling Remaining Traffic With No Explicit Traffic Control Profile

CLI Quick Configuration

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

```
set class-of-service interfaces interface-set svlan0 output-traffic-control-profile
    tcp-svlan0
set class-of-service interfaces interface-set svlan1 output-traffic-control-profile tcp-svlan1
set class-of-service interfaces interface-set svlan1 output-traffic-control-profile-remaining
    tcp-svlan1-remaining
set class-of-service traffic-control-profiles tcp-svlan1 shaping-rate 400m guaranteed-rate
    300m
set class-of-service traffic-control-profiles tcp-svlan1-remaining shaping-rate 300m
    guaranteed-rate 200m scheduler-map smap-remainder
```

Step-by-Step Procedure

To control remaining traffic with no explicit traffic control profile:

1. Set the logical interfaces for the S-VLANs.

Copyright © 2017, Juniper Networks, Inc.

```
[edit class-of-service interfaces]
user@host# set interface-set svlan0 output-traffic-control-profile tcp-svlan0
user@host# set interface-set svlan1 output-traffic-control-profile tcp-svlan1
user@host# set interface-set svlan1 output-traffic-control-profile-remaining
  tcp-svlan1-remaining
```

2. Set the shaping and guaranteed transmit rates for traffic heading for those C-VLANs.

```
[edit class-of-service traffic-control-profiles]
user@host# set tcp-svlan1 shaping-rate 400m guaranteed-rate 300m
user@host# set tcp-svlan1-remaining shaping-rate 300m guaranteed-rate 200m
  scheduler-map smap-remainder
```

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

```
[edit]
user@host# show class-of-service interfaces
interface-set svlan0 {
  output-traffic-control-profile tcp-svlan0;
}
interface-set svlan1 {
  output-traffic-control-profile tcp-svlan1;
  output-traffic-control-profile-remaining tcp-svlan1-remaining; # For all remaining traffic
}
```

```
[edit]
user@host# show class-of-service traffic-control-profiles
tcp-svlan1 {
  shaping-rate 400m;
  guaranteed-rate 300m;
}
tcp-svlan1-remaining {
  shaping-rate 300m;
  guaranteed-rate 200m;
  scheduler-map smap-remainder; # this smap is not shown in detail
}
```

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

Controlling Remaining Traffic With An Interface Set

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

```
set class-of-service interfaces interface-set svlan0 output-traffic-control-profile
  tcp-svlan0
```


Copyright © 2017, Juniper Networks, Inc.

```
set class-of-service interfaces ge-1/0/0 output-traffic-control-profile-remaining
  tcp-svlan0-rem unit 1output-traffic-control-profile tcp-ifl1
set class-of-service traffic-control-profiles tcp-svlan0 shaping-rate 200m guaranteed-rate
  100m
set class-of-service traffic-control-profiles tcp-svlan0-rem shaping-rate 300m
  guaranteed-rate 200m scheduler-map smap-svlan0-rem
set class-of-service traffic-control-profiles tcp-ifl1 scheduler-map smap-ifl1
set class-of-service scheduler-maps smap-svlan0-rem forwarding-class best-effort
  scheduler-sched-foo
set class-of-service scheduler-maps smap-ifl1 forwarding-class best-effort
  scheduler-sched-bar
set class-of-service scheduler-maps smap-ifl1 forwarding-class assured-forwarding
  scheduler-sched-bar
```

Step-by-Step Procedure

To control remaining traffic with an interface set:

1. Set the interface set for the S-VLAN.

```
[edit class-of-service interfaces]
user@host# set interface-set svlan0 output-traffic-control-profile tcp-svlan0
user@host# set ge-1/0/0 output-traffic-control-profile-remaining tcp-svlan0-rem
  unit 1output-traffic-control-profile tcp-ifl1
```

2. Set the traffic control profiles.

```
[edit class-of-service traffic-control-profiles]
user@host# set tcp-svlan0 shaping-rate 200m guaranteed-rate 100m
user@host# set tcp-svlan0-rem shaping-rate 300m guaranteed-rate 200m
  scheduler-map smap-svlan0-rem
user@host# set tcp-ifl1 scheduler-map smap-ifl1
```

3. Set the scheduler map.

```
[edit class-of-service scheduler-maps]
user@host# set smap-svlan0-rem forwarding-class best-effort scheduler-sched-foo
user@host# set smap-ifl1 forwarding-class best-effort scheduler-sched-bar
user@host# set smap-ifl1 forwarding-class assured-forwarding scheduler-sched-bar
```

Results

From configuration mode, confirm your configuration by entering the **show class-of-service interfaces**, **show class-of-service traffic-control-profiles** , and **show class-of-service scheduler-maps** commands. If the output does not display the intended configuration, repeat the configuration instructions in this example to correct it. Example 2 does not include the **[edit interfaces]** configuration.

```
[edit]
user@host# show class-of-service interfaces
interface-set {
  svlan0 {
    output-traffic-control-profile tcp-svlan0; # Guarantee & shaper for svlan0
  }
}
ge-1/0/0 {
  output-traffic-control-profile-remaining tcp-svlan0-rem
```

Copyright © 2017, Juniper Networks, Inc.

```
        # Unit 3 and 4 are not explicitly configured, but captured by “remaining'
        unit 1 {
        output-traffic-control-profile tcp-ifl1; # Unit 1 be & ef queues
      }
}

[edit]
user@host# show class-of-service traffic-control-profiles
tcp-svlan0 {
   shaping-rate 200m;
   guaranteed-rate 100m;
}
tcp-svlan0-rem {
   shaping-rate 300m;
   guaranteed-rate 200m;
   scheduler-map smap-svlan0-rem; # This specifies queues for remaining traffic
}
tcp-ifl1 {
   scheduler-map smap-ifl1;
}

[edit]
user@host# show class-of-service scheduler-maps
smap-svlan0-rem {
   forwarding-class best-effort scheduler sched-foo;
}
smap-ifl1 {
   forwarding-class best-effort scheduler sched-bar;
   forwarding-class assured-forwarding scheduler sched-baz;
}
```

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

The configuration for the referenced schedulers is not given for this example.

Verification

Verifying Remaining Traffic Control

Purpose Verify that the remaining traffic is controlled properly.

Action From operational mode, enter the following commands:

- **show class-of-service interfaces**
- **show class-of-service traffic-control-profiles**
- **show class-of-service scheduler-maps**

Related Documentation

- Understanding Hierarchical Schedulers on page 175

Copyright © 2017, Juniper Networks, Inc.

PART 4

Configuring Class of Service for IPv6

- Configuring Class of Service for IPv6 Traffic on page 201

Copyright © 2017, Juniper Networks, Inc.

Copyright © 2017, Juniper Networks, Inc.

CHAPTER 14

Configuring Class of Service for IPv6 Traffic

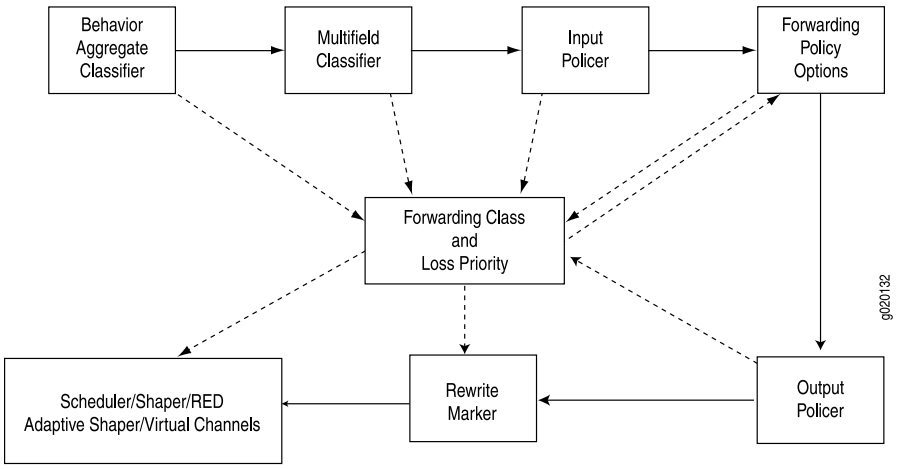
- CoS Functions for IPv6 Traffic Overview on page 201
- Understanding CoS with DSCP IPv6 BA Classifier on page 203
- Example: Configuring CoS with DSCP IPv6 BA Classifiers on page 205
- Understanding DSCP IPv6 Rewrite Rules on page 208
- Example: Configuring CoS with DSCP IPv6 Rewrite Rules on page 209

CoS Functions for IPv6 Traffic Overview

Class-of-service (CoS) processing for IPv6 traffic uses the IPv6 DiffServ code point (DSCP) value. The IPv6 DSCP value is the first six bits in the 8-bit Traffic Class field of the IPv6 header. The DSCP value is used to determine the behavior aggregate (BA) classification for the packet entering the network device. You use classifier rules to map the DSCP code points to a forwarding class and packet loss priority. You use rewrite rules to map the forwarding class and packet loss priority back to DSCP values on packets exiting the device.

Figure 14 on page 201 shows the components of the CoS features for Juniper Networks devices, illustrating the sequence in which they interact.

Figure 14: Packet Flow Through an SRX Series Device



Copyright © 2017, Juniper Networks, Inc.

NOTE: Not all CoS features are supported on all devices.

- CoS components perform the following operations:

  BA classifier rules map DSCP code points to a forwarding class and loss priority. The forwarding class and loss priority determine the per-hop behavior of the packet throughout the system. The forwarding class associates a packet with an outbound transmission queue. Loss priority affects the scheduling of a packet without affecting the relative ordering of packets. BA classification is a simple way that "downstream" nodes can honor the CoS objectives that were encoded "upstream."

  See "Example: Configuring CoS with DSCP IPv6 BA Classifiers" on page 205.

- Multifield classifier rules overwrite the initial forwarding class and loss priority determination read by the BA classifier rule. You typically use multifield classifier rules on nodes close to the content origin, where a packet might not have been encoded with the desired DSCP values in the headers. A multifield classifier rule assigns packets to a forwarding class and assigns a packet loss priority based on filters, such as source IP, destination IP, port, or application.

  See "Example: Configuring and Applying a Firewall Filter for a Multifield Classifier" on page 49.

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

- Scheduler maps are applied to interfaces and associate the outgoing packets with a scheduler and a forwarding class.

  The scheduler manages the output transmission queue, including:

  - Buffer size—Defines the period for which a packet is stored during congestion.
  - Scheduling priority and transmit rate—Determines the order in which a packet is transmitted.
  - Drop profile—Defines how aggressively to drop a packet that is using a particular scheduler.

  See "Example: Configuring Class-of-Service Schedulers" on page 94.

- Output policers meter traffic and might change the forwarding class and loss priority of a packet if a traffic flow exceeds its service level.

- Rewrite rules map forwarding class and packet loss priority to DSCP values. You typically use rewrite rules in conjunction with multifield classifier rules close to the content origin, or when the device is at the border of a network and must alter the code points to meet the policies of the targeted peer.

  See "Example: Configuring CoS with DSCP IPv6 Rewrite Rules" on page 209.

 Copyright © 2017, Juniper Networks, Inc.

Only BA classification rules and rewrite rules require special consideration to support CoS for IPv6 traffic. The program logic for the other CoS features is not sensitive to differences between IPv4 and IPv6 traffic.

Related Documentation

- Understanding CoS with DSCP IPv6 BA Classifier on page 203
- Example: Configuring CoS with DSCP IPv6 BA Classifiers on page 205
- Understanding DSCP IPv6 Rewrite Rules on page 208
- Example: Configuring CoS with DSCP IPv6 Rewrite Rules on page 209

Understanding CoS with DSCP IPv6 BA Classifier

A behavior aggregate (BA) classifier rule maps DSCP code points to a forwarding class and loss priority. The forwarding class and loss priority determine the per-hop behavior of the packet throughout the system. The forwarding class associates a packet with an outbound transmission queue. Loss priority affects the scheduling of a packet without affecting the relative ordering of packets.

BA classification can be applied within one DiffServ domain or between two domains, where each domain honors the CoS results generated by the other domain. Table 38 on page 203 shows the mapping for the default DSCP IPv6 BA classifier.

Table 38: Default IPv6 BA Classifier Mapping

Code Points	DSCP IPv6 Alias	Forwarding Class	Packet Loss Priority
101110	ef	expedited-forwarding	low
001010	af11	assured-forwarding	low
001100	af12	assured-forwarding	high
001110	af13	assured-forwarding	high
010010	af21	best-effort	low
010100	af22	best-effort	low
010110	af23	best-effort	low
011010	af31	best-effort	low
011100	af32	best-effort	low
011110	af33	best-effort	low
100010	af41	best-effort	low
100100	af42	best-effort	low

Copyright © 2017, Juniper Networks, Inc.

Table 38: Default IPv6 BA Classifier Mapping *(continued)*

Code Points	DSCP IPv6 Alias	Forwarding Class	Packet Loss Priority
100110	af43	best-effort	low
000000	be	best-effort	low
001000	cs1	best-effort	low
010000	cs2	best-effort	low
011000	cs3	best-effort	low
100000	cs4	best-effort	low
101000	cs5	best-effort	low
110000	nc1/cs6	network-control	low
111000	nc2/cs7	network-control	low

You can use the CLI **show** command to display the settings for the CoS classifiers. The following command shows the settings for the default DSCP IPv6 classifier:

```
user@host# show class-of-service classifier type dscp-ipv6
Classifier: dscp-ipv6-default, Code point type: dscp-ipv6, Index: 8
  Code point         Forwarding class                   Loss priority
  000000             best-effort                        low
  000001             best-effort                        low
  000010             best-effort                        low
  000011             best-effort                        low
  000100             best-effort                        low
  000101             best-effort                        low
  011011             best-effort                        low
...
Classifier: dscp-ipv6-compatibility, Code point type: dscp-ipv6, Index: 9
  Code point         Forwarding class                   Loss priority
  000000             best-effort                        low
  000001             best-effort                        low
  000010             best-effort                        low
  000011             best-effort                        low
  000100             best-effort                        low
  000101             best-effort                        low
  000110             best-effort                        low
  000111             best-effort                        low
 ...
```


Copyright © 2017, Juniper Networks, Inc.

NOTE: The predefined classifier named dscp-ipv6-compatibility maps all code point loss priorities to low. It maps 110000 and 111000 (typically seen in network control packets) to the network-control class and all other code points to the best-effort class. The dscp-ipv6-compatibility classifier is an implicit classifier similar to ipprec-compatibility, which is provided to map IP precedence bits in IPv4 traffic when no classifier has been configured.

Related Documentation

- Example: Configuring CoS with DSCP IPv6 BA Classifiers on page 205
- CoS Functions for IPv6 Traffic Overview on page 201
- Understanding DSCP IPv6 Rewrite Rules on page 208
- Example: Configuring CoS with DSCP IPv6 Rewrite Rules on page 209

Example: Configuring CoS with DSCP IPv6 BA Classifiers

This example shows how to associate an interface with a default or user-defined DSCP IPv6 BA classifier.

- Requirements on page 205
- Overview on page 205
- Configuration on page 205
- Verification on page 208

Requirements

Before you begin, configure the ge-0/0/0 interface on the device for IPv6 and define your user-defined DSCP IPv6 classifier settings. See "Understanding CoS with DSCP IPv6 BA Classifier" on page 203.

Overview

In this example, you configure CoS and define forwarding classes. You create the behavior aggregate classifier for DiffServ CoS as dscp-ipv6-example and import the default DSCP IPv6 classifier.

You then specify the best-effort forwarding class as be-class, the expedited forwarding class as ef-class, the assured forwarding class as af-class, and the network control forwarding class as nc-class. Finally, you apply your user-defined classifier to interface ge-0/0/0.

Configuration

CLI Quick Configuration

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

Copyright © 2017, Juniper Networks, Inc.

```
set class-of-service forwarding-classes queue 0 be-class
set class-of-service forwarding-classes queue 1 ef-class
set class-of-service forwarding-classes queue 2 af-class
set class-of-service forwarding-classes queue 3 nc-class
set class-of-service classifiers dscp-ipv6 dscp-ipv6-example import default
set class-of-service classifiers dscp-ipv6 dscp-ipv6-example forwarding-class be-class
    loss-priority high code-points 000001
set class-of-service classifiers dscp-ipv6 dscp-ipv6-example forwarding-class ef-class
    loss-priority high code-points 101111
set class-of-service classifiers dscp-ipv6 dscp-ipv6-example forwarding-class af-class
    loss-priority high code-points 001100
set class-of-service classifiers dscp-ipv6 dscp-ipv6-example forwarding-class nc-class
    loss-priority high code-points 110001
set class-of-service interfaces ge-0/0/0 unit 0 classifiers dscp-ipv6 dscp-ipv6-example
```

Step-by-Step Procedure

The following example requires you to navigate various levels in the configuration hierarchy. For instructions on how to do that, see *Using the CLI Editor in Configuration Mode* in the *CLI User Guide*.

To configure CoS with a user-defined DSCP IPv6 BA classifier:

1. Configure CoS.

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

2. Define forwarding classes.

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

3. Create a behavior aggregate classifier for DiffServ CoS.

   ```
   [edit class-of-service]
   user@host# edit classifiers dscp-ipv6 dscp-ipv6-example
   ```

4. Import a DSCP IPv6 classifier.

   ```
   [edit class-of-service classifiers dscp-ipv6 dscp-ipv6-example]
   user@host# set import default
   ```

5. Specify a best-effort forwarding class classifier.

   ```
   [edit class-of-service classifiers dscp-ipv6 dscp-ipv6-example]
   user@host# set forwarding-class be-class loss-priority high code-points 000001
   ```

6. Specify an expedited forwarding class classifier.

   ```
   [edit class-of-service classifiers dscp-ipv6 dscp-ipv6-example]
   user@host# set forwarding-class ef-class loss-priority high code-points 101111
   ```


Copyright © 2017, Juniper Networks, Inc.

7. Specify an assured forwarding class classifier.

```
[edit class-of-service classifiers dscp-ipv6 dscp-ipv6-example]
user@host# set forwarding-class af-class loss-priority high code-points 001100
```

8. Specify a network control forwarding class classifier.

```
[edit class-of-service classifiers dscp-ipv6 dscp-ipv6-example]
user@host# set forwarding-class nc-class loss-priority high code-points 110001
```

9. Associate a user-defined classifier with an interface.

```
[edit class-of-service]
user@host# set interfaces ge-0/0/0 unit 0 classifiers dscp-ipv6 dscp-ipv6-example
```

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

```
[edit]
user@host# show class-of-service
classifiers {
  dscp-ipv6 dscp-ipv6-example {
  import default;
  forwarding-class be-class {
    loss-priority high code-points 000001;
  }
  forwarding-class ef-class {
    loss-priority high code-points 101111;
  }
  forwarding-class af-class {
    loss-priority high code-points 001100;
  }
  forwarding-class nc-class {
    loss-priority high code-points 110001;
    }
  }
  }
  forwarding-classes {
    queue 0 be-class;
    queue 1 ef-class;
    queue 2 af-class;
    queue 3 nc-class;
  }
  interfaces {
    ge-0/0/0 {
    unit 0 {
    classifiers {
    dscp-ipv6 dscp-ipv6-example;
      }
    }
  }
}
```

Copyright © 2017, Juniper Networks, Inc.

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

Verification

Verifying the CoS with DSCP IPv6 BA Classifier Configuration

Purpose Verify that the user-defined DSCP IPv6 BA classifier is associated with an interface.

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

Related Documentation

- Understanding CoS with DSCP IPv6 BA Classifier on page 203
- CoS Functions for IPv6 Traffic Overview on page 201
- Understanding DSCP IPv6 Rewrite Rules on page 208
- Example: Configuring CoS with DSCP IPv6 Rewrite Rules on page 209

Understanding DSCP IPv6 Rewrite Rules

After Junos OS CoS processing, a rewrite rule maps the forwarding class and loss priority after Junos OS CoS processing to a corresponding DSCP value specified in the rule. Typically, you use rewrite rules to alter the CoS values in outgoing packets to meet the requirements of the targeted peer.

You can use the CLI show command to display the configuration for the CoS classifiers. The following command shows the configuration of the default DSCP IPv6 rewrite rule:

```
user@host# show class-of-service rewrite-rule type dscp-ipv6
Rewrite rule: dscp-ipv6-default, Code point type: dscp-ipv6, Index: 32
  Forwarding class                Loss priority       Code point
  best-effort                     low                 000000
  best-effort                     high                000000
  expedited-forwarding            low                 101110
  expedited-forwarding            high                101110
  assured-forwarding              low                 001010
  assured-forwarding              high                001100
  network-control                 low                 110000
  network-control                 high                111000
```

Related Documentation

- Example: Configuring CoS with DSCP IPv6 Rewrite Rules on page 209
- CoS Functions for IPv6 Traffic Overview on page 201
- Understanding CoS with DSCP IPv6 BA Classifier on page 203
- Example: Configuring CoS with DSCP IPv6 BA Classifiers on page 205

Copyright © 2017, Juniper Networks, Inc.

Example: Configuring CoS with DSCP IPv6 Rewrite Rules

This example shows how to associate an interface with a default or user-defined DSCP IPv6 rewrite rule. Typically, you use rewrite rules to alter CoS values in outgoing packets to meet the requirements of the targeted peer.

- Requirements on page 209
- Overview on page 209
- Configuration on page 209
- Verification on page 211

Requirements

Before you begin, configure the ge-0/0/0 interface on the device for IPv6 and define your user-defined DSCP IPv6 rewrite rules.

Overview

In this example, you configure CoS and create a user-defined rewrite rule called rewrite-ipv6-dscps. You then specify rewrite rules for the best-effort forwarding class as be-class, the expedited forwarding class as ef-class, the assured forwarding class as af-class, and the network control forwarding class as nc-class. Finally, you associate interface ge-0/0/0 with the user-defined rule.

Configuration

CLI Quick Configuration

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

```
set class-of-service rewrite-rules dscp-ipv6 rewrite-ipv6-dscps
set class-of-service rewrite-rules dscp-ipv6 rewrite-ipv6-dscps forwarding-class be-class
    loss-priority low code-point 000000
set class-of-service rewrite-rules dscp-ipv6 rewrite-ipv6-dscps forwarding-class be-class
    loss-priority high code-point 000001
set class-of-service rewrite-rules dscp-ipv6 rewrite-ipv6-dscps forwarding-class ef-class
    loss-priority low code-point 101110
set class-of-service rewrite-rules dscp-ipv6 rewrite-ipv6-dscps forwarding-class ef-class
    loss-priority high code-point 101111
set class-of-service rewrite-rules dscp-ipv6 rewrite-ipv6-dscps forwarding-class af-class
    loss-priority low code-point 001010
set class-of-service rewrite-rules dscp-ipv6 rewrite-ipv6-dscps forwarding-class af-class
    loss-priority high code-point 001100
set class-of-service rewrite-rules dscp-ipv6 rewrite-ipv6-dscps forwarding-class nc-class
    loss-priority low code-point 110000
set class-of-service rewrite-rules dscp-ipv6 rewrite-ipv6-dscps forwarding-class nc-class
    loss-priority high code-point 110001
set class-of-service interfaces ge-0/0/0 unit 0 rewrite-rules dscp rewrite-dscps
```

Copyright © 2017, Juniper Networks, Inc.

Step-by-Step Procedure

The following example requires you to navigate various levels in the configuration hierarchy. For instructions on how to do that, see *Using the CLI Editor in Configuration Mode* in the *CLI User Guide*.

To configure a CoS with a user-defined DSCP IPv6 rewrite rule:

1. Configure CoS.

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

2. Create a user-defined rewrite rule.

   ```
   [edit class-of-service]
   user@host# edit rewrite-rules dscp-ipv6 rewrite-ipv6-dscps
   ```

3. Specify rewrite rules for the best-effort forwarding class.

   ```
   [edit class-of-service rewrite-rules dscp-ipv6 rewrite-ipv6-dscps]
   user@host# set forwarding-class be-class loss-priority low code-point 000000
   user@host# set forwarding-class be-class loss-priority high code-point 000001
   ```

4. Specify rewrite rules for the expedited-forwarding forwarding class.

   ```
   [edit class-of-service rewrite-rules dscp-ipv6 rewrite-ipv6-dscps]
   user@host# set forwarding-class ef-class loss-priority low code-point 101110
   user@host# set forwarding-class ef-class loss-priority high code-point 101111
   ```

5. Specify rewrite rules for the assured-forwarding forwarding class.

   ```
   [edit class-of-service rewrite-rules dscp-ipv6 rewrite-ipv6-dscps]
   user@host# set forwarding-class af-class loss-priority low code-point 001010
   user@host# set forwarding-class af-class loss-priority high code-point 001100
   ```

6. Specify rewrite rules for the network-control forwarding class.

   ```
   [edit class-of-service rewrite-rules dscp-ipv6 rewrite-ipv6-dscps]
   user@host# set forwarding-class nc-class loss-priority low code-point 110000
   user@host# set forwarding-class nc-class loss-priority high code-point 110001
   ```

7. Associate an interface with a user-defined rule.

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

Results

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

```
[edit]
user@host# show class-of-service
interfaces {
   ge-0/0/0 {
```


Copyright © 2017, Juniper Networks, Inc.

```
        unit 0 {
          rewrite-rules {
            dscp rewrite-dscps;
          }
        }
      }
    }
    rewrite-rules {
      dscp-ipv6 rewrite-ipv6-dscps {
        forwarding-class be-class {
          loss-priority low code-point 000000;
          loss-priority high code-point 000001;
        }
        forwarding-class ef-class {
          loss-priority low code-point 101110;
          loss-priority high code-point 101111;
        }
        forwarding-class af-class {
          loss-priority low code-point 001010;
          loss-priority high code-point 001100;
        }
        forwarding-class nc-class {
          loss-priority low code-point 110000;
          loss-priority high code-point 110001;
        }
      }
    }
```

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

Verification

Verifying the CoS with DSCP IPv6 Rewrite Rule Configuration

Purpose Verify that the user-defined CoS with DSCP IPv6 rewrite rule is associated with an interface.

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

Related Documentation

- Understanding DSCP IPv6 Rewrite Rules on page 208
- CoS Functions for IPv6 Traffic Overview on page 201
- Understanding CoS with DSCP IPv6 BA Classifier on page 203
- Example: Configuring CoS with DSCP IPv6 BA Classifiers on page 205

Copyright © 2017, Juniper Networks, Inc.

Copyright © 2017, Juniper Networks, Inc.

PART 5

Configuring Class of Service for I/O Cards

- Configuring Class of Service for I/O Cards on page 215

Copyright © 2017, Juniper Networks, Inc.

Copyright © 2017, Juniper Networks, Inc.

CHAPTER 15

Configuring Class of Service for I/O Cards

- PIR-Only and CIR Mode Overview on page 215
- Understanding Priority Propagation on page 217
- Understanding IOC Hardware Properties on page 218
- Understanding IOC Map Queues on page 220
- WRED on the IOC Overview on page 221
- MDRR on the IOC Overview on page 225
- CoS Support on the SRX5000 Module Port Concentrator Overview on page 227
- Example: Configuring CoS on SRX5000 Devices with an MPC on page 228

PIR-Only and CIR Mode Overview

The actual behavior of many CoS parameters, especially the shaping rate and guaranteed rate, depends on whether the physical interface is operating in one of the following modes:

- PIR-only Mode on page 215
- CIR Mode on page 216

PIR-only Mode

In PIR-only (peak information rate) mode, one or more nodes perform shaping. The physical interface is in PIR-only mode if no child (or grandchild) node under the port has a guaranteed rate configured. The mode of the port is important because in PIR-only mode, the scheduling across the child nodes is in proportion to their shaping rates (PIRs) and not the guaranteed rates (CIRs). This can be important if the observed behavior is not what is anticipated.

In PIR-only mode, nodes cannot send if they are above the configured shaping rate. Table 39 on page 215 shows the mapping between the configured priority and the hardware priority for PIR-only.

Table 39: Internal Node Queue Priority for PIR-Only Mode

Configured Priority	Hardware Priority
Strict-high	0

Copyright © 2017, Juniper Networks, Inc.

Table 39: Internal Node Queue Priority for PIR-Only Mode *(continued)*

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

CIR Mode

In CIR (committed information rate) mode, one or more nodes applies a guaranteed rate and might perform shaping. A physical interface is in CIR mode if at least one child (or grandchild) node has a guaranteed rate configured. In addition, any child or grandchild node under the physical interface can have a shaping rate configured. Only the guaranteed rate matters. In CIR mode, nodes that do not have a guaranteed rate configured are assumed to have a very small guaranteed rate (queuing weight).

In CIR mode, the priority for each internal node depends on whether the highest active child node is above or below the guaranteed rate. Table 40 on page 216 shows the mapping between the highest active child's priority and the hardware priority below and above the guaranteed rate.

Table 40: Internal Node Queue Priority for CIR Mode

Configured Priority of Highest Active Child Node	Hardware Priority Below Guaranteed Rate	Hardware Priority Above Guaranteed Rate
Strict-high	0	0
High	0	3
Medium-high	1	3
Medium-low	1	3
Low	2	3

Related Documentation

- Understanding Priority Propagation on page 217
- Understanding IOC Hardware Properties on page 218
- Understanding IOC Map Queues on page 220
- WRED on the IOC Overview on page 221
- MDRR on the IOC Overview on page 225

Copyright © 2017, Juniper Networks, Inc.

Understanding Priority Propagation

SRX5600 and SRX5800 devices with input/output cards (IOCs) perform priority propagation. Priority propagation is useful for mixed traffic environments when, for example, you want to make "Understanding IOC Map Queues" on page 220 sure that the voice traffic of one customer does not suffer from the data traffic of another customer. Nodes and queues are always serviced in the order of their priority. The priority of a queue is decided by configuration (the default priority is low) in the scheduler. However, not all elements of hierarchical schedulers have direct priorities configured. Internal nodes, for example, must determine their priority in other ways.

The priority of any internal node is decided as follows:

- By the highest priority of an active child (interface sets only take the highest priority of their active children
- Whether the node is above its configured guaranteed rate (CIR) or not (this is relevant only if the physical interface is in CIR mode)

Each queue has a configured priority and a hardware priority. Table 41 on page 217 shows the usual mapping between the configured priority and the hardware priority.

Table 41: Queue Priority

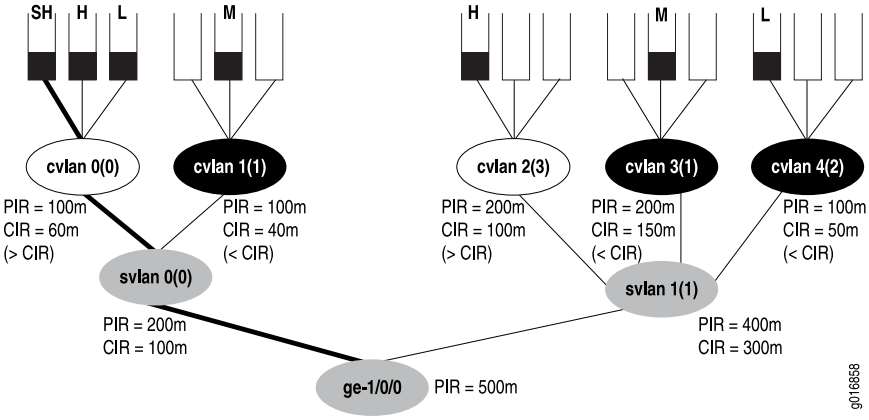
Configured Priority	Hardware Priority
Strict-high	0
High	0
Medium-high	1
Medium-low	1
Low	2

Figure 15 on page 218 shows a physical interface with hierarchical schedulers configured. The configured priorities are shown for each queue at the top of the figure. The hardware priorities for each node are shown in parentheses. Each node also shows any configured shaping rate (PIR) or guaranteed rate (CIR) and whether or not the queues are above or below the CIR. The nodes are shown in one of the following three states:

- Above the CIR (clear)
- Below the CIR (dark)
- Condition where the CIR does not matter (gray)

Copyright © 2017, Juniper Networks, Inc.

Figure 15: Hierarchical Schedulers and Priorities



In Figure 15 on page 218, the strict high queue for C-VLAN 0 (cvlan 0) receives service first, even though the C-VLAN is above the configured CIR. Once that queue has been drained, and the priority of the node has become 3 instead of 0 (because of the lack of strict-high traffic), the system moves on to the medium queues (cvlan 1 and cvlan 3), draining them in a round-robin fashion where empty queues lose their hardware priority. The low queue on cvlan 4 (priority 2) is sent next because that mode is below the CIR. Then, the high queues on cvlan 0 and cvlan2 (both now with priority 3) are drained in a round-robin fashion, and finally the low queue on cvlan 0 is drained (because svlan 0 has a priority of 3).

Related Documentation

- PIR-Only and CIR Mode Overview on page 215
- Understanding IOC Hardware Properties on page 218
- Understanding IOC Map Queues on page 220
- WRED on the IOC Overview on page 221
- MDRR on the IOC Overview on page 225

Understanding IOC Hardware Properties

On SRX5600 and SRX5800 devices, two IOCs (40x1GE IOC and 4x10GE IOC) are supported on which you can configure schedulers and queues. You can configure 15 VLAN sets per Gigabit Ethernet (40x1GE IOC) port and 255 VLAN sets per 10-Gigabit Ethernet (4x10GE IOC) port. The IOC performs priority propagation from one hierarchy level to another, and drop statistics are available on the IOC per color per queue instead of just per queue.

SRX5600 and SRX5800 devices with IOCs have Packet Forwarding Engines that can support up to 512 MB of frame memory, and packets are stored in 512-byte frames. Table 42 on page 219 compares the major properties of the Packet Forwarding Engine within the IOC.

Copyright © 2017, Juniper Networks, Inc.

Table 42: Packet Forwarding Engine Properties within 40x1GE IOC and 4x10GE IOC

Feature	PFE Within 40x1GE IOC and 4x10GE IOC
Number of usable queues	16,000
Number of shaped logical interfaces	2,000 with 8 queues each, or 4,000 with 4 queues each.
Number of hardware priorities	4
Priority propagation	Yes
Dynamic mapping	Yes: schedulers per port are not fixed.
Drop statistics	Per queue per color (PLP high, low)

Additionally, the IOC features also support hierarchical weighted random early detection (WRED).

The IOC supports the following hierarchical scheduler characteristics:

- Shaping at the physical interface level
- Shaping and scheduling at the service VLAN interface set level
- Shaping and scheduling at the customer VLAN logical interface level
- Scheduling at the queue level

The IOC supports the following features for scalability:

- 16,000 queues per PFE
- 4 PFEs per IOC
  - 4000 schedulers at logical interface level (level 3) with 4 queues each
  - 2000 schedulers at logical interface level (level 3) with 8 queues each
- 255 schedulers at the interface set level (level 2) per 1-port PFE on a 10-Gigabit Ethernet IOC (4x10GE IOC)
- 15 schedulers at the interface set level (level 2) per 10-port PFE on a 1-Gigabit Ethernet IOC (40x1GE IOC)
- About 400 milliseconds of buffer delay (this varies by packet size and if large buffers are enabled)
- 4 levels of priority (strict-high, high, medium, and low)

NOTE: **The exact option for a transmit-rate (transmit-rate rate exact) is not supported on the IOCs on SRX Series devices.**

Copyright © 2017, Juniper Networks, Inc.

Related Documentation

- PIR-Only and CIR Mode Overview on page 215
- Understanding Priority Propagation on page 217
- Understanding IOC Map Queues on page 220
- WRED on the IOC Overview on page 221
- MDRR on the IOC Overview on page 225

Understanding IOC Map Queues

The manner in which the IOC maps a queue to a scheduler depends on whether 8 queues or 4 queues are configured. By default, a scheduler at level 3 has 4 queues. Level 3 scheduler X controls queue X*4 to X*4+3, so that scheduler 100 (for example) controls queues 400 to 403. However, when 8 queues per scheduler are enabled, the odd-numbered schedulers are disabled, allowing twice the number of queues per subscriber as before. With 8 queues, level 3 scheduler X controls queue X*4 to X*4+7, so that scheduler 100 (for example) now controls queues 400 to 407.

You configure the **max-queues-per-interface** statement to set the number of queues at 4 or 8 at the FPC level of the hierarchy. Changing this statement will result in a restart of the FPC.

The IOC maps level 3 (customer VLAN) schedulers in groups to level 2 (service VLAN) schedulers. Sixteen contiguous level 3 schedulers are mapped to level 2 when 4 queues are enabled, and 8 contiguous level 3 schedulers are mapped to level 2 when 8 queues are enabled. All the schedulers in the group should use the same queue priority mapping. For example, if the queue priorities of one scheduler are high, medium, and low, all members of the group should have the same queue priority.

Groups at level 3 to level 2 can be mapped at any time. However, a group at level 3 can only be unmapped from a level 2 scheduler, and only if all the schedulers in the group are free. Once unmapped, a level 3 group can be remapped to any level 2 scheduler. There is no restriction on the number of level 3 groups that can be mapped to a particular level 2 scheduler. There can be 256 level 3 groups, but fragmentation of the scheduler space can reduce the number of schedulers available. In other words, there are scheduler allocation patterns that might fail even though there are free schedulers.

In contrast to level 3 to level 2 mapping, the IOC maps level 2 (service VLAN) schedulers in a fixed mode to level 1 (physical interface) schedulers. On 40-port Gigabit Ethernet IOCs, there are 16 level 1 schedulers, and 10 of these are used for the physical interfaces. There are 256 level 2 schedulers, or 16 per level 1 schedulers. A level 1 scheduler uses level schedulers X*16 through X*16+15. Therefore level 1 scheduler 0 uses level 2 schedulers 0 through 15, level 1 scheduler 1 uses level 2 schedulers 16 through 31, and so on. On 4-port 10-Gigabit Ethernet PICs, there is one level 1 scheduler for the physical interface, and 256 level 2 schedulers are mapped to the single level 1 scheduler.

The maximum number of level 3 (customer VLAN) schedulers that can be used is 4076 (4 queues) or 2028 (8 queues) for the 10-port Gigabit Ethernet Packet Forwarding Engine

Copyright © 2017, Juniper Networks, Inc.

and 4094 (4 queues) or 2046 (8 queues) for the 10-Gigabit Ethernet Packet Forwarding Engine.

Related Documentation

- PIR-Only and CIR Mode Overview on page 215
- Understanding Priority Propagation on page 217
- Understanding IOC Hardware Properties on page 218
- WRED on the IOC Overview on page 221
- MDRR on the IOC Overview on page 225

WRED on the IOC Overview

Shaping to drop out-of-profile traffic is done on the IOC at all levels except the queue level. However, weighed random early discard (WRED) is done at the queue level with much the same result. With WRED, the decision to drop or send the packet is made before the packet is placed in the queue.

WRED shaping on the IOC involves two levels. The probabilistic drop region establishes a minimum and a maximum queue depth. Below the minimum queue depth, the drop probability is 0 (send). Above the maximum level, the drop probability is 100 (certainty).

There are four drop profiles associated with each queue. These correspond to each of four loss priorities (low, medium-low, medium-high, and high). Sixty-four sets of four drop profiles are available (32 for ingress and 32 for egress). In addition, there are eight WRED scaling profiles in each direction.

An example of an IOC drop profile for expedited forwarding traffic is as follows:

```
[edit class-of-service drop-profiles]
drop-ef {
   fill-level 20 drop-probability 0; # Minimum Q depth
   fill-level 100 drop-probability 100; # Maximum Q depth
}
```

NOTE: You can specify only two fill levels for the IOC.

You can configure the **interpolate** statement, but only two fill levels are used. The **delay-buffer-rate** statement in the traffic control profile determines the maximum queue size. This delay buffer rate is converted to packet delay buffers, where one buffer is equal to 512 bytes. For example, at 10 Mbps, the IOC will allocate 610 delay buffers when the delay buffer rate is set to 250 milliseconds. The WRED threshold values are specified in terms of absolute buffer values.

The WRED scaling factor multiples all WRED thresholds (both minimum and maximum) by the value specified. There are eight values in all: 1, 2, 4, 8, 16, 32, 64, and 128. The WRED scaling factor is chosen to best match the user-configured drop profiles. This is done because the hardware supports only certain values of thresholds (all values must be a multiple of 16). So if the configured value of a threshold is 500 (for example), the multiple

Copyright © 2017, Juniper Networks, Inc.

of 16 is 256 and the scaling factor applied is 2, making the value 512, which allows the value of 500 to be used. If the configured value of a threshold is 1500, the multiple of 16 is 752 and the scaling factor applied is 2, making the value 1504, which allows the value of 1500 to be used.

Hierarchical RED is used to support the oversubscription of the delay buffers (WRED is configured only at the queue, physical interface, and PIC levels). Hierarchical RED works with WRED as follows:

- If any level accepts the packet (the queue depth is less than the minimum buffer levels), this level accepts the packet.
- If any level probabilistically drops the packet, then this level drops the packet.

However, these rules might lead to the accepting of packets under loaded conditions that might otherwise have been dropped. In other words, the logical interface will accept packets if the physical interface is not congested.

Because of the limits placed on shaping thresholds used in the hierarchy, there is a granularity associated with the IOCs. The shaper accuracies differ at various levels of the hierarchy:

- Level 3
- Level 2
- Level 1

Shapers at the logical interface level (level 3) are more accurate than shapers at the interface set level (level 2) or at the port level (level 1).

This section contains the following topics:

- Shapers at the Logical Interface Level (Level 3) on page 222
- Shapers at the Interface Set Level (Level 2) on page 223
- Shapers at the Port Level (Level 1) on page 224

Shapers at the Logical Interface Level (Level 3)

Because of the limits placed on shaping thresholds used in the hierarchy, there is a granularity associated with the IOCs. The shaper accuracies differ at various levels of the hierarchy, with shapers at the logical interface level (level 3) being more accurate than shapers at the interface set level (level 2) or at the port level (level 1). Table 43 on page 222 shows the accuracy of the logical interface shaper at various speeds for Ethernet ports operating at 1 Gbps.

Table 43: Shaper Accuracy of 1-Gbps Ethernet at the Logical Interface Level

Range of Logical Interface Shaper	Step Granularity
Up to 4.096 Mbps	16 Kbps

Copyright © 2017, Juniper Networks, Inc.

Table 43: Shaper Accuracy of 1-Gbps Ethernet at the Logical Interface Level *(continued)*

Range of Logical Interface Shaper	Step Granularity
4.096 to 8.192 Mbps	32 Kbps
8.192 to 16.384 Mbps	64 Kbps
16.384 to 32.768 Mbps	128 Kbps
32.768 to 65.535 Mbps	256 Kbps
65.535 to 131.072 Mbps	512 Kbps
131.072 to 262.144 Mbps	1024 Kbps
262.144 to 1 Gbps	4096 Kbps

Table 44 on page 223 shows the accuracy of the logical interface shaper at various speeds for Ethernet ports operating at 10 Gbps.

Table 44: Shaper Accuracy of 10-Gbps Ethernet at the Logical Interface Level

Range of Logical Interface Shaper	Step Granularity
Up to 10.24 Mbps	40 Kbps
10.24 to 20.48 Mbps	80 Kbps
10.48 to 40.96 Mbps	160 Kbps
40.96 to 81.92 Mbps	320 Kbps
81.92 to 163.84 Mbps	640 Kbps
163.84 to 327.68 Mbps	1280 Kbps
327.68 to 655.36 Mbps	2560 Kbps
655.36 to 2611.2 Mbps	10240 Kbps
2611.2 to 5222.4 Mbps	20480 Kbps
5222.4 to 10 Gbps	40960 Kbps

Shapers at the Interface Set Level (Level 2)

Table 45 on page 224 shows the accuracy of the interface set shaper at various speeds for Ethernet ports operating at 1 Gbps.

Copyright © 2017, Juniper Networks, Inc.

Table 45: Shaper Accuracy of 1-Gbps Ethernet at the Interface Set Level

Range of Interface Set Shaper	Step Granularity
Up to 20.48 Mbps	80 Kbps
20.48 Mbps to 81.92 Mbps	320 Kbps
81.92 Mbps to 327.68 Mbps	1.28 Mbps
327.68 Mbps to 1 Gbps	20.48 Mbps

Table 46 on page 224 shows the accuracy of the interface set shaper at various speeds for Ethernet ports operating at 10 Gbps.

Table 46: Shaper Accuracy of 10-Gbps Ethernet at the Interface Set Level

Range of Interface Set Shaper	Step Granularity
Up to 128 Mbps	500 Kbps
128 Mbps to 512 Mbps	2 Mbps
512 Mbps to 2.048 Gbps	8 Mbps
2.048 Gbps to 10 Gbps	128 Mbps

Shapers at the Port Level (Level 1)

Table 47 on page 224 shows the accuracy of the physical port shaper at various speeds for Ethernet ports operating at 1 Gbps.

Table 47: Shaper Accuracy of 1-Gbps Ethernet at the Physical Port Level

Range of Physical Port Shaper	Step Granularity
Up to 64 Mbps	250 Kbps
64 Mbps to 256 Mbps	1 Mbps
256 Mbps to 1 Gbps	4 Mbps

Table 48 on page 224 shows the accuracy of the physical port shaper at various speeds for Ethernet ports operating at 10 Gbps.

Table 48: Shaper Accuracy of 10-Gbps Ethernet at the Physical Port Level

Range of Physical Port Shaper	Step Granularity
Up to 640 Mbps	2.5 Mbps
640 Mbps to 2.56 Gbps	10 Mbps

Copyright © 2017, Juniper Networks, Inc.

Table 48: Shaper Accuracy of 10-Gbps Ethernet at the Physical Port Level *(continued)*

Range of Physical Port Shaper	Step Granularity
2.56 Gbps to 10 Gbps	40 Mbps

Related Documentation

- PIR-Only and CIR Mode Overview on page 215
- Understanding Priority Propagation on page 217
- Understanding IOC Hardware Properties on page 218
- Understanding IOC Map Queues on page 220
- MDRR on the IOC Overview on page 225

MDRR on the IOC Overview

The guaranteed rate CIR at the interface set level is implemented by using modified deficit round-robin (MDRR). The IOC hardware provides four levels of strict priority. There is no restriction on the number of queues for each priority. MDRR is used among queues of the same priority. Each queue has one priority when it is under the guaranteed rate and another priority when it is over the guaranteed rate but still under the shaping rate PIR. The IOC hardware implements the priorities with 256 service profiles. Each service profile assigns eight priorities for eight queues. One set is for logical interfaces under the guaranteed rate and another set is for logical interfaces over the guaranteed rate but under the shaping rate. Each service profile is associated with a group of 16 level 3 schedulers, so there is a unique service profile available for all 256 groups at level 3, giving 4,096 logical interfaces.

Junos OS provides three priorities for traffic under the guaranteed rate and one reserved priority for traffic over the guaranteed rate that is not configurable. Junos OS provides three priorities when there is no guaranteed rate configured on any logical interface.

Table 49 on page 225 shows the relationship between Junos OS priorities and the IOC hardware priorities below and above the guaranteed rate CIR.

Table 49: Junos Priorities Mapped to IOC Hardware Priorities

Junos OS Priority	IOC Hardware Priority Below Guaranteed Rate	IOC Hardware Priority Above Guaranteed Rate
Strict-high	High	High
High	High	Low
Medium-high	Medium-high	Low
Medium-low	Medium-high	Low
Low	Medium-low	Low

Copyright © 2017, Juniper Networks, Inc.

The Junos OS parameters are set in the scheduler map:

```
[edit class-of-service schedulers]
best-effort-scheduler {
   transmit-rate percent 30; # if no shaping rate
   buffer-size percent 30;
   priority high;
}
expedited-forwarding-scheduler {
   transmit-rate percent 40; # if no shaping rate
   buffer-size percent 40;
   priority strict-high;
}
```

NOTE: The use of both a shaping rate and a guaranteed rate at the interface set level (level 2) is not supported.

MDRR is provided at three levels of the scheduler hierarchy of the IOC with a granularity of 1 through 255. There are 64 MDRR profiles at the queue level, 16 at the interface set level, and 32 at the physical interface level.

Queue transmit rates are used for queue-level MDRR profile weight calculation. The queue MDRR weight is calculated differently based on the mode set for sharing excess bandwidth. If you configure the **equal** option for excess bandwidth, then the queue MDRR weight is calculated as:

Queue weight = (255 * Transmit-rate-percentage) / 100

If you configure the **proportional** option for excess bandwidth, which is the default, then the queue MDRR weight is calculated as:

Queue weight = Queue-transmit-rate / Queue-base-rate, where

Queue-transmit-rate = (Logical-interface-rate * Transmit-rate-percentage) / 100, and

Queue-base-rate = Excess-bandwidth-proportional-rate / 255

To configure the way that the IOC should handle excess bandwidth, configure the **excess-bandwidth-share** statement at the [edit interface-set *interface-set-name*] hierarchy level. By default, the excess bandwidth is set to **proportional** with a default value of 32.64 Mbps. In this mode, the excess bandwidth is shared in the ratio of the logical interface shaping rates. If set to **equal**, the excess bandwidth is shared equally among the logical interfaces.

The following example sets the excess bandwidth sharing to proportional at a rate of 100 Mbps with a shaping rate of 80 Mbps:

```
[edit interface-set example-interface-set]
excess-bandwidth-share proportional 100m;
output-traffic-control-profile PIR-80Mbps;
```

Shaping rates established at the logical interface level are used to calculate the MDRR weights used at the interface set level. The 16 MDRR profiles are set to initial values, and

Copyright © 2017, Juniper Networks, Inc.

the closest profile with rounded values is chosen. By default, the physical port MDRR weights are preset to the full bandwidth on the interface.

Related Documentation

- PIR-Only and CIR Mode Overview on page 215
- Understanding Priority Propagation on page 217
- Understanding IOC Hardware Properties on page 218
- Understanding IOC Map Queues on page 220
- WRED on the IOC Overview on page 221

CoS Support on the SRX5000 Module Port Concentrator Overview

The SRX5000 Module Port Concentrator (SRX5K-MPC) for the SRX5600 and SRX5800 uses the Trio chipset-based queuing model, which supports class of service (CoS) characteristics that are optimized compared to CoS characteristics supported by the standard queuing model. These CoS features enable SRX5600 and SRX5800 devices to achieve end-to-end quality of service and protect the network using various security functions.

CoS features on the SRX5600 and SRX5800 devices provide differentiated services to traffic in addition to the best-effort packet processing. The main CoS features include classification, CoS field rewriting, queuing, scheduling, and traffic shaping.

When a network experiences congestion and delay, you can use the CoS features to classify packets; assign them with different levels of packet loss priority, delay, and throughput; and mark their CoS-related fields defined in Layer 2 and Layer 3 headers.

The MPC supports the following CoS features:

- BA classifier based on IEEE 802.1p for packet classification (Layer 2 headers) for priority bits of ingress packets
- Rewrite rule based on IEEE 802.1p for priority bits of egress packets

NOTE: You can configure up to 32 IEEE 802.1p rewriters on each SRX5K-MPC on the SRX5600 and SRX5800 devices.

- Eight priority queues per port with configurable schedulers at the egress physical interface

By default, the MPC supports eight queues. You can use the following CLI statement to change that setting to four queues:

set chassis fpc *fpc-number* pic *pic-number* max-queues-per-interface *4*

Changing to four-queue mode limits that number of configurable queues to four on the MPC. This does not have any effect on the performance.

Copyright © 2017, Juniper Networks, Inc.

The CoS features on the MPC have the following limitations:

- On the MPC, the per-unit-scheduler or the hierarchical-scheduler is not supported. For egress scheduling and queuing, only the default mode is supported.
- When an SPU is too busy to process every ingress packet from the MPC, some high-priority packets, such as voice packets, might be delayed or dropped by the SRX5600 or SRX5800.

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

Related Documentation

- Example: Configuring CoS on SRX5000 Devices with an MPC on page 228

Example: Configuring CoS on SRX5000 Devices with an MPC

This example shows how to configure CoS on an SRX5000 line device with an MPC.

- Requirements on page 228
- Overview on page 229
- Configuration on page 230
- Verification on page 236

Requirements

This example uses the following hardware and software components:

- SRX5600 with an SRX5K-MPC
- Junos OS Release 12.1X46-D10 or later for SRX Series

Before you begin:

- Understand CoS. See "Understanding Class of Service" on page 3.
- Understand chassis cluster configuration. See *Example: Configuring an Active/Passive Chassis Cluster on SRX Series Services Gateways*.
- Understand chassis cluster redundant interface configuration. See *Example: Configuring Chassis Cluster Redundant Ethernet Interfaces for IPv4 and IPv6 Addresses*.

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

Copyright © 2017, Juniper Networks, Inc.

Overview

In this example, you create a BA classifier to classify traffic based on the IEEE 802.1p value of the packet and assign forwarding-class priority queue to the traffic. You then configure the scheduler map and set the priority for the traffic.

By default, the SRX5K-MPC supports eight queues. In this example, you are configuring eight queues.

You apply the BA classifier to input interface and apply the scheduler map to the output interface.

Table 50 on page 229 and Table 51 on page 229 show forwarding class details with priority, assigned queue numbers, and allocated queue buffers used in this example.

Table 50: Forwarding Class Samples

Forwarding Class	Queue Number
BE	0
SIG	1
AF	2
Bronze-class	3
Silver-class	4
Gold-class	5
Control	6
VOIP	7

Table 51: Scheduler Samples

Scheduler	For CoS Traffic Type	Assigned Priority	Allocated Portion of Queue Buffer (Transmit Rate)
s-be	0	low	15
s-sig	1	low	15
s-af	2	medium-low	20
s-bronze	3	medium-low	20
s-silver	4	medium-high	10

Copyright © 2017, Juniper Networks, Inc.

Table 51: Scheduler Samples *(continued)*

Scheduler	For CoS Traffic Type	Assigned Priority	Allocated Portion of Queue Buffer (Transmit Rate)
s-gold	5	medium-high	10
s-nc	6	high	5
s-voip	7	high	5

Configuration

CLI Quick Configuration

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

```
set class-of-service classifiers ieee-802.1 c802 forwarding-class BE loss-priority low
  code-points 000
set class-of-service classifiers ieee-802.1 c802 forwarding-class SIG loss-priority low
  code-points 001
set class-of-service classifiers ieee-802.1 c802 forwarding-class AF loss-priority low
  code-points 010
set class-of-service classifiers ieee-802.1 c802 forwarding-class Bronze-Class loss-priority
  low code-points 011
set class-of-service classifiers ieee-802.1 c802 forwarding-class Silver-Class loss-priority
  low code-points 100
set class-of-service classifiers ieee-802.1 c802 forwarding-class Gold-Class loss-priority
  low code-points 101
set class-of-service classifiers ieee-802.1 c802 forwarding-class Central loss-priority low
  code-points 110
set class-of-service classifiers ieee-802.1 c802 forwarding-class VOIP loss-priority low
  code-points 111
set class-of-service forwarding-classes class BE queue-num 0
set class-of-service forwarding-classes class SIG queue-num 1
set class-of-service forwarding-classes class AF queue-num 2
set class-of-service forwarding-classes class Bronze-Class queue-num 3
set class-of-service forwarding-classes class Silver-Class queue-num 4
set class-of-service forwarding-classes class Gold-Class queue-num 5
set class-of-service forwarding-classes class Control queue-num 6
set class-of-service forwarding-classes class VOIP queue-num 7
set class-of-service scheduler-maps test forwarding-class BE scheduler s-be
set class-of-service scheduler-maps test forwarding-class SIG scheduler s-sig
set class-of-service scheduler-maps test forwarding-class AF scheduler s-af
set class-of-service scheduler-maps test forwarding-class Bronze-Class scheduler
  s-bronze
set class-of-service scheduler-maps test forwarding-class Silver-Class scheduler s-silver
set class-of-service scheduler-maps test forwarding-class Gold-Class scheduler s-gold
set class-of-service scheduler-maps test forwarding-class Control scheduler s-nc
set class-of-service scheduler-maps test forwarding-class VOIP scheduler s-voip
set class-of-service rewrite-rules ieee-802.1 rw802 forwarding-class BE loss-priority low
  code-point 000
```


Copyright © 2017, Juniper Networks, Inc.

```
set class-of-service rewrite-rules ieee-802.1 rw802 forwarding-class SIG loss-priority
  low code-point 001
set class-of-service rewrite-rules ieee-802.1 rw802 forwarding-class AF loss-priority low
  code-point 010
set class-of-service rewrite-rules ieee-802.1 rw802 forwarding-class Bronze-Class
  loss-priority low code-point 011
set class-of-service rewrite-rules ieee-802.1 rw802 forwarding-class Silver-Class
  loss-priority low code-point 100
set class-of-service rewrite-rules ieee-802.1 rw802 forwarding-class Gold-Class
  loss-priority low code-point 101
set class-of-service rewrite-rules ieee-802.1 rw802 forwarding-class Control loss-priority
  low code-point 110
set class-of-service rewrite-rules ieee-802.1 rw802 forwarding-class VOIP loss-priority
  low code-point 111
set class-of-service schedulers s-be transmit-rate percent 15
set class-of-service schedulers s-be priority low
set class-of-service schedulers s-sig transmit-rate percent 15
set class-of-service schedulers s-sig priority low
set class-of-service schedulers s-af transmit-rate percent 20
set class-of-service schedulers s-af priority medium-low
set class-of-service schedulers s-bronze transmit-rate percent 20
set class-of-service schedulers s-bronze priority medium-low
set class-of-service schedulers s-silver transmit-rate percent 10
set class-of-service schedulers s-silver priority medium-high
set class-of-service schedulers s-gold transmit-rate percent 10
set class-of-service schedulers s-gold priority medium-high
set class-of-service schedulers s-nc transmit-rate percent 5
set class-of-service schedulers s-nc priority high
set class-of-service schedulers s-voip transmit-rate percent 5
set class-of-service schedulers s-voip priority high
set class-of-service interfaces reth0 unit 0 classifiers ieee-802.1 c802
set class-of-service interfaces reth0 unit 0 rewrite-rules ieee-802.1 rw802
set class-of-service interfaces reth0 shaping-rate 1g
```

Step-by-Step Procedure

The following example requires you to navigate various levels in the configuration hierarchy. For instructions on how to do that, see *Using the CLI Editor in Configuration Mode* in the *CLI User Guide*.

To configure forwarding classes:

1. Configure a classifier.

```
[edit class-of-service]
user@host# set classifiers ieee-802.1 c802 forwarding-class BE loss-priority low
  code-points 000
user@host# set classifiers ieee-802.1 c802 forwarding-class SIG loss-priority low
  code-points 001
user@host# set classifiers ieee-802.1 c802 forwarding-class AF loss-priority low
  code-points 010
user@host# set classifiers ieee-802.1 c802 forwarding-class Bronze-Class
  loss-priority low code-points 011
user@host# set classifiers ieee-802.1 c802 forwarding-class Silver-Class loss-priority
  low code-points 100
user@host# set classifiers ieee-802.1 c802 forwarding-class Gold-Class loss-priority
  low code-points 101
```

Copyright © 2017, Juniper Networks, Inc.

```
user@host# set classifiers ieee-802.1 c802 forwarding-class Central loss-priority
  low code-points 110
user@host# set classifiers ieee-802.1 c802 forwarding-class VOIP loss-priority low
  code-points 111
```

2. Assign best-effort traffic to queue.

```
[edit class-of-service forwarding-classes class]
user@host# BE queue-num 0
user@host# SIG queue-num 1
user@host# AF queue-num 2
user@host# Bronze-Class queue-num 3
user@host# Silver-Class queue-num 4
user@host# Gold-Class queue-num 5
user@host# Control queue-num 6
user@host# VOIP queue-num 7
```

3. Define mapping of forwarding classes to packet schedulers.

```
[edit class-of-service]
user@host# set scheduler-maps test forwarding-class BE scheduler s-be
user@host# set scheduler-maps test forwarding-class SIG scheduler s-sig
user@host# set scheduler-maps test forwarding-class AF scheduler s-af
user@host# set scheduler-maps test forwarding-class Bronze-Class scheduler
  s-bronze
user@host# set scheduler-maps test forwarding-class Silver-Class scheduler s-silver
user@host# set scheduler-maps test forwarding-class Gold-Class scheduler s-gold
user@host# set scheduler-maps test forwarding-class Control scheduler s-nc
user@host# set scheduler-maps test forwarding-class VOIP scheduler s-voip
```

4. Configure the CoS rewrite rules to map the forwarding class to the desired value for the 802.1p field.

```
[edit class-of-service]
user@host# set rewrite-rules ieee-802.1 rw802 forwarding-class BE loss-priority
  low code-point 000
user@host# set rewrite-rules ieee-802.1 rw802 forwarding-class SIG loss-priority
  low code-point 001
user@host# set rewrite-rules ieee-802.1 rw802 forwarding-class AF loss-priority
  low code-point 010
user@host# set rewrite-rules ieee-802.1 rw802 forwarding-class Bronze-Class
  loss-priority low code-point 011
user@host# set rewrite-rules ieee-802.1 rw802 forwarding-class Silver-Class
  loss-priority low code-point 100
user@host# set rewrite-rules ieee-802.1 rw802 forwarding-class Gold-Class
  loss-priority low code-point 101
user@host# set rewrite-rules ieee-802.1 rw802 forwarding-class Control loss-priority
  low code-point 110
user@host# set rewrite-rules ieee-802.1 rw802 forwarding-class VOIP loss-priority
  low code-point 111
```

5. Configure eight packet schedulers with scheduling priority and transmission rates.

```
[edit class-of-service]
user@host# set schedulers s-be transmit-rate percent 15
```


Copyright © 2017, Juniper Networks, Inc.

```
user@host# set schedulers s-be priority low
user@host# set schedulers s-sig transmit-rate percent 15
user@host# set schedulers s-sig priority low
user@host# set schedulers s-af transmit-rate percent 20
user@host# set schedulers s-af priority medium-low
user@host# set schedulers s-bronze transmit-rate percent 20
user@host# set schedulers s-bronze priority medium-low
user@host# set schedulers s-silver transmit-rate percent 10
user@host# set schedulers s-silver priority medium-high
user@host# set schedulers s-gold transmit-rate percent 10
user@host# set schedulers s-gold priority medium-high
user@host# set schedulers s-nc transmit-rate percent 5
user@host# set schedulers s-nc priority high
user@host# set schedulers s-voip transmit-rate percent 5
user@host# set schedulers s-voip priority high
```

6. Apply the classifier and rewrite rules to interfaces.

```
[edit class-of-service]
user@host# set interfaces reth0 unit 0 classifiers ieee-802.1 c802
user@host# set interfaces reth1 unit 0 rewrite-rules ieee-802.1 rw802
```

7. Apply the shaping rates to control the maximum rate of traffic transmitted on an interface.

```
[edit class-of-service]
user@host# set interfaces reth0 shaping-rate 1g
```

Results From configuration mode, confirm your configuration by entering the **show xxx** command. If the output does not display the intended configuration, repeat the configuration instructions in this example to correct it.

```
classifiers {
  ieee-802.1 c802 {
    forwarding-class BE {
      loss-priority low code-points 000;
    }
    forwarding-class SIG {
      loss-priority low code-points 001;
    }
    forwarding-class AF {
      loss-priority low code-points 010;
    }
    forwarding-class Bronze-Class {
      loss-priority low code-points 011;
    }
    forwarding-class Silver-Class {
      loss-priority low code-points 100;
    }
    forwarding-class Gold-Class {
      loss-priority low code-points 101;
    }
    forwarding-class Control {
```

Copyright © 2017, Juniper Networks, Inc.

```
            loss-priority low code-points 110;
        }
        forwarding-class VOIP {
            loss-priority low code-points 111;
        }
    }
}
forwarding-classes {
    class BE queue-num 0;
    class SIG queue-num 1;
    class VOIP queue-num 7;
    class AF queue-num 2;
    class Bronze-Class queue-num 3;
    class Silver-Class queue-num 4;
    class Gold-Class queue-num 5;
    class Control queue-num 6;
}
interfaces {
    e1-0/0/0 {
        shaping-rate 1g;
        unit 0 {
            scheduler-map test;
        }
    }
    reth0 {
        shaping-rate 1g;
        unit 0 {
            classifiers {
                ieee-802.1 c802;
            }
            rewrite-rules {
                ieee-802.1 rw802;
            }
        }
    }
}
rewrite-rules {
    ieee-802.1 rw802 {
        forwarding-class BE {
            loss-priority low code-point 000;
        }
        forwarding-class SIG {
            loss-priority low code-point 001;
        }
        forwarding-class AF {
            loss-priority low code-point 010;
        }
        forwarding-class Bronze-Class {
            loss-priority low code-point 011;
        }
        forwarding-class Silver-Class {
            loss-priority low code-point 100;
        }
        forwarding-class Gold-Class {
            loss-priority low code-point 101;
        }
```


Copyright © 2017, Juniper Networks, Inc.

```
      forwarding-class Control {
        loss-priority low code-point 110;
      }
      forwarding-class VOIP {
        loss-priority low code-point 111;
      }
    }
  }
  scheduler-maps {
    test {
      forwarding-class BE scheduler s-be;
      forwarding-class VOIP scheduler s-voip;
      forwarding-class Gold-Class scheduler s-gold;
      forwarding-class SIG scheduler s-sig;
      forwarding-class AF scheduler s-af;
      forwarding-class Bronze-Class scheduler s-bronze;
      forwarding-class Silver-Class scheduler s-silver;
      forwarding-class Control scheduler s-nc;
    }
  }
  schedulers {
    s-be {
      transmit-rate percent 15;
      priority low;
    }
    s-nc {
      transmit-rate percent 5;
      priority high;
    }
    s-gold {
      transmit-rate percent 10;
      priority medium-high;
    }
    s-sig {
      transmit-rate percent 15;
      priority low;
    }
    s-af {
      transmit-rate percent 20;
      priority medium-low;
    }
    s-bronze {
      transmit-rate percent 20;
      priority medium-low;
    }
    s-silver {
      transmit-rate percent 10;
      priority medium-high;
    }
    s-voip {
      transmit-rate percent 5;
      priority high;
    }
  }
```

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

Copyright © 2017, Juniper Networks, Inc.

Verification

Confirm that the configuration is working properly.

- Verifying Class-of-Service Configuration on page 236
- Verifying the Number of Dedicated Queues Configured on MPC Interfaces on page 236

Verifying Class-of-Service Configuration

Purpose Verify that CoS is configured.

Action From operational mode, enter the **show class-of-service classifier** command.

```
user@host> show class-of-service classifier type ieee-802.1

Forwarding class          ID    Queue Restricted queue  Fabric priority  Policing
 priority   SPU priority
  BE                       0       0          0             low
normal            low
  SIG                      1       1          1             low
normal            low
  AF                       2       2          2             low
normal            low
  Bronze-Class             3       3          3             low
normal            low
  Silver-Class             4       4          0             low
normal            low
  Gold-Class               5       5          1             low
normal            low
  Control                  6       6          2             low
normal            low
  VOIP                     7       7          3             low
normal            low
```

Verifying the Number of Dedicated Queues Configured on MPC Interfaces

Purpose Display the number of dedicated queue resources that are configured for the interfaces on a port.

Action From operational mode, enter the **show class-of-service interface** command.

```
user@host> show class-of-service interface reth0

Physical interface: reth0, Index: 129
Queues supported: 8, Queues in use: 4
  Scheduler map: <default>, Index: 2
  Congestion-notification: Disabled

  Logical interface: reth0.0, Index: 71
    Object                 Name                    Type                    Index

    Classifier             dscp-ipv6-compatibility dscp-ipv6                   9
```


Copyright © 2017, Juniper Networks, Inc.

```
  Classifier              ipprec-compatibility   ip                    13

Logical interface: reth1.32767, Index: 70
```

Related Documentation

- Understanding IOC Hardware Properties on page 218
- CoS Support on the SRX5000 Module Port Concentrator Overview on page 227

Copyright © 2017, Juniper Networks, Inc.

Copyright © 2017, Juniper Networks, Inc.

PART 6

Configuration Statements and Operational Commands

- Configuration Statements on page 241
- Operational Commands on page 279

Copyright © 2017, Juniper Networks, Inc.

Copyright © 2017, Juniper Networks, Inc.

CHAPTER 16

Configuration Statements

- adaptive-shaper on page 242
- adaptive-shapers on page 243
- application-traffic-control on page 244
- buffer-size (Schedulers) on page 245
- classifiers (CoS) on page 246
- code-points (CoS) on page 247
- copy-outer-dscp on page 247
- default (CoS) on page 248
- dscp-code-point (CoS Host Outbound Traffic) on page 249
- egress-shaping-overhead on page 250
- forwarding-class (CoS Host Outbound Traffic) on page 252
- forwarding-classes (CoS) on page 253
- frame-relay-de (CoS Interfaces) on page 254
- frame-relay-de (CoS Loss Priority) on page 255
- frame-relay-de (CoS Rewrite Rule) on page 256
- host-outbound-traffic (Class-of-Service) on page 257
- ingress-policer-overhead on page 258
- interfaces (CoS) on page 260
- logical-interface-policer on page 261
- loss-priority (CoS Loss Priority) on page 262
- loss-priority (CoS Rewrite Rules) on page 263
- loss-priority-maps (CoS Interfaces) on page 264
- loss-priority-maps (CoS) on page 264
- non-strict-priority-scheduling on page 265
- rate-limiters on page 266
- rewrite-rules (CoS) on page 267
- rewrite-rules (CoS Interfaces) on page 268
- rule-sets (CoS AppQoS) on page 269

Copyright © 2017, Juniper Networks, Inc.

- scheduler-map (CoS Virtual Channels) on page 271
- shaping-rate (CoS Adaptive Shapers) on page 272
- shaping-rate (CoS Interfaces) on page 273
- shaping-rate (CoS Virtual Channels) on page 274
- trigger (CoS) on page 275
- tunnel-queuing on page 275
- virtual-channels on page 276
- virtual-channel-group (CoS Interfaces) on page 277
- virtual-channel-groups on page 278

adaptive-shaper

Syntax adaptive-shaper *adaptive-shaper-name*;

Hierarchy Level [edit class-of-service interfaces *interface-name* unit *logical-unit-number*]

Release Information Statement introduced in Release 8.5 of Junos OS.

Description Assign an adaptive shaper to an interface.

Adaptive shapers enable bandwidth limits on Frame Relay interfaces when the device receives frames containing the backward explicit congestion notification (BECN) bit.

Options ***adaptive-shaper-name***—Name of the adaptive shaper.

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

Related Documentation

- adaptive-shapers on page 243
- *Junos OS Class of Service Configuration Guide for Security Devices*

Copyright © 2017, Juniper Networks, Inc.

adaptive-shapers

Syntax

```
adaptive-shapers {
  adaptive-shaper-name {
    trigger type shaping-rate (percent percentage | rate);
  }
}
```

Hierarchy Level [edit class-of-service]

Release Information Statement introduced in Release 8.5 of Junos OS.

Description Define trigger types and associated rates. Adaptive shapers enable bandwidth limits on Frame Relay interfaces when the Services Router receives frames containing the backward explicit congestion notification (BECN) bit.

Options ***adaptive-shaper-name***—Name of the adaptive shaper.

The remaining statements are explained separately. See CLI Explorer.

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

Related Documentation

- adaptive-shaper on page 242
- *Junos OS Class of Service Configuration Guide for Security Devices*

Copyright © 2017, Juniper Networks, Inc.

application-traffic-control

Syntax

```
application-traffic-control {
  rate-limiters {
    rate-limiter-name {
      bandwidth-limit value-in-kbps;
      burst-size-limit value-in-bytes;
    }
  }
  rule-sets ruleset-name{
    {
      rule rule-name {
        match {
          application application-name1;
          application-any;
          application-group application-group-name;
          application-known;
          application-unknown;
        }
        then {
          dscp-code-point dscp-value;
          forwarding-class forwarding-class-name;
          log;
          loss-priority [ high | medium-high | medium-low | low ];
          rate-limit {
            loss-priority-high;
            client-to-server rate-limiter-name;
            server-to-client rate-limiter-name;
          }
        }
      }
    }
  }
}
```

Hierarchy Level [edit class-of-service]

Release Information Statement introduced in Junos OS Release 11.4.

Description Mark DSCP values for outgoing packets or apply rate limits based on the specified Layer 7 application types.

Options The remaining statements are explained separately. See CLI Explorer.

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

Related Documentation

- *Example: Configuring AppTrack*

Copyright © 2017, Juniper Networks, Inc.

buffer-size (Schedulers)

Syntax buffer-size (percent *percentage* | remainder | temporal *microseconds*);

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

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

Description Specify buffer size.

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

Default If you do not include this statement, the default scheduler transmission rate and buffer size percentages for queues 0 through 7 are 95, 0, 0, 5, 0, 0, 0, and 0 percent, respectively.

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

Range: 0 through 100

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

remainder—Remaining buffer available.

temporal *microseconds*—Buffer size as a temporal value. The queuing algorithm starts dropping packets when it queues more than a computed number of bytes. This maximum is computed by multiplying the logical interface speed by the configured temporal value.

Range: The ranges vary by platform as follows:

- For SRX Series Services Gateways: 1 through 2,000,000 microseconds.
- For vSRX instances: 1 through 32,000,000 microseconds.

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

Copyright © 2017, Juniper Networks, Inc.

Related Documentation

- *Managing Congestion on the Egress Interface by Configuring the Scheduler Buffer Size*
- *Buffer Size Temporal Value Ranges by Router Type*

classifiers (CoS)

Syntax

```
classifiers {
  (dscp | dscp-ipv6 | exp | ieee-802.1 | ieee-802.1ad | inet-precedence) classifier-name {
    forwarding-class forwarding-class-name {
    loss-priority (high | low | medium-high | medium-low) {
      code-point alias-or-bit-string ;
    }
    import (default | user-defined;
  }
}
```

Hierarchy Level [edit class-of-service]

Release Information Statement introduced in Junos OS Release 9.2

Description Configure a user-defined behavior aggregate (BA) classifier.

Options

- *classifier-name*—User-defined name for the classifier.
- import (default | *user-defined*)—Specify the template to use to map any code points not explicitly mapped in this configuration. For example, if the classifier is of type **dscp** and you specify **import default**, code points you do not map in your configuration will use the predefined DSCP default mapping; if you specify **import mymap**, for example, code points not mapped in the forwarding-class configuration would use the mappings in a user-defined classifier named **mymap**.
- forwarding-class *class-name*—Specify the name of the forwarding class. You can use the default forwarding class names or define new ones.
- loss-priority *level*—Specify a loss priority for this forwarding class: **high**, **low**, **medium-high**, **medium-low**.
- code-points (*alias* | *bits*)—Specify a code-point alias or the code points that map to this forwarding class.

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

Related Documentation

- *Understanding Interfaces*

Copyright © 2017, Juniper Networks, Inc.

code-points (CoS)

Syntax code-points [*aliases*] [*6–bit-patterns*];

Hierarchy Level [edit class-of-service classifiers *type classifier-name* forwarding-class *class-name*]

Release Information Statement introduced in Junos OS Release 8.5.

Description Specify one or more DSCP code-point aliases or bit sets for association with a forwarding class.

Options *aliases*—Name of the DSCP alias.

6–bit patterns—Value of the code-point bits, in decimal form.

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

copy-outer-dscp

Syntax copy-outer-dscp;

Hierarchy Level [dit security ipsec vpn *vpn-name*]

Release Information Statement introduced in Junos OS Release 15.1X49-D30.

Description Enable copying of Differentiated Services Code Point (DSCP) (outer DSCP+ECN) field from the outer IP header encrypted packet to the inner IP header plain text message on the decryption path. The benefit in enabling this feature is that after IPsec decryption, clear text packets can follow the inner CoS (DSCP+ECN) rules.

Default By default, the copy outer dscp feature is disabled.

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

Related Documentation

- *VPN Feature Guide for Security Devices*

Copyright © 2017, Juniper Networks, Inc.

default (CoS)

Syntax default;

Hierarchy Level [edit class-of-service virtual-channel-groups *group-name virtual-channel-name*]

Release Information Statement introduced in Release 8.5 of Junos OS.

Description Specify the default channel. You must configure one of the virtual channels in the group to be the default. Any traffic not explicitly directed to a virtual channel is transmitted by way of this default.

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

Related Documentation

- scheduler-map (CoS Virtual Channels) on page 271
- shaping-rate (CoS Virtual Channels) on page 274
- virtual-channel-group (CoS Interfaces) on page 277
- virtual-channel-groups on page 278
- virtual-channels on page 276
- *Junos OS Class of Service Configuration Guide for Security Devices*

Copyright © 2017, Juniper Networks, Inc.

dscp-code-point (CoS Host Outbound Traffic)

Syntax dscp-code-point *value*;

Hierarchy Level [edit class-of-service host-outbound-traffic]

Release Information Statement introduced in Junos OS Release 8.4.
Statement introduced before Junos OS Release 11.4 for EX Series switches.
Support for distributed protocol handler traffic introduced in Junos OS Release 13.2.

Description Specify the value of the DSCP bits in the type of service (ToS) field of host outbound traffic (packets generated by the local Routing Engine) as they are placed in the default or specified output queue on all egress interfaces. This statement does not affect transit traffic or incoming traffic.

If you use the **ping** operational mode command with the **tos *type-of-servce*** option, the value specified in this configuration statement overrides the DSCP value you specify in the **ping** command.

NOTE: Any DSCP rewrite rules configured on a 10-Gigabit Ethernet LAN/WAN PIC with SFP+ overwrite this DSCP value.

For egress interfaces hosted on MX Series routers, M120 routers, or Enhanced III FPCs in M320 routers, both Routing Engine sourced traffic and distributed protocol handler traffic are affected. For all other egress interfaces, only Routing Engine sourced traffic is affected.

Options **code-point**—Six-bit DSCP code point value.

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

Related Documentation

- *Understanding How Behavior Aggregate Classifiers Prioritize Trusted Traffic*
- *Default DSCP and DSCP IPv6 Classifiers*
- *Changing the Default Queuing and Marking of Host Outbound Traffic*.

Copyright © 2017, Juniper Networks, Inc.

egress-shaping-overhead

Syntax egress-shaping-overhead *number*;

Hierarchy Level [edit chassis fpc *slot-number* pic *pic-number* traffic-manager],
[edit chassis lcc *number* fpc *slot-number* pic *pic-number* traffic-manager]

Release Information Statement introduced in Junos OS Release 8.3.

Description Number of bytes to add to packet to determine shaped session packet length.

NOTE: On M Series and T Series routers with Gigabit Ethernet Intelligent Queuing 2 (IQ2) PICs and Enhanced IQ2 (IQ2E) PICs and on MX Series routers with Dense Port Concentrators (DPCs) only, to account for egress shaping overhead bytes added to output traffic on the line card, you must use the **egress-policer-overhead** statement to explicitly configure corresponding egress policing overhead for Layer 2 policers, MAC policers, or queue rate limits applied to output traffic on the line card.

NOTE: For MIC and MPC interfaces on MX Series routers, by default the value of **egress-shaping-overhead** is configured to 20, which means that the number of class-of-service (CoS) shaping overhead bytes to be added to the packets is 20. The interfaces on DPCs in MX Series routers, the default value is zero. For interfaces on PICs other than the 10-port 10-Gigabit Oversubscribed Ethernet (OSE) Type 4, you should configure **egress-shaping-overhead** to a minimum of 20 bytes to add a shaping overhead of 20 bytes to the packets.

NOTE: When you change the **egress-shaping-overhead** value, on M Series and T Series routers the PIC on which it is changed is restarted. On MX Series routers the DPC/MPC on which it is changed is restarted.

Options *number*—When traffic management (queuing and scheduling) is configured on the egress side, the number of CoS shaping overhead bytes to add to the packets on the egress interface.

Range:

- –63 through 192.
- –62 through 192 for vSRX.

Copyright © 2017, Juniper Networks, Inc.

NOTE: The L2 headers (DA/SA + VLAN tags) are automatically a part of the shaping calculation.

Required Privilege Level

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

Related Documentation

- *egress-policer-overhead*
- *Configuring CoS for L2TP Tunnels on ATM Interfaces*
- *ingress-shaping-overhead*
- *mode (Layer 2 Tunneling Protocol Shaping)*, *ingress-shaping-overhead*
- *traffic-manager*

Copyright © 2017, Juniper Networks, Inc.

forwarding-class (CoS Host Outbound Traffic)

Syntax forwarding-class *class-name*;

Hierarchy Level [edit class-of-service host-outbound-traffic]

Release Information Statement introduced in Junos OS Release 8.4.
Statement introduced before Junos OS Release 11.4 for EX Series switches.
Support for distributed protocol handler traffic introduced in Junos OS Release 13.2.

Description Specify the name of the forwarding class to which host outbound traffic is assigned on all egress interfaces. The output queue associated with the forwarding class must be properly configured on all interfaces. In the case of a restricted interface, the traffic flows through a restricted queue.

For egress interfaces hosted on MX Series routers, M120 routers, or Enhanced III FPCs in M320 routers, both Routing Engine sourced traffic and distributed protocol handler traffic are affected. For all other egress interfaces, only Routing Engine sourced traffic is affected.

This statement does not affect transit traffic or incoming traffic.

Default If you do not configure an output queue for host outbound traffic, the router uses the default queue assignments for host outbound traffic.

Options ***class-name***—Name of the forwarding class.

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

Related Documentation

- *Understanding How Forwarding Classes Assign Classes to Output Queues*
- Default Routing Engine Protocol Queue Assignments on page 72
- *Changing the Default Queuing and Marking of Host Outbound Traffic*.

 Copyright © 2017, Juniper Networks, Inc.

forwarding-classes (CoS)

Syntax

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

Hierarchy Level [edit class-of-service]

Release Information Statement introduced in Junos OS Release 8.5. Statement updated in Junos OS Release 11.4. The **spu-priority** option introduced in Junos OS Release 11.4R2.

Description Configure forwarding classes and assign queue numbers.

Options

- **class class-name**—Display the forwarding class name assigned to the internal queue number.

  NOTE: This option is supported only on SRX1500, SRX5400, SRX5600, and SRX5800.

  NOTE: AppQoS forwarding classes must be different from those defined for interface-based rewriters.

- **priority**—Fabric priority value:
  - **high**—Forwarding class' fabric queuing has high priority.
  - **low**—Forwarding class' fabric queuing has low priority.

  The default **priority** is **low**.
- **queue queue-number**—Specify the internal queue number to which a forwarding class is assigned.
- **spu-priority**—Services Processing Unit (SPU) priority queue, **high**, **medium-high**, **medium-low**, or **low**. The default **spu-priority** is **low**.

Copyright © 2017, Juniper Networks, Inc.

NOTE: The spu-priority option is only supported on SRX1500 devices and SRX5000 line devices.

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

Related Documentation
- *Example: Configuring AppQoS*

frame-relay-de (CoS Interfaces)

Syntax
frame-relay-de (*name* | default);

Hierarchy Level
[edit class-of-service interfaces *interface-name* unit *logical-unit-number* loss-priority-maps],
[edit class-of-service interfaces *interface-name* unit *logical-unit-number* rewrite-rules]

Release Information
Statement introduced in Release 8.5 of Junos OS.

Description
Assign the loss priority map or the rewrite rule to a logical interface.

Options
- **default**—Apply default loss priority map or default rewrite rule. The default loss priority map contains the following settings:

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

- The default rewrite rule contains the following settings:

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

- ***name***—Name of loss priority map or rewrite rule to be applied.

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

Related Documentation
- *Junos OS Class of Service Configuration Guide for Security Devices*

Copyright © 2017, Juniper Networks, Inc.

frame-relay-de (CoS Loss Priority)

Syntax

```
frame-relay-de map-name {
  loss-priority level code-point (0 | 1);
}
```

Hierarchy Level [edit class-of-service loss-priority-maps]

Release Information Statement introduced in Release 8.5 of Junos OS.

Description Define a Frame Relay discard-eligible bit loss-priority map.

Options

- ***level***—Level of loss priority to be applied based on the specified CoS values. The level can be **low**, **medium-low**, **medium-high**, or **high**.
- ***map-name***—Name of the loss-priority map.

  The remaining statements are explained separately. See CLI Explorer.

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

Related Documentation

- *Junos OS Class of Service Configuration Guide for Security Devices*

Copyright © 2017, Juniper Networks, Inc.

frame-relay-de (CoS Rewrite Rule)

Syntax

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

Hierarchy Level [edit class-of-service rewrite-rules]

Release Information Statement introduced in Release 8.5 of Junos OS.

Description Define a Frame Relay discard-eligible bit rewrite rule.

Options

- ***level***—Level of loss priority on which to base the rewrite rule. The loss priority level can be **low**, **medium-low**, **medium-high**, or **high**.
- ***rewrite-name***—Name of a rewrite-rules mapping configured at the **[edit class-of-service rewrite-rules]** hierarchy level.

The remaining statements are explained separately. See CLI Explorer.

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

Related Documentation

- *Junos OS Class of Service Configuration Guide for Security Devices*

Copyright © 2017, Juniper Networks, Inc.

host-outbound-traffic (Class-of-Service)

Syntax

```
host-outbound-traffic {
  dscp-code-point value;
  forwarding-class class-name;
  ieee-802.1 {
    default value;
    rewrite-rules;
  }
  protocol {
    isis-over-gre {
      dscp-code-point dscp-code-point;
    }
  }
}
```

Hierarchy Level [edit class-of-service]

Release Information

Statement introduced in Junos OS Release 8.4.
Statement introduced before Junos OS Release 11.4 for EX Series switches.
Support for **ieee-802.1** statement introduced in Junos OS Release 12.3.
Support for distributed protocol handler traffic introduced in Junos OS Release 13.2.
Support for **protocol** statement introduced in Junos OS Release 17.3 for MX Series and PTX Series devices.

Description Classify and mark host outbound traffic. This statement does not affect transit traffic or incoming traffic.

Default If you do not specify a forwarding class or DSCP value, the router uses the default queue and DSCP bit assignments for host outbound traffic.

Options The remaining statements are explained separately. See CLI Explorer.

Required Privilege Level

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

Related Documentation

- Default Routing Engine Protocol Queue Assignments on page 72
- *Default DSCP and DSCP IPv6 Classifiers*
- *Changing the Default Queuing and Marking of Host Outbound Traffic*.
- *Configuring a Global Default IEEE 802.1p Value for All Host Outbound Traffic*
- *Applying Egress Interface Rewrite Rules to the IEEE 802.1p Field for All Host Outbound Traffic on the Interface*

Copyright © 2017, Juniper Networks, Inc.

ingress-policer-overhead

Syntax ingress-policer-overhead *bytes*;

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

Release Information Statement introduced before Junos OS Release 11.1.
Statement introduced in Junos OS Release 15.1X49-D30 for vSRX.

Description Add the configured number of bytes to the length of a packet entering the interface.

Configure a policer overhead to control the rate of traffic received on an interface. Use this feature to help prevent denial-of-service (DoS) attacks or to enforce traffic rates to conform to the service-level agreement (SLA). When you configure a policer overhead, the configured policer overhead value (bytes) is added to the length of the final Ethernet frame. This calculated length of frame is used to determine the policer or the rate-limiting action.

Traffic policing combines the configured policy bandwidth limits and the burst size to determine how to meter the incoming traffic. If you configure a policer overhead on an interface, Junos OS adds those bytes to the length of incoming Ethernet frames. This added overhead fills each frame closer to the burst size, allowing you to control the rate of traffic received on an interface.

You can configure the policer overhead to rate-limit queues and Layer 2 and Layer 3 policers, for standalone (SA) and high-avalability (HA) deployments. The policer overhead and the shaping overhead can be configured simultaneously on an interface.

NOTE: vSRX supports policer overhead on Layer 3 policers only.

The policer overhead applies to all interfaces on the PIC. In the following example, Junos OS adds 10 bytes of overhead to all incoming Ethernet frames on ports ge-0/0/0 through ge-0/0/4.

```
set chassis fpc 0 pic 0 ingress-policer-overhead 10
```

NOTE: vSRX only supports fpc 0 pic 0. When you commit the ingress-policer-overhead statement, the vSRX takes the PIC offline and then back online.

You need to craft the policer overhead size to match your network traffic. A value that is too low will have minimal impact on traffic bursts. A value that is too high will rate-limit too much of your incoming traffic.

Copyright © 2017, Juniper Networks, Inc.

In this example, the policer overhead of 255 bytes is configured for ge-0/0/0 through ge-0/0/4. The firewall policer is configured to discard traffic when the burst size is over 1500 bytes. This policer is applied to ge-0/0/0 and ge 0/0/1. Junos OS adds 255 bytes to every Ethernet frame that comes into the configured ports. If, during a burst of traffic, the combined length of incoming frames and the overhead bytes exceeds 1500 bytes, the policer starts to discard further incoming traffic.

```
set chassis fpc 0 pic 0 ingress-policer-overhead 255
set interfaces ge-0/0/0 unit 0 family inet policer input overhead_policer
set interfaces ge-0/0/0 unit 0 family inet address 10.9.1.2/24
set interfaces ge-0/0/1 unit 0 family inet policer input overhead_policer
set interfaces ge-0/0/1 unit 0 family inet address 10.9.2.2/24
set firewall policer overhead_policer if-exceeding bandwidth-limit 32k
set firewall policer overhead_policer if-exceeding burst-size-limit 1500
set firewall policer overhead_policer then discard
```

Options

bytes—Number of bytes added to a frame entering an interface.

Range: 0–255 bytes

Default: 0

```
[edit chassis fpc 0 pic 0]
user@host# set ingress-policer-overhead 10;
```

Required Privilege Level

interface—To view this statement in the configuration.

interface-control—To add this statement to the configuration.

Related Documentation

- *ingress-shaping-overhead*
- *Policer Overhead to Account for Rate Shaping Overview*
- *Example: Configuring Policer Overhead to Account for Rate Shaping*
- *Configuring a Policer Overhead*
- *CoS on Enhanced IQ2 PICs Overview*

Copyright © 2017, Juniper Networks, Inc.

interfaces (CoS)

Syntax

```
interfaces
  interface-name {
    input-scheduler-map map-name ;
    input-shaping-rate rate ;
    scheduler-map map-name ;
    scheduler-map-chassis map-name ;
    shaping-rate rate ;
    unit logical-unit-number {
      adaptive-shaper adaptive-shaper-name ;
      classifiers {
        (dscp | dscp-ipv6 | exp | ieee-802.1 | inet-precedence)
        ( classifier-name | default);
      }
      forwarding-class class-name ;
      fragmentation-map map-name ;
      input-scheduler-map map-name ;
      input-shaping-rate (percent percentage | rate );
      input-traffic-control-profile profiler-name shared-instance instance-name ;
      loss-priority-maps {
        default;
        map-name ;
      }
      output-traffic-control-profile profile-name shared-instance instance-name ;
      rewrite-rules {
        dscp ( rewrite-name | default);
        dscp-ipv6 ( rewrite-name | default);
        exp ( rewrite-name | default) protocol protocol-types ;
        frame-relay-de ( rewrite-name | default);
        inet-precedence ( rewrite-name | default);
      }
      scheduler-map map-name ;
      shaping-rate rate ;
      virtual-channel-group group-name ;
    }
  }
}
```

Hierarchy Level [edit class-of-service interface *interface-name* unit *number*]

Release Information Statement introduced in Junos OS Release 8.5.

Description Associate the class-of-service configuration elements with an interface.

Options interface *interface-name* unit *number*—The user-specified interface name and unit number.

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

Copyright © 2017, Juniper Networks, Inc.

Related Documentation
- *Class of Service Feature Guide for Security Devices*

logical-interface-policer

Syntax logical-interface-policer;

Hierarchy Level [edit firewall policer *policer-name*]

Release Information Statement introduced in Junos OS Release 9.5.

Description Configure a logical interface (aggregate) policer.

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

Related Documentation
- Two-Color Policer Configuration Overview on page 37
- Example: Configuring a Two-Color Logical Interface (Aggregate) Policer on page 40

Copyright © 2017, Juniper Networks, Inc.

loss-priority (CoS Loss Priority)

Syntax loss-priority *level* code-points [*values*];

Hierarchy Level [edit class-of-service loss-priority-maps frame-relay-de *map-name*]

Release Information Statement introduced in Junos OS Release 9.2.

Description Map CoS values to a packet loss priority (PLP). In Junos OS, classifiers associate incoming packets with a forwarding class (FC) and PLP. PLPs allow you to set the priority for dropping packets. Typically, you mark packets exceeding some service level with a high loss priority—that is, a greater likelihood of being dropped.

Options ***level*** can be one of the following:

- **high**—Packet has high loss priority.
- **medium-high**—Packet has medium-high loss priority.
- **medium-low**—Packet has medium-low loss priority.
- **low**—Packet has low loss priority.

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

Related Documentation

- *Understanding Interfaces*
- Understanding Packet Loss Priorities on page 16

 Copyright © 2017, Juniper Networks, Inc.

loss-priority (CoS Rewrite Rules)

Syntax loss-priority *level*;

Hierarchy Level [edit class-of-service rewrite-rules *type rewrite-name* forwarding-class *class-name*]

Release Information Statement introduced in Junos OS Release 9.2.

Description Specify a loss priority to which to apply a rewrite rule. The rewrite rule sets the code-point aliases and bit patterns for a specific forwarding class and packet loss priority (PLP). The inputs for the map are the forwarding class and the PLP. The output of the map is the code-point alias or bit pattern.

Options ***level*** can be one of the following:

- **high**—The rewrite rule applies to packets with high loss priority.
- **low**—The rewrite rule applies to packets with low loss priority.
- **medium-high**—The rewrite rule applies to packets with medium-high loss priority.
- **medium-low**—The rewrite rule applies to packets with medium-low loss priority.

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

Related Documentation

- *Class of Service Feature Guide for Security Devices*

Copyright © 2017, Juniper Networks, Inc.

loss-priority-maps (CoS Interfaces)

Syntax

```
loss-priority-maps {
  frame-relay-de (map-name | default);
}
```

Hierarchy Level [edit class-of-service interfaces *interface-name* unit *logical-unit-number*]

Release Information Statement introduced in Junos OS Release 9.2.

Description Assign the loss priority map to a logical interface.

Options

- **default**—Apply default loss priority map. The default map contains the following:

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

- ***map-name***—Name of loss priority map to be applied.

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

Related Documentation

- *Understanding Interfaces*

loss-priority-maps (CoS)

Syntax

```
loss-priority-maps {
  frame-relay-de loss-priority-map-name {
    loss-priority (high | low | medium-high | medium-low) {
      code-points [bit-string];
    }
  }
}
```

Hierarchy Level [edit class-of-service]

Release Information Statement introduced in Junos OS Release 9.2.

Description Map the loss priority of incoming packets based on CoS values.

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

Related Documentation

- *Understanding Interfaces*

Copyright © 2017, Juniper Networks, Inc.

non-strict-priority-scheduling

Syntax non-strict-priority-scheduling;

Hierarchy Level [edit class-of-service non-strict-priority-scheduling]

Release Information Statement introduced in Junos OS Release 15.1X49-D80.

NOTE: This statement is supported only on SRX300, SRX320, SRX340, SRX345, SRX550M, SRX1500, and vSRX2.0 devices.

Description Configure non-strict priority scheduling to avoid starvation of lower-priority queues on SRX300, SRX320, SRX340, SRX345, SRX1500, SRX550M, and vSRX 2.0 devices.

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

Related Documentation

- Example: Configuring CoS Non-Strict Priority Scheduling on page 126

Copyright © 2017, Juniper Networks, Inc.

rate-limiters

Syntax

```
rate-limiters {
  rate-limiter-name {
    bandwidth-limit value-in-kbps;
    burst-size-limit value-in-bytes;
  }
}
```

Hierarchy Level [edit class-of-service application-traffic-control]

Release Information Statement introduced in Junos OS Release 11.4.

Description Share the available bandwidth and burst size of a device's PICs by defining rate limiter profiles and applying them in AppQoS rules.

Options

- ***rate-limiter-name***—Name of the rate limiter. It is applied in AppQoS rules to share device resources based on quality-of-service requirements.

  The combination of rate limiting parameters, namely bandwidth- limit and burst-size-limit rate limit, make up the rate limiter profile. A maximum of 16 profiles are allowed per device. The same profile can be used by multiple rate limiters. For example, a profile with a bandwidth-limit of 200 Kbps and a burst-limit of 130,000 bytes, could be used in several rate limiters.

  A maximum of 1000 rate limiters can be created. Rate limiters are defined for the device, and are assigned in rules in a rule set. A single rate limiter can be used multiple times within the same rule set. However, the rate limiter cannot be used in another rule set.

- **bandwidth-limit *value-in-Kbps***—Maximum number of kilobits to be transmitted per second for this rate limiter. Up to 2 GB of bandwidth can be provisioned among multiple rate limiters to share the resource proportionally.

- **burst-size-limit *value-in-bytes***—Maximum number of bytes to be transferred in a single burst or time-slice. This limit ensures that a high-priority transmission does not keep a lower priority transmission from transmitting.

NOTE: The number of **bandwidth-limit** and **burst-size-limit** combinations cannot exceed 16.

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

Related Documentation

- *Example: Configuring AppQoS*

Copyright © 2017, Juniper Networks, Inc.

rewrite-rules (CoS)

Syntax

```
rewrite-rules {
  type rewrite-name{
    import (rewrite-name | default);
    forwarding-class class-name {
      loss-priority level code-point [ aliases ] [ 6–bit-patterns ];
    }
  }
}
```

Hierarchy Level [edit class-of-service]

Release Information Statement introduced in Release 8.5 of Junos OS. **ieee-802.1ad** option introduced in 9.2 of Junos OS.

Description Specify a rewrite-rules mapping for the traffic that passes through all queues on the interface.

Options

- ***rewrite-name***—Name of a **rewrite-rules** mapping.
- ***type***—Traffic type.

  Values: **dscp**, **dscp-ipv6**, **exp**, **frame-relay-de** (J Series only), **ieee-802.1**, **ieee-802.1ad**, **inet-precedence**

The remaining statements are explained separately. See CLI Explorer.

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

Related Documentation

- rewrite-rules (CoS Interfaces) on page 268
- *Junos OS Class of Service Configuration Guide for Security Devices*

Copyright © 2017, Juniper Networks, Inc.

rewrite-rules (CoS Interfaces)

Syntax

```
rewrite-rules {
  dscp (rewrite-name | default);
  dscp-ipv6 (rewrite-name | default);
  exp (rewrite-name | default) protocol protocol-types;
  exp-push-push-push default;
  exp-swap-push-push default;
  ieee-802.1 (rewrite-name | default) vlan-tag (outer | outer-and-inner);
  inet-precedence (rewrite-name | default);
}
```

Hierarchy Level [edit class-of-service interfaces *interface-name* unit *logical-unit-number*]

Release Information Statement introduced in Release 8.5 of Junos OS.

Description Associate a rewrite-rules configuration or default mapping with a specific interface.

Options

- ***rewrite-name***—Name of a **rewrite-rules** mapping configured at the **[edit class-of-service rewrite-rules]** hierarchy level.
- **default**—The default mapping.

The remaining statements are explained separately. See CLI Explorer.

Required Privilege Level

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

Related Documentation

- rewrite-rules (CoS) on page 267
- *Class of Service Feature Guide for Security Devices*

Copyright © 2017, Juniper Networks, Inc.

rule-sets (CoS AppQoS)

Syntax

```
rule-sets {
  rule-set-name {
    rule rule-name {
      match {
        application application-name;
        application-any;
        application-group application-group-name;
        application-known;
        application-unknown;
      }
      then {
        dscp-code-point dscp-value ;
        forwarding-class forwarding-class-name;
        log;
        loss-priority [ high | medium-high | medium-low | low ];
        rate-limit {
          loss-priority-high;
          client-to-server rate-limiter-name;
          server-to-client rate-limiter-name;
        }
      }
    }
  }
}
```

Hierarchy Level [edit class-of-service application-traffic-control]

Release Information Statement introduced in Junos OS Release 11.4.

Description Defines AppQoS rule sets and the rules that establish priorities based on quality-of-service requirements for the associated applications. AppQoS rules can be included in policy statements to implement application-aware quality of service control.

Options

- ***rule-set-name***—Name used to refer to a collection of AppQoS rules.
- **rule *rule-name***—Name applied to the match criteria and resulting actions that control the quality-of-service provided to any matching applications.
- **application *application-name***—Name of the application to be used as match criteria for the rule.
- **application-any** —Any application encountering this rule. Note that when you use this specification, all application matching ends. Any application rule following this one will never be encountered.
- **application-group *application-group-name***—Group of applications to be used as match criteria for the rule. Both applications and application groups can be match criteria for a single rule.

Copyright © 2017, Juniper Networks, Inc.

- **application-known**—Match criteria specifying any session that is identified, but its corresponding application is not specified.
- **application-unknown**—Match criteria specifying any session that is not identified.
- **forwarding-class *forwarding-class-name***—The AppQoS class with which matching applications will be marked. This field identifies the rewriter that has marked the DSCP value . Therefore, the AppQoS forwarding class must be different from those used by IDP or firewall filters. With this class specified, firewall filter class will not overwrite the existing DSCP value.
- **dscp-code-point**—DSCP alias or bit map with which matching applications will be marked to establish the output queue. This value can be marked by rewriters from IDP, AppQoS, or a firewall filter. The forwarding-class value identifies which rewriter has re-marked the packet with the current DSCP value. If a packet triggers all three rewriters, IDP takes precedence over AppQoS, which takes precedence over a firewall filter.
- **loss-priority**—Loss priority with which matching applications will be marked. This value is used to determine the likelihood that a packet would be dropped when encountering congestion. A high loss priority means that there is an 80% chance of packet loss in congestion. Possible values are high, medium-high, medium-low and low.
- **rate-limit**—Rate limiters to be associated with client-to-server and with server-to-client traffic for this application. The rate limiter profile defines maximum speed and volume limits for matching applications.
- **log**—AppQoS event logging.

Required Privilege Level

security—To view this statement in the configuration.
security-control—To add this statement to the configuration.

Related Documentation

- *Example: Configuring AppQoS*

Copyright © 2017, Juniper Networks, Inc.

scheduler-map (CoS Virtual Channels)

Syntax scheduler-map *map-name*;

Hierarchy Level [edit class-of-service virtual-channel-groups *group-name virtual-channel-name*]

Release Information Statement introduced in Junos OS Release 9.2.

Description Apply a scheduler map to this virtual channel.

Options *map-name*—Name of the scheduler map.

The remaining statements are explained separately. See CLI Explorer.

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

Related Documentation

- default (CoS) on page 248
- shaping-rate (CoS Virtual Channels) on page 274
- virtual-channel-group (CoS Interfaces) on page 277
- virtual-channel-groups on page 278
- virtual-channels on page 276

Copyright © 2017, Juniper Networks, Inc.

shaping-rate (CoS Adaptive Shapers)

Syntax shaping-rate (percent *percentage* | *rate*);

Hierarchy Level [edit class-of-service adaptive-shapers *adaptive-shaper-name* trigger *type*]

Release Information Statement introduced in Release 8.5 of Junos OS.

Description Define the list of trigger types and associated rates.

Options

- **percent** ***percentage***—Shaping rate as a percentage of the available interface bandwidth.

  Range: 0 through 100 percent

- ***rate***—Peak rate, in bits per second (bps). You can specify a value in bits per second either as a complete decimal number or as a decimal number followed by the abbreviation **k** (1000), **m** (1,000,000), or **g** (1,000,000,000).

  Range: 3200 through 32,000,000,000 bps

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

Related Documentation

- trigger (CoS) on page 275
- *Junos OS Class of Service Configuration Guide for Security Devices*

 Copyright © 2017, Juniper Networks, Inc.

shaping-rate (CoS Interfaces)

Syntax shaping-rate *rate*;

Hierarchy Level [edit class-of-service interfaces *interface-name*],
[edit class-of-service interfaces *interface-name* unit *logical-unit-number*]

Release Information Statement introduced in Junos OS Release 9.2.

Description For logical interfaces on which you configure packet scheduling, configure traffic shaping by specifying the amount of bandwidth to be allocated to the logical interface.

Logical and physical interface traffic shaping can be configured together. This means you can include the **shaping-rate** statement at the **[edit class-of-service interfaces interface *interface-name*]** hierarchy level *and* the **[edit class-of-service interfaces interface *interface-name* unit *logical-unit-number*]** hierarchy level. If you configure traffic shaping at both the logical and physical interface levels, the logical interface shaping credit is checked and updated before the physical interface shaping credit.

Alternatively, you can configure a shaping rate for a logical interface and oversubscribe the physical interface by including the **shaping-rate** statement at the **[edit class-of-service traffic-control-profiles]** hierarchy level. With this configuration approach, you can independently control the delay-buffer rate.

Default If you do not include this statement at the **[edit class-of-service interfaces interface *interface-name* unit *logical-unit-number*]** hierarchy level, the default logical interface bandwidth is the average of unused bandwidth for the number of logical interfaces that require default bandwidth treatment. If you do not include this statement at the **[edit class-of-service interfaces interface *interface-name*]** hierarchy level, the default physical interface bandwidth is the average of unused bandwidth for the number of physical interfaces that require default bandwidth treatment.

Options *rate*—Peak rate, in bits per second (bps). You can specify a value in bits per second either as a complete decimal number or as a decimal number followed by the abbreviation **k** (1000), **m** (1,000,000), or **g** (1,000,000,000).

Range: For logical interfaces, 1000 through 6,400,000,000,000 bps.

For physical interfaces, 1000 through 6,400,000,000,000 bps.

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

Related Documentation

- Class of Service Feature Guide for Security Devices

Copyright © 2017, Juniper Networks, Inc.

shaping-rate (CoS Virtual Channels)

Syntax shaping-rate (percent *percentage* | *rate*);

Hierarchy Level [edit class-of-service virtual-channel-groups *group-name virtual-channel-name*]

Release Information Statement introduced in Release 8.5 of Junos OS.

Description Define the shaping rates to be associated with the virtual channel.

Options

- **percent** ***percentage***—Shaping rate as a percentage of the available interface bandwidth.

  Range: 0 through 100 percent

- ***rate***—Peak rate, in bits per second (bps). You can specify a value in bits per second either as a complete decimal number or as a decimal number followed by the abbreviation **k** (1000), **m** (1,000,000), or **g** (1,000,000,000).

  Range: 3200 through 32,000,000,000 bps

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

Related Documentation

- default (CoS) on page 248
- scheduler-map (CoS Virtual Channels) on page 271
- virtual-channel-group (CoS Interfaces) on page 277
- virtual-channel-groups on page 278
- virtual-channels on page 276
- *Junos OS Class of Service Configuration Guide for Security Devices*

Copyright © 2017, Juniper Networks, Inc.

trigger (CoS)

Syntax trigger *type* shaping-rate (percent *percentage* | *rate*);

Hierarchy Level [edit class-of-service adaptive-shapers *adaptive-shaper-name*]

Release Information Statement introduced in Release 8.5 of Junos OS.

Description Specify a trigger type and its associated rate.

Options ***type***—The type of trigger. Currently, the trigger type can be **becn** only.

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

Related Documentation
- shaping-rate (CoS Adaptive Shapers) on page 272
- *Junos OS Class of Service Configuration Guide for Security Devices*

tunnel-queuing

Syntax tunnel-queuing;

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

Release Information Statement modified in Release 9.0 of Junos OS.

Description Enable class-of-service (CoS) queuing for generic routing encapsulation (GRE) and IP-IP tunnels.

NOTE: The tunnel-queuing option is not supported in chassis cluster mode.

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

Copyright © 2017, Juniper Networks, Inc.

virtual-channels

Syntax

```
virtual-channels {
  virtual-channel-name;
}
```

Hierarchy Level [edit class-of-service]

Release Information Statement introduced in Release 8.5 of Junos OS.

Description Specify a list of virtual channels.

Each virtual channel has eight transmission queues.

Options ***virtual-channel-name***—Name of the virtual channel.

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

Related Documentation

- default (CoS) on page 248
- scheduler-map (CoS Virtual Channels) on page 271
- shaping-rate (CoS Virtual Channels) on page 274
- virtual-channel-group (CoS Interfaces) on page 277
- virtual-channel-groups on page 278
- *Junos OS Class of Service Configuration Guide for Security Devices*

Copyright © 2017, Juniper Networks, Inc.

virtual-channel-group (CoS Interfaces)

Syntax virtual-channel-group *group-name*;

Hierarchy Level [edit class-of-service interfaces *interface-name* unit *logical-unit-number*]

Release Information Statement introduced in Release 8.5 of Junos OS.

Description Assign a virtual channel group to a logical interface.

If you apply a virtual channel group to multiple logical interfaces, separate queues are created on each of the interfaces. The same virtual channel names are used on all the interfaces. You can specify the scheduler and shaping rates in the virtual channels in percentages so that you can apply the same virtual channel group to logical interfaces with different available bandwidths.

Options ***group-name***—Name of the virtual channel group.

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

Related Documentation

- default (CoS) on page 248
- scheduler-map (CoS Virtual Channels) on page 271
- shaping-rate (CoS Virtual Channels) on page 274
- virtual-channels on page 276
- virtual-channel-groups on page 278
- *Junos OS Class of Service Configuration Guide for Security Devices*

Copyright © 2017, Juniper Networks, Inc.

virtual-channel-groups

Syntax

```
virtual-channel-groups {
  virtual-channel-group-name {
    virtual-channel-name {
      scheduler-map map-name;
      shaping-rate (percent percentage | rate);
      default;
    }
  }
}
```

Hierarchy Level [edit class-of-service]

Release Information Statement introduced in Release 8.5 of Junos OS.

Description Associate a virtual channel with a scheduler map and a shaping rate. Virtual channels and virtual channel groups enable you to direct traffic into a virtual channel and apply bandwidth limits to the channel.

Options *group-name*—Name of the virtual channel group.

The remaining statements are explained separately. See CLI Explorer.

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

Related Documentation

- default (CoS) on page 248
- scheduler-map (CoS Virtual Channels) on page 271
- shaping-rate (CoS Virtual Channels) on page 274
- virtual-channel-group (CoS Interfaces) on page 277
- virtual-channels on page 276
- *Junos OS Class of Service Configuration Guide for Security Devices*

Copyright © 2017, Juniper Networks, Inc.

CHAPTER 17

Operational Commands

- show class-of-service application-traffic-control counter
- show class-of-service application-traffic-control statistics rate-limiter
- show class-of-service application-traffic-control statistics rule
- show class-of-service forwarding-class

Copyright © 2017, Juniper Networks, Inc.

show class-of-service application-traffic-control counter

Syntax show class-of-service application-traffic-control counter

Release Information Command introduced in Junos OS Release 11.4.

Description Display AppQoS DSCP marking and honoring statistics based on Layer 7 application classifiers.

Required Privilege Level view

Related Documentation
- *Example: Configuring AppQoS*

List of Sample Output show class-of-service application-traffic-control counter on page 280

Output Fields Table 52 on page 280 lists the output fields for the **show class-of-service application-traffic-control counter** command. Output fields are listed in the approximate order in which they appear.

Table 52: show class-of-service application-traffic-control counter Output Fields

Field Name	Field Description
pic	PIC number of the accumulated statistics. NOTE: The PIC number is always displayed as **0** for SRX300, SRX320, SRX340, SRX345, SRX550M, and SRX1500 devices.
Sessions processed	The number of sessions where the class of service was checked.
Sessions marked	The number of sessions marked based on application-aware DSCP marking.
Sessions honored	The number of sessions honored based on application-aware traffic honoring.
Sessions rate limited	The number of sessions that have been rate limited.
Client-to-server flows rate limited	The number of client-to-server flows that have been rate limited.
Server-to-client flows rate limited	The number of server-to-client flows that have been rate limited.

Sample Output

show class-of-service application-traffic-control counter

```
user@host> show class-of-service application-traffic-control counter
pic: 2/1
  Counter type                                                Value
```


Copyright © 2017, Juniper Networks, Inc.

```
  Sessions processed                                300
  Sessions marked                                   200
  Sessions honored                                  0
  Sessions rate limited                             100
  Client-to-server flows rate limited               100
  Server-to-client flows rate limited               70

pic: 2/0
  Counter type                                      Value
  Sessions processed                                400
  Sessions marked                                   300
  Sessions honored                                  0
  Sessions rate limited                             200
  Client-to-server flows rate limited               200
  Server-to-client flows rate limited               100
```

Copyright © 2017, Juniper Networks, Inc.

show class-of-service application-traffic-control statistics rate-limiter

Syntax show class-of-service application-traffic-control statistics rate-limiter

Release Information Command introduced in Junos OS Release 11.4.

Description Display AppQoS real-time run information about application rate limiting of current or recent sessions.

Required Privilege Level view

Related Documentation
- *Example: Configuring AppQoS*

List of Sample Output show class-of-service application-traffic-control statistics rate-limiter on page 282

Output Fields Table 53 on page 282 lists the output fields for the **show class-of-service application-traffic-control statistics rate-limiter** command. Output fields are listed in the approximate order in which they appear.

Table 53: show class-of-service application-traffic-control statistics rate-limiter Output Fields

Field Name	Field Description
pic	PIC number. NOTE: The PIC number is always displayed as **0** for SRX300, SRX320, SRX340, SRX345, and SRX550M devices.
Ruleset	The rule set applied on the session.
Application	The application match for applying the rule set.
Client-to-server	The rate limiter applied from client to server.
Rate(kbps)	The rate in the client-to-server direction
Server-to-client	The rate limiter applied from server to client.
Rate(kbps)	The rate in the server-to-client direction.

Sample Output

show class-of-service application-traffic-control statistics rate-limiter

```
user@host> show class-of-service application-traffic-control statistics rate-limiter
pic: 2/1
  Ruleset       Application  Client-to-server   Rate(kbps)   Server-to-client
```


Copyright © 2017, Juniper Networks, Inc.

```
Rate(kbps)
  my-ruleset-1  HTTP         my-http-c2s-rl     10000000    my-http-s2c-rl
20000000
  my-ruleset-2  HTTP         my-http-c2s-rl-2   20000000    my-http-s2c-rl-2
30000000
  my-ruleset-2  FTP          my-ftp-c2s-rl      50000       my-ftp-s2c-rl
50000
  ...

pic: 2/0
  Ruleset       Application  Client-to-server   Rate(kbps)   Server-to-client
Rate(kbps)
  my-ruleset-1  HTTP         my-http-c2s-rl     10000000    my-http-s2c-rl
20000000
  my-ruleset-2  HTTP         my-http-c2s-rl-2   20000000    my-http-s2c-rl-2
30000000
  my-ruleset-2  FTP          my-ftp-c2s-rl      50000       my-ftp-s2c-rl
50000
```

Copyright © 2017, Juniper Networks, Inc.

show class-of-service application-traffic-control statistics rule

Syntax show class-of-service application-traffic-control statistics rule

Release Information Command introduced in Junos OS Release 11.4.

Description Display AppQoS counters identifying rule hits.

Required Privilege Level view

Related Documentation
- *Example: Configuring AppQoS*

List of Sample Output show class-of-service application-traffic-control statistics rule on page 284

Output Fields Table 54 on page 284 lists the output fields for the **show class-of-service application-traffic-control statistics rule** command. Output fields are listed in the approximate order in which they appear.

Table 54: show class-of-service application-traffic-control statistics rule Output Fields

Field Name	Field Description
pic	PIC number where the rule is applied. NOTE: The PIC number is always displayed as **0** for for SRX300, SRX320, SRX340, SRX345, and SRX550M devices.
Ruleset	The rule set containing the rule.
Rule	The rule to which the statistic applies.
Hits	The number of times a match for the rule was encountered.

Sample Output

show class-of-service application-traffic-control statistics rule

```
user@host> show class-of-service application-traffic-control statistics rule
pic: 2/0
  Ruleset          Rule           Hits
  my-ruleset-1     ftp-rule       100
  my-ruleset-1     httpp-rule     100
  my-ruleset-2     telnet-rule    300
  my-ruleset-2     smtp-rule      300
  ...

pic: 2/1
  Ruleset          Rule           Hits
  my-ruleset-1     ftp-rule       200
```


Copyright © 2017, Juniper Networks, Inc.

```
my-ruleset-1     httpp-rule     300
my-ruleset-2     telnet-rule    400
my-ruleset-2     smtp-rule      500
```

Copyright © 2017, Juniper Networks, Inc.

show class-of-service forwarding-class

Syntax show class-of-service forwarding-class

Release Information Command introduced before Junos OS Release 12.1.

Description Display mapping of forwarding class names to queues.

Required Privilege Level view

Related Documentation
- Forwarding Classes Overview on page 57

List of Sample Output show class-of-service forwarding-class on page 286

Output Fields Table 55 on page 286 lists the output fields for the **show class-of-service forwarding-class** command. Output fields are listed in the approximate order in which they appear.

Table 55: show class-of-service forwarding-class Output Fields

Field Name	Field Description
Forwarding class	Forwarding class name.
ID	ID number assigned to the forwarding class.
Queue	Queue number.
Restricted queue	Restricted queue number.
Fabric priority	Fabric priority, either low or high.
Policing priority	Layer 2 policing, either premium or normal.
SPU priority	Services Processing Unit (SPU) priority queue, either high or low.

Sample Output

show class-of-service forwarding-class

```
user@host> show class-of-service forwarding-class
Forwarding class        ID  Queue  Restricted queue  Fabric priority  Policing
priority  SPU priority
  best-effort            0   0          0             low              normal
          low
  expedited-forwarding   1   1          1             low              normal
          high
  assured-forwarding     2   2          2             low              normal
          low
```


Copyright © 2017, Juniper Networks, Inc.

```
                    network-control        3   3          3             low                 normal
                            low
```

Copyright © 2017, Juniper Networks, Inc.