OcNOS®

Open Compute Network Operating System for Service Providers Version 7.0.0

Quality of Service Guide

February 2026

IP Infusion Inc. Proprietary

© 2026 IP Infusion Inc. All Rights Reserved.

This documentation is subject to change without notice. The software described in this document and this documentation are furnished under a license agreement or nondisclosure agreement. The software and documentation may be used or copied only in accordance with the terms of the applicable agreement. No part of this publication may be reproduced, stored in a retrieval system, or transmitted in any form or any means electronic or mechanical, including photocopying and recording for any purpose other than the purchaser's internal use without the written permission of IP Infusion Inc.

IP Infusion Inc.

3979 Freedom Circle

Suite 900

Santa Clara, California 95054

http://www.ipinfusion.com/

For support, questions, or comments via E-mail, contact:

support@ipinfusion.com

Trademarks:

IP Infusion and OcNOS are trademarks or registered trademarks of IP Infusion. All other trademarks, service marks, registered trademarks, or registered service marks are the property of their respective owners.

Use of certain software included in this equipment is subject to the IP Infusion, Inc. End User License Agreement at http://www.ipinfusion.com/license. By using the equipment, you accept the terms of the End User License Agreement.

Contents

© 2026 IP Infusion Inc. Proprietary

Preface

This guide describes how to configure OcNOS.

Audience

This guide is intended for network administrators and other engineering professionals who configure OcNOS.

Conventions

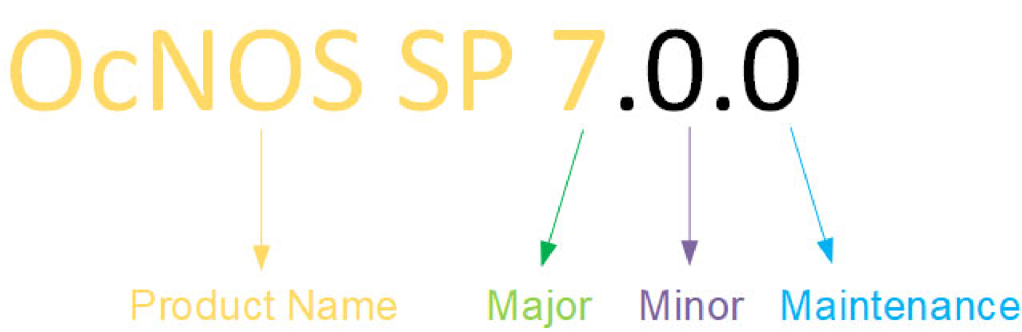
Table 1 on page 7 shows the conventions used in this guide.

Table 1: Conventions

Convention	Description
Italics	Emphasized terms or titles of books
Note:	Special instructions, suggestions, or warnings
`monospaced type`	Code elements such as commands, parameters, files, and directories

IP Infusion Product Release Version

Each integer in release number indicates Major, Minor, and Maintenance release versions. Build numbers that follow the release numbers are for internal tracking and verification of the software build process and are visible to customers as part of the software version number.



Product Name: IP Infusion Product Family

Major Version: New customer-facing functionality that represents a significant change to the code base; including, a significant marketing change or direction in the product.

Minor Version: Enhancements or extensions to existing features, changes to address external needs, or internal ements might be motivated by improvements to satisfy new sales regions or marketing initiatives.

Maintenance Version: A collection of product bugs or hotfixes usually scheduled every 30 or 60 days, based on the number of hotfixes.

Related Documentation

For information about installing OcNOS, see the *Installation Guide* for your platform.

Feature Availability

Each OcNOS SKU contains a set of supported features. For a list of available features based on the SKU that you purchased. Refer to the *Feature Matrix*.

Migration Guide

Check the *Migration Guide* for necessary configuration changes before migrating from one version of OcNOS to another.

IP Maestro Support

Monitor devices running OcNOS Release 6.3.4-70 and above using IP Maestro software.

Technical Support

IP Infusion maintains an online technical support site that provides a variety of technical support programs for licensed OcNOS customers at the Technical Assistance Center.

Customers and partners enjoy full access to the support website. The site allows customers and partners to open technical support calls, update open calls with new information, and review the status of open or closed calls. The password-protected site includes technical documentation, Release Notes, and descriptions of service offerings.

Technical Sales

Contact the IP Infusion sales representative for more information about the OcNOS solution.

Technical Documentation

For core commands and configuration procedures, visit: Product Documentation.

For training videos, visit: OcNOS Free Training Videos.

For a list of supported platforms and SKUs of OcNOS features, refer to the OcNOS Feature Matrix.

Disclaimer

The global documentation site is evolving to provide an enhanced website user experience for select topics included in this release. Some guides are now available outside the existing documentation library and can be accessed directly from custom documentation landing pages. These guides offer robust in-built search functionality.

For the latest documentation, visit the product-specific documentation landing page and select the relevant guide.

Comments

If you have comments, or need to report a problem with the content, contact techpubs@ipinfusion.com.

Command Line Interface

This chapter introduces the OcNOS Command Line Interface (CLI) and how to use its features.

Overview

You use the CLI to configure, monitor, and maintain OcNOS devices. The CLI is text-based and each command is usually associated with a specific task.

You can give the commands described in this manual locally from the console of a device running OcNOS or remotely from a terminal emulator such as `putty` or `xterm`. You can also use the commands in scripts to automate configuration tasks.

Chapter Organization

The chapters in command references are organized as described in Command Description Format.

The chapters in configuration guides are organized into these major sections:

- An overview that explains a configuration in words
- Topology with a diagram that shows the devices and connections used in the configuration
- Configuration steps in a table for each device where the left-hand side shows the commands you enter and the right-hand side explains the actions that the commands perform
- Validation which shows commands and their output that verify the configuration

Command Line Interface Help

You access the CLI help by entering a full or partial command string and a question mark "?". The CLI displays the command keywords or parameters along with a short description. For example, at the CLI command prompt, type:

```
> show ?
```

The CLI displays this keyword list with short descriptions for each keyword:

```
show ?
  application-priority            Application Priority
  arp                             Internet Protocol (IP)
  bfd                             Bidirectional Forwarding Detection (BFD)
  bgp                             Border Gateway Protocol (BGP)
  bi-lsp                          Bi-directional lsp status and configuration
  bridge                          Bridge group commands
  ce-vlan                         COS Preservation for Customer Edge VLAN
  class-map                       Class map entry
  cli                             Show CLI tree of current mode
  clns                            Connectionless-Mode Network Service (CLNS)
  control-adjacency               Control Adjacency status and configuration
  control-channel                 Control Channel status and configuration
  cspf                            CSPF Information
  customer                        Display Customer spanning-tree
  cvlan                           Display CVLAN information
  debugging                       Debugging functions
```

```
  etherchannel                    LACP etherchannel
  ethernet                        Layer-2
...
```

If you type the ? in the middle of a keyword, the CLI displays help for that keyword only.

```
> show de?
debugging  Debugging functions
```

If you type the ? in the middle of a keyword, but the incomplete keyword matches several other keywords, OcNOS displays help for all matching keywords.

```
> show i? (CLI does not display the question mark).
interface  Interface status and configuration
ip         IP information
isis       ISIS information
```

Command Completion

The CLI can complete the spelling of a command or a parameter. Begin typing the command or parameter and then press the tab key. For example, at the CLI command prompt type `sh`:

```
> sh
```

Press the tab key. The CLI displays:

```
> show
```

If the spelling of a command or parameter is ambiguous, the CLI displays the choices that match the abbreviation. Type `show i` and press the tab key. The CLI displays:

```
> show i
  interface  ip         ipv6       isis
> show i
```

The CLI displays the `interface` and `ip` keywords. Type `n` to select `interface` and press the tab key. The CLI displays:

```
> show in
> show interface
```

Type ? and the CLI displays the list of parameters for the `show interface` command.

```
> show interface
  IFNAME  Interface name
  |       Output modifiers
  >       Output redirection
  <cr>
```

The CLI displays the only parameter associated with this command, the `IFNAME` parameter.

Command Abbreviations

The CLI accepts abbreviations that uniquely identify a keyword in commands. For example:

```
> sh int xe0
```

is an abbreviation for:

```
> show interface xe0
```

Command Line Errors

Any unknown spelling causes the CLI to display the error `Unrecognized command` in response to the ?. The CLI displays the command again as last entered.

```
> show dd?
% Unrecognized command
> show dd
```

When you press the Enter key after typing an invalid command, the CLI displays:

```
(config)#router ospf here
                         ^
% Invalid input detected at '^' marker.
```

where the ^ points to the first character in error in the command.

If a command is incomplete, the CLI displays the following message:

```
> show
% Incomplete command.
```

Some commands are too long for the display line and can wrap mid-parameter or mid-keyword, as shown below. This does *not* cause an error and the command performs as expected:

```
area 10.10.0.18 virtual-link 10.10.0.19 authent
ication-key 57393
```

Command Negation

Many commands have a `no` form that resets a feature to its default value or disables the feature. For example:

- The `ip address` command assigns an IPv4 address to an interface
- The `no ip address` command removes an IPv4 address from an interface

Syntax Conventions

Table 2 on page 12 describes the conventions used to represent command syntax in this reference.

Table 2: Syntax conventions

Convention	Description	Example
`monospaced font`	Command strings entered on a command line	`show ip ospf`
`lowercase`	Keywords that you enter exactly as shown in the command syntax.	`show ip ospf`
`UPPERCASE`	See Variable Placeholders	`IFNAME`
`()`	Optional parameters, from which you must select one. Vertical bars delimit the selections. Do not enter the parentheses or vertical bars as part of the command.	`(A.B.C.D|<0-4294967295>)`

Table 2: Syntax conventions (Continued)

<table>
<tr><th>Convention</th><th>Description</th><th>Example</th></tr>
<tr><td>()</td><td>Optional parameters, from which you select one or none. Vertical bars delimit the selections. Do not enter the parentheses or vertical bars as part of the command.</td><td>(A.B.C.D|<0-4294967295>|)</td></tr>
<tr><td>()</td><td>Optional parameter which you can specify or omit. Do not enter the parentheses or vertical bar as part of the command.</td><td>(IFNAME|)</td></tr>
<tr><td>{}</td><td>Optional parameters, from which you must select one or more. Vertical bars delimit the selections. Do not enter the braces or vertical bars as part of the command.</td><td>{intra-area <1-255>|inter-area <1-255>|external <1-255>}</td></tr>
<tr><td>[]</td><td>Optional parameters, from which you select zero or more. Vertical bars delimit the selections. Do not enter the brackets or vertical bars as part of the command.</td><td>[<1-65535>|AA:NN|internet|local-AS|no-advertise|no-export]</td></tr>
<tr><td>?</td><td>Nonrepeatable parameter. The parameter that follows a question mark can only appear once in a command string. Do not enter the question mark as part of the command.</td><td>?route-map WORD</td></tr>
<tr><td>.</td><td>Repeatable parameter. The parameter that follows a period can be repeated more than once. Do not enter the period as part of the command.</td><td>set as-path prepend .<1-65535></td></tr>
</table>

Variable Placeholders

Table 3 on page 13 shows the tokens used in command syntax use to represent variables for which you supply a value.

Table 3: Variable placeholders

Token	Description
WORD	A contiguous text string (excluding spaces)
LINE	A text string, including spaces; no other parameters can follow this parameter
IFNAME	Interface name whose format varies depending on the platform; examples are: eth0, Ethernet0, ethernet0, xe0
A.B.C.D	IPv4 address
A.B.C.D/M	IPv4 address and mask/prefix
X:X::X:X	IPv6 address
X:X::X:X/M	IPv6 address and mask/prefix
HH:MM:SS	Time format

© 2026 IP Infusion Inc. Proprietary

Table 3: Variable placeholders

Token	Description
`AA:NN`	BGP community value
`XX:XX:XX:XX:XX:XX`	MAC address

`<1-5>`
`<1-65535>`
`<0-2147483647>`
`<0-4294967295>`
	Numeric range

Command Description Format

Table 4 on page 14 explains the sections used to describe each command in this reference.

Table 4: Command descriptions

Section	Description
Command Name	The name of the command, followed by what the command does and when should it be used
Command Syntax	The syntax of the command
Parameters	Parameters and options for the command
Default	The state before the command is executed
Command Mode	The mode in which the command runs; see Command Modes
Applicability	The command introduced in a specific release version and modified or updated in subsequent versions.
Example	An example of the command being executed

Keyboard Operations

Table 5 on page 14 lists the operations you can perform from the keyboard.

Table 5: Keyboard operations

Key combination	Operation
Left arrow or Ctrl+b	Moves one character to the left. When a command extends beyond a single line, you can press left arrow or Ctrl+b repeatedly to scroll toward the beginning of the line, or you can press Ctrl+a to go directly to the beginning of the line.
Right arrow or Ctrl-f	Moves one character to the right. When a command extends beyond a single line, you can press right arrow or Ctrl+f repeatedly to scroll toward the end of the line, or you can press Ctrl+e to go directly to the end of the line.

© 2026 IP Infusion Inc. Proprietary

Table 5: Keyboard operations (Continued)

Key combination	Operation
Esc, b	Moves back one word
Esc, f	Moves forward one word
Ctrl+e	Moves to end of the line
Ctrl+a	Moves to the beginning of the line
Ctrl+u	Deletes the line
Ctrl+w	Deletes from the cursor to the previous whitespace
Alt+d	Deletes the current word
Ctrl+k	Deletes from the cursor to the end of line
Ctrl+y	Pastes text previously deleted with Ctrl+k, Alt+d, Ctrl+w, or Ctrl+u at the cursor
Ctrl+t	Transposes the current character with the previous character
Ctrl+c	Ignores the current line and redisplays the command prompt
Ctrl+z	Ends configuration mode and returns to exec mode
Ctrl+l	Clears the screen
Up Arrow or Ctrl+p	Scroll backward through command history
Down Arrow or Ctrl+n	Scroll forward through command history

Show Command Modifiers

Note: The show command output included in the guides is for illustration purposes only. Based on the combination of features enabled and ongoing enhancements made to the commands, the output for these commands may vary. For instance, the actual command output may differ depending on the software version, configuration, and platform. Field names, values, and formats are subject to change.

You can use two tokens to modify the output of a `show` command. Enter a question mark to display these tokens:

```
# show users ?
  | Output modifiers
  > Output redirection
```

You can type the | (vertical bar character) to use output modifiers. For example:

```
> show rsvp | ?
  begin     Begin with the line that matches
  exclude   Exclude lines that match
  include   Include lines that match
  last      Last few lines
  redirect  Redirect output
```

© 2026 IP Infusion Inc. Proprietary

Begin Modifier

The `begin` modifier displays the output beginning with the first line that contains the input string (everything typed after the `begin` keyword). For example:

```
# show running-config | begin xe1
...skipping
interface xe1
 ipv6 address fe80::204:75ff:fee6:5393/64
!
interface xe2
 ipv6 address fe80::20d:56ff:fe96:725a/64
!
line con 0
 login
!
end
```

You can specify a regular expression after the `begin` keyword, This example begins the output at a line with either “xe2” or “xe4”:

```
# show running-config | begin xe[2-4]

...skipping
interface xe2
 shutdown
!
interface xe4
 shutdown
!
interface svlan0.1
 no shutdown
!
route-map myroute permit 2
!
route-map mymap1 permit 10
!
route-map rmap1 permit 2
!
line con 0
 login
line vty 0 4
 login
!
end
```

Include Modifier

The `include` modifier includes only those lines of output that contain the input string. In the output below, all lines containing the word “input” are included:

```
# show interface xe1 | include input
    input packets 80434552, bytes 2147483647, dropped 0, multicast packets 0
    input errors 0, length 0, overrun 0, CRC 0, frame 0, fifo 1, missed 0
```

You can specify a regular expression after the `include` keyword. This examples includes all lines with “input” or “output”:

```
#show interface xe0 | include (in|out)put
    input packets 597058, bytes 338081476, dropped 0, multicast packets 0
    input errors 0, length 0, overrun 0, CRC 0, frame 0, fifo 0, missed 0
    output packets 613147, bytes 126055987, dropped 0
    output errors 0, aborted 0, carrier 0, fifo 0, heartbeat 0, window 0
```

Exclude Modifier

The `exclude` modifier excludes all lines of output that contain the input string. In the following output example, all lines containing the word “input” are excluded:

```
# show interface xe1 | exclude input
Interface xe1
  Scope: both
  Hardware is Ethernet, address is 0004.75e6.5393
  index 3 metric 1 mtu 1500 <UP,BROADCAST,RUNNING,MULTICAST>
  VRF Binding: Not bound
  Administrative Group(s): None
  DSTE Bandwidth Constraint Mode is MAM
  inet6 fe80::204:75ff:fee6:5393/64
    output packets 4438, bytes 394940, dropped 0
    output errors 0, aborted 0, carrier 0, fifo 0, heartbeat 0, window 0
    collisions 0
```

You can specify a regular expression after the `exclude` keyword. This example excludes lines with “output” or “input”:

```
# show interface xe0 | exclude (in|out)put
Interface xe0
  Scope: both
  Hardware is Ethernet  Current HW addr: 001b.2139.6c4a
  Physical:001b.2139.6c4a  Logical:(not set)
  index 2 metric 1 mtu 1500 duplex-full arp ageing timeout 3000
  <UP,BROADCAST,RUNNING,MULTICAST>
  VRF Binding: Not bound
  Bandwidth 100m
  DHCP client is disabled.
  inet 10.1.2.173/24 broadcast 10.1.2.255
  VRRP Master of :  VRRP is not configured on this interface.
  inet6 fe80::21b:21ff:fe39:6c4a/64
    collisions 0
```

Redirect Modifier

The `redirect` modifier writes the output into a file. The output is not displayed.

```
# show cli history | redirect /var/frame.txt
```

The output redirection token (>) does the same thing:

```
# show cli history >/var/frame.txt
```

Last Modifier

The `last` modifier displays the output of last few number of lines (As per the user input). The last number ranges from 1 to 9999.

For example:

```
#show running-config | last 10
```

© 2026 IP Infusion Inc. Proprietary

String Parameters

The restrictions in Table 6 on page 18 apply for all string parameters used in OcNOS commands, unless some other restrictions are noted for a particular command.

Table 6: String parameter restrictions

Restriction	Description
Input length	1965 characters or less
Restricted special characters	"?", ",", ">", "|", and "=" The "|" character is allowed only for the `description` command in interface mode.

Command Modes

Commands are grouped into modes arranged in a hierarchy. Each mode has its own set of commands. Table P-7 lists the command modes common to all protocols.

Table 7: Common command modes

Name	Description
Executive mode	Also called *view* mode, this is the first mode to appear after you start the CLI. It is a base mode from where you can perform basic commands such as `show`, `exit`, `quit`, `help`, and `enable`.
Privileged executive mode	Also called *enable* mode, in this mode you can run additional basic commands such as `debug`, `write`, and `show`.
Configure mode	Also called *configure terminal* mode, in this mode you can run configuration commands and go into other modes such as interface, router, route map, key chain, and address family. Configure mode is single user. Only one user at a time can be in configure mode.
Interface mode	In this mode you can configure protocol-specific settings for a particular interface. Any setting you configure in this mode overrides a setting configured in router mode.
Router mode	This mode is used to configure router-specific settings for a protocol such as BGP or OSPF.

Command Mode Tree

The diagram below shows the common command mode hierarchy.

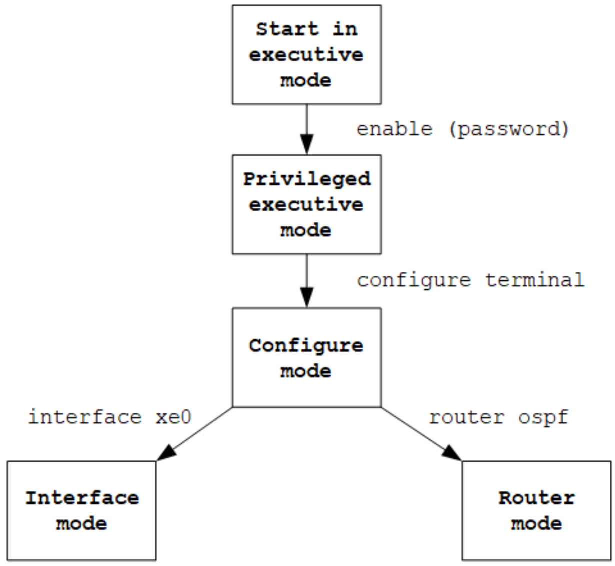


Figure 1: Common command modes

To change modes:

1. Enter privileged executive mode by entering `enable` in Executive mode.
2. Enter configure mode by entering `configure terminal` in Privileged Executive mode.

The example below shows moving from executive mode to privileged executive mode to configure mode and finally to router mode:

```
> enable mypassword
# configure terminal
Enter configuration commands, one per line.  End with CNTL/Z.
(config)# router ospf
(config-router)#
```

Note: Each protocol can have modes in addition to the common command modes. See the command reference for the respective protocol for details.

Transaction-based Command-line Interface

The OcNOS command line interface is transaction based:

- Any changes done in configure mode are stored in a separate *candidate* configuration that you can view with the `show transaction current` command.
- When a configuration is complete, apply the candidate configuration to the running configuration with the `commit` command.
- If a `commit` fails, no configuration is applied as the entire transaction is considered failed. You can continue to change the candidate configuration and then retry the `commit`.
- Discard the candidate configuration with the `abort transaction` command.
- Check the last aborted transaction with the `show transaction last-aborted` command.
- Multiple configurations cannot be removed with a single `commit`. You must remove each configuration followed by a `commit`.

Note: All commands MUST be executed only in the default CML shell (`cmlsh`). If you log in as root and start `imish`, then the system configurations will go out of sync. The `imish` shell is not supported and should not be started manually.

Quality of Service Configuration

CHAPTER 1 Introduction

This chapter provides an overview of Quality of Service (QoS) functionality and terminology.

QoS Functionality

QoS prioritizes certain traffic over others, improving network performance and bandwidth utilization. Without QoS, all traffic has the same chance of being delivered or dropped during congestion.

QoS is based on the DiffServ architecture, which classifies packets upon entry into the network. Classification information is carried in either the Layer-3 IP packet header or the Layer-2 frame. IP packet headers utilize 6-bits from the deprecated IP type of service (TOS) field, while Layer-2 802.1Q frames utilize a 2-byte Tag Control Information (TCI) field. All switches and routers accessing the Internet rely on class information to uniformly treat packets with the same class information and differentiate treatment for packets with different class information. Packets can be assigned class information, as follows:

- By end hosts or switches along a path, based on a configured policy
- Through sdtailed packet examination, which typically occurs closer to the network edge to prevent overloading core switches and routers.
- Use a combination of the above two techniques

Class information can be used by switches and routers along a path to limit the amount of allotted resources per traffic class. Per-hop behavior is an individual device's behaviour when handling traffic in the DiffServ architecture. An end-to-end QoS solution can be created if all devices along a path have consistent per-hop behavior.

Quality of Service (QoS) provides preferential treatment to specific traffic, possibly at the expense of other traffic. Without QoS, Qumran offers best-effort service to each packet, however, this may cause unpredictable network behavior. Implementing QoS in a network makes performance more predictable and bandwidth utilization more effective.

QoS design in Qumran complies with IETF-DiffServ and IEEE 802.1p standards. A typical QoS model deployment is based on the following elements:

- The packet received on customer edge port will be assigned to a QoS service. The service is assigned based on the packet header information.
- The QoS service defines the packet's internal QoS handling (i.e. traffic class/queue and drop precedence/color) and optionally the packet's external QoS marking, through either the IEEE 802.1p User Priority or the IP header DSCP field.
- Qumran provides end-to-end QoS behavior by providing consistent QoS treatment to the traffic within the network core based on packet's IEEE 802.1 or DSCP marking.
- Qumran can modify the assigned service of the packets if a packet stream exceeds the ingress configured rate by marking drop precedence and remarking packet's IEEE 802.1p or DSCP at the egress.
- Qumran incorporates the required QoS features to implement network-edge, as well as, network-core devices.
- Qumran provides flexible mechanisms to classify packets into different service levels.
- Service application mechanism is based on eight egress priority queues per port.
- The packet Priority fields can be remarked to reflect the QoS assignment on L2 and L3 networks.

  Note: Packet priority remarking on an MPLS network is not supported.

© 2026 IP Infusion Inc. Proprietary

Terminology

Following is a brief description of terms and concepts used to describe QoS.

ACL

Access control lists (ACLs) classify traffic with the same characteristics.

CoS Value

Class of Service (CoS) is a 3-bit value used to classify the priority of Layer-2 frames upon entry into a network. QoS classifies frames by assigning priority-indexed CoS values to them, and gives preference to higher-priority traffic.CoS values range from zero to seven, seven being the highest priority.

DSCP Value

DSCP Value Differentiated Services Code Point (DSCP) is a 6-bit value used to classify the priority of Layer-3 packets upon entry into a network. DSCP values range from 0 to 63, 63 being the highest priority, 0 being best-effort traffic.

Classification

Classification distinguishes one kind of traffic from another by examining the fields in the packet. Classification is enabled only if QoS is globally enabled on the switch. By default, QoS is globally disabled, thus, no classification occurs. Classification occurs on an ingress physical port. Classification can be based on QoS ACLs, or class maps and policy maps.

Policing

Policing can occur on ingress interfaces. Policer limits the bandwidth consumed by a traffic flow with the results given to the marker. The two types of policers:

- Individual: QoS applies the bandwidth limits specified in the policer, separately, to each matched traffic class. An individual policer is configured within a policy map.

Marking

Marking determines how to handle a packet when it is out of profile. It assesses the policer and the configuration data to determine the action required for the packet, and then handles the packet using one of the following methods:

- Let the packet through without modification
- Drop the packet

Marking can occur on ingress and egress interfaces.

Queuing

Queuing maps packets to a queue. Each egress port can accommodate up 8 queues, prioritized as 0 lowest and 7 highest. The tagged packet incoming priority can be mapped to one of the 8 queues obtained from the filtering

© 2026 IP Infusion Inc. Proprietary

mechanism result. The untagged packet priority is also obtained from the filtering mechanism result. After the packets are mapped to a queue, they are scheduled.

Scheduling

Scheduling forwards or conditions packets using one of the following methods:

- Strict Priority-Based (SP), in which any high-priority packets are first transmitted. Lower-priority packets are transmitted only when the higher-priority queues are empty. A problem may occur when too many lower-priority packets are not transmitted. Strict Priority will be operating on the remaining bandwidth available for the port
- WFQ (Weighted Fair Queuing) weight-based scheduling – In this scheduling, some weight based bandwidth is allocated to all queues. In this scheduling, egress traffic will be served based on the configured weight distribution.
- Combination of WFQ and SP, the Remaining Bandwidth will be scheduled in the strict order for the SP Queues. The Remaining Bandwidth will be scheduled in the WFQ mode for WFQ Queues.

Class Map

A class map names and isolates specific traffic from other traffic. The class map defines the criteria used to match against a specific traffic flow to classify it further. The criteria can include:

- Matching the access group defined by the ACL
- Matching a specific list of CoS, DSCP, Exp and etc.

If there is more than one type of traffic to be classified, another class map can be created under a different name. After a packet is matched against the class-map criteria, it is further classified using a policy map.

Policy Map

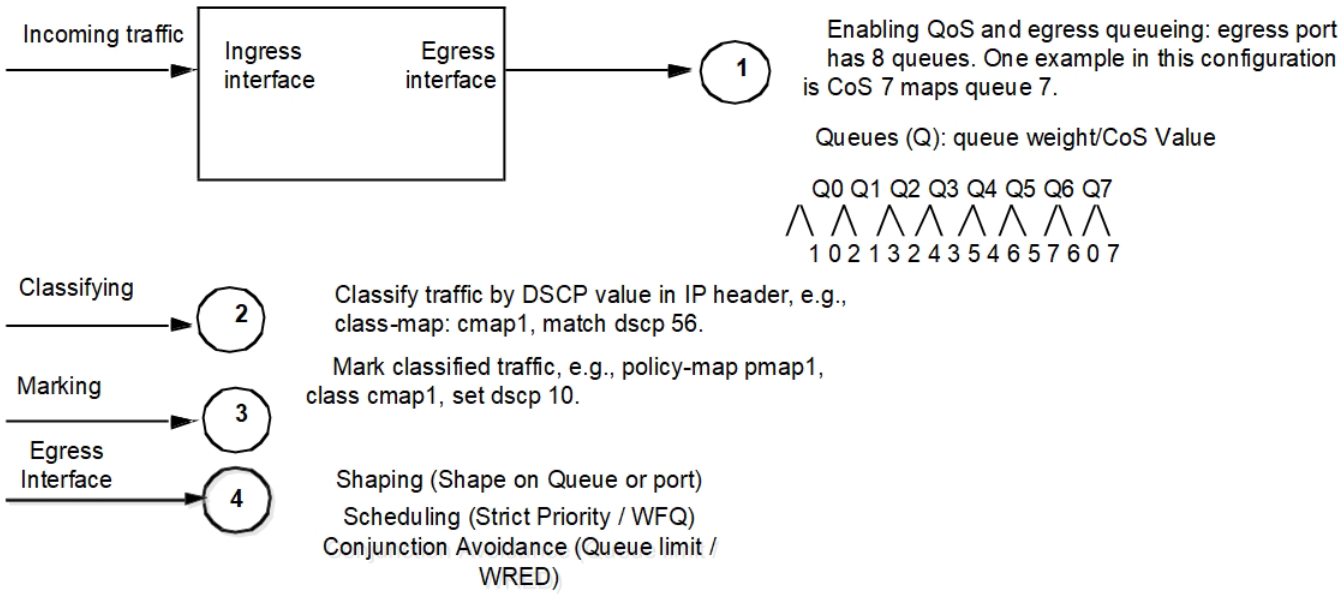
A policy map specifies on which traffic class to act. This can be implemented as follows:

- Set a specific CoS or DSCP value in the traffic class.
- Specify the traffic bandwidth limitations for each matched traffic class (policer) and the action to take (marking) when the traffic is out of profile.

Policy maps have the following attributes:

- A policy map can contain Maximum 256 class-map per policy-map, each with different match criteria and policers.
- A separate policy-map class can exist for each type of traffic received through an interface.
- There can be only one policy map per interface per direction. The same policy map can be applied to multiple interfaces and directions.
- Before a policy map can be effective, it must be attached to an interface.

© 2026 IP Infusion Inc. Proprietary



QoS model

This section explains the implementation and configuration details of QoS:

Traffic types

Data — Packets can be network-to-network traffic or traffic from CPU. Network-to-network packets are considered data traffic and QoS parameters can be applied on data traffic.

Control — Packets to and from the CPU are considered as control traffic. Incoming control traffic is prioritized based on ingress map and are sent to their own designated CPU queues. Each CPU queue has a fixed rate limit to guard the CPU. Outgoing control traffic will always be sent on highest priority queue (Q7) on the data port.

Setting packet header QoS fields

The device supports modifying the packets header IEEE 802.1p user priority or IP-DSCP.

Packet QoS Attributes

Every data packet is assigned a set of QoS attributes that can be modified in several stages of the ingress pipeline engine.

The ingress pipeline engine also contains a QoS Remark option for L3 traffic that can modify the initial QoS attributes of the packet.

Color and Drop Precedence relation: Green has the lowest drop precedence, Yellow has a higher drop precedence, and Red has the highest drop precedence. See Table 1-1.

© 2026 IP Infusion Inc. Proprietary

Table 1-1: Packet QoS attributes

QoS Parameter	Description
TC (Traffic Class or queue)	This is the priority level assigned to the packet. When the transmission engine queries the packet, it uses this field to select the appropriate priority queue
DP (Drop Precedence or color)	The transmission engine uses this field for congestion resolution. Packets with higher drop precedence are more likely to be discarded in the event of congestion. By default, packets with red color will be dropped by a Qumran even if there is no congestion. Disabling red packet drop is configurable per device. In case of L2 packet, DEI 0 will be marked to color green and DEI 1 will be marked to color yellow. In case of L3 packet, AFx1 will be marked as green while both AFx2 and AFx3 will be marked as yellow (where x=1,2,3).

© 2026 IP Infusion Inc. Proprietary

CHAPTER 2 Configuring a QoS Policy-map

The following section explains the configuration of basic infrastructure to apply QoS treatment on the ingress interface.

Ingress QoS treatment can be achieved by two types of configuration.

- Policy-map configuration
- Profile mapping configuration

This section explains policy-map configuration method. QoS feature must be enabled to configure policy-maps.

This infrastructure contains two entities - class-map and policy-map. Class-map holds the match criteria and class-maps can be bound to policy-map to configure QoS treatment for the matching traffic.

Creating a QoS class-map

A Class-map contains the matching criteria for the traffic. Class-maps with no match criteria will match all the data traffic.

Class-maps support the following match criteria: vlan, inner vlan, cos, inner cos, dscp, precedence, layer4 tcp/udp port, exp, Ether-type, or ACL.

Class-map can be of two types, match-any or match-all type. By default class-map is of match-all type. For match-all class-maps, only the traffic that qualifies all match criteria configured in class-map will take QoS action. For match-any class-maps, traffic qualifying any one of the configured match criteria will take QoS action.

Class-maps can be created once the QoS feature is enabled.

Use the following command to create a class-map:

```
class-map (type qos|) (match-any|match-all|) NAME
```

This command will create a class-map entity that can be configured with one or more match criteria.

Class-map can be of two types:

1. Empty class-map: Class-map with no matching criteria is called empty class. Empty class will classify all the traffic coming on the port on which the policy containing empty class will be applied.
2. Non-empty class-map: Class-map with matching criteria is called non-empty class. It will classify the traffic according to the given matching criteria.

User can always add/delete/modify the matching criteria of the class-map. User is not allowed to make a non-empty class as empty class if the class is attached to a policy-map. User has to remove the class from policy-map to modify the matching criteria. User is allowed to add matching criteria to an empty class attached to a policy-map.

Use the following commands to add match criteria:

```
match vlan WORD
match vlan inner WORD
match cos WORD
match cos inner WORD
match dscp [WORD | af11 | af12 | af13 | af21 | af22 | af23 | af31 | af32 | af33 | af41 |
af42 | af43 | cs1 | cs2 | cs3 | cs4 | cs5 | cs6 | cs7 | default | ef]
match precedence [WORD | critical | flash | flashoverride | immediate | internet |
network | priority | routine]
match mpls experimental topmost WORD
match ip rtp WORD
match ethertype WORD
match layer4 (tcp|udp|any) (source-port|destination-port) WORD
```

© 2026 IP Infusion Inc. Proprietary

```
match access-group NAME
```

An example of creating a class-map with match VLAN configuration is shown below:

```
(config)#qos enable
(config)#class-map n1-class-10
(config-cmap-qos)#match vlan 10
(config-cmap-qos)#exit
```

Notice that data traffic with an outer VLAN value of 10 will be matched for QoS treatment.

Creating a QoS policy-map

Policy-maps can be configured with multiple class-maps with each class-map configured with policing or other supported actions.

Use the following command to create a policy-map:

```
policy-map (type qos|) NAME
```

This command will create a policy-map entity in which one or more class-map entities can be bound.

Binding a class-map to a policy-map

In order to apply QoS treatment to traffic that matches the criteria configured in a class-map, the class-map must be bound to the policy-map entity, then QoS treatment actions can be configured on that node.

Use the following command to bind a class-map to a policy-map:

```
class (type qos|) NAME
```

This command will create a class node for class-map unique to the policy-map. This class node can then be configured with QoS treatment actions.

There are two types of actions, `police` and set.

- Police actions can be used to control the rate of traffic flow and is explained in the next section.
- Set actions can be used to set internal priority or packet priority.

Use the following commands to add set actions:

```
set (dscp (<0-63>| af11| af12| af13| af21| af22| af23| af31| af32| af33| af41| af42|
af43| cs1| cs2| cs3| cs4| cs5| cs6| cs7| default| ef ))
set cos <0-7>
set queue <0-7>
```

An example of creating a policy-map and binding a class-map to it with the `set` action is shown below:

```
(config)#policy-map n1-police-10
(config-pmap-qos)#class n1-class-10
(config-pmap-c-qos)#set cos 3
(config-pmap-c-qos)#exit
(config-pmap-qos)#exit
```

In case of multiple class-maps, implicit priority of class-maps will be decided based on the following the type of class-map as shown:

match-all > match-any > default class-map

If all class-maps are of type match-all, the class-map with the higher number match criteria will have higher priority. If the class-maps have the same number of match criteria, then the behavior can be unpredictable if traffic matches both class-map criteria. In such cases, class-maps can be explicitly prioritized using the priority command.

© 2026 IP Infusion Inc. Proprietary

priority <1-1000>

An example of configuring priority per class-map is shown below:

```
(config)#policy-map p1
(config-pmap-qos)#class c1
(config-pmap-c-qos)#priority 2
```

A higher the value means higher priority, and user defined priority is always higher prioritized over implicit priority.

If the match criteria is ACL, different hardware resources will be used. Therefore, configuring match ACL class-map and other non-ACL match classes in a single policy is not allowed because if a policy-map with two class-maps (one with ACL match criteria and one with some other match criteria) are allowed (configured in different hardware resource groups), traffic matching both match criteria can take actions of both class-maps if the actions are not conflicting but since statistics are common and can be read once (hardware limitation), statistics will be displayed for one matching criteria only.

Assumptions and limitations

DSCP and precedence matches are mutually exclusive and can't be part of same class-map.

Each rule of an ACL is matched independently, so ACL matches are allowed to be configured only on match-any type class-map. If the configured matches on class-maps bound to and interface via policy-map are not mutually exclusive, there are chances of multiple class-maps matching a single stream of traffic. In this case, implicit priority of the class-map based on number of rules decides which class-map the traffic will hit. When implicit priority is the same for class-maps, their behavior will be random, and the user is expected to configure class priority in such a case.

Binding a QoS policy-map

Policy-map configuration is effective only when it is bound to a supported ingress interface. Use the following command to bind a policy-map to an interface:

```
service-policy type qos input NAME
```

Notice that `NAME` represents the name of the qos policy-map.

Note: A QoS TCAM group must be enabled before binding a policy-map to an interface. See the `hardware-profile filter` and `show hardware-profile filters` commands in the *System Management Guide* for more about hardware filter groups.

An example of binding a policy-map to an interface is shown below:

```
(config)#hardware-profile filter qos-ext enable
(config)#interface xe1
(config-if)#service-policy type qos input n1-police-10
(config-if)#exit
```

In this example, traffic with VLAN value 10 received on interface `xe1` is matched by `class-map` configuration, and traffic will be set with internal and packet priority (cos) 3 (as shown in the previous example).

Note: For an ACL match configuration, ingress-l2-ext/ingress-ipv4-ext/ingress-ipv6 TCAM group must be enabled based on ACL type.

When the class-map is configured with match ACL, only the `police` action is supported – `set` actions are not supported.

CHAPTER 3 Traffic Policing

Traffic policing can be achieved by using a policy-map based method. Policy-map based configurations allow the flexibility to police the traffic per-port and a set of other matching criteria:

- VLAN (outer vlan and inner vlan)
- CoS (outer cos and inner cos)
- DSCP
- Topmost EXP
- Ether-typ
- Precedence
- TCP or UDP port
- ACL

Note: Qumran supports rate limiting of ingress traffic. Rate limiting egress traffic is not supported.

Applying Traffic Policing Parameters Using a policy-map

Policy-map based traffic policing can be achieved by binding policy-map on the interface in ingress direction. A policy-map is associated with two sections. One is class-map which will have match criteria configured. Other is police configuration to apply traffic policing on the matching traffic on the policy-map bound port in ingress direction (see Chapter 2, *Configuring a QoS Policy-map*).

Traffic policing determines the rate of ingress traffic that is allowed per port (traffic that matches the configuration in the class-map).

Note: Qumran supports two types of policing:

- Single rate three color traffic rate limiting (RFC 2697).
- Two rate three color traffic rate limiting (RFC 4115).

Single rate or two rate operations are in compliance with the RFCs mentioned.

Note: Packets marked with color red are dropped by default in Qumran devices. This default behavior can be modified with global command to disable red packet drop. However, traffic policing and storm control will not work if red packet drop is disabled.

For more information about color, refer to the Packet QoS Attributes section.

Configuration Considerations

- Policy map based rate limiting is supported only for ingress traffic.
- Only one policy-map of same type can be bound on an interface.
- One policy-map can have up to 256 class-maps.
- CIR and EIR configuration should be in same format. Example, if one of them is configured in percentage, other also should be configured in percentage.
- The minimum configurable rate is 22 kilobits per second.
- The minimum supported burst size is 1 kilobyte, while the maximum supported burst size is 4161 kilobytes.

Configuring Traffic Policing

The following section shows how to configure policing on an interface. See Chapter 2, *Configuring a QoS Policy-map* for configuring policy-maps.

Note: Policer action must be configured on the class node to achieve traffic policing for matching traffic.

Use the following command to configure a policer for Qumran MX:

```
police (colour-blind | colour-aware |) (cir) (<1-720000000> (kbps|mbps|gbps) | percent
<1-100>) ((eir (<1-720000000> (kbps|mbps|gbps) | percent <1-100>)|) ((bc) <1-4161>
(kbytes|mbytes|ms|us)|) ((be) <1-4161> (kbytes|mbytes|ms|us)|))
```

Use the following command to configure a policer for Qumran AX:

```
police (colour-blind | colour-aware |) (cir) (<1-500000000> (kbps|mbps|gbps) | percent
<1-100>) ((eir (<1-500000000> (kbps|mbps|gbps) | percent <1-100>)|) ((bc) <1-4161>
(kbytes|mbytes|ms|us)|) ((be) <1-4161> (kbytes|mbytes|ms|us)|))
```

- For Qumran MX, the configurable rate range is 22 kbps to 720 gbps.
- For Qumran AX, the configurable rate range is 22 kbps to 500 gbps.

An example of creating a policy-map and binding a class-map to it with police action is shown below:

```
(config)#qos enable
(config)#class-map n1-class-10
(config-cmap-qos)#match vlan 10
(config-cmap-qos)#exit
(config)#policy-map n1-police-10
(config-pmap-qos)#class n1-class-10
(config-pmap-c-qos)#police cir 10 mbps eir 20 mbps
(config-pmap-c-qos)#exit
(config-pmap-qos)#exit
(config)#hardware-profile filter qos-policer enable
(config)#interface xe1
(config-if)#service-policy type qos input n1-police-10
(config-if)#exit
```

In the following example, traffic with VLAN ID 10 received on interface `xe1` will be policed to a total of 30 mbps with 10 mbps of traffic being marked green, and 20 mbps of traffic marked yellow, any remaining traffic will be dropped at ingress.

Example configuration for color aware police:

```
(config-pmap-c-qos)#police colour-aware cir 10 mbps eir 20 mbps
```

With this configuration, if traffic with vlan ID 10 (with CFI bit set) is received on interface `xe1` it is “policed” to a total of 20 mbps only because the traffic will be treated as yellow and will be subjected only to the EIR bucket.

Displaying Rate Limiting Policies

Use the following commands to verify the configuration and statistics.

- `show policy-map` – This command displays the configuration of policy map.
- `show policy-map interface INTERFACE-NAME` – This command displays the policy-map details on the interface along with statistics of how many packets and bytes matches and how many packets and bytes are dropped due to policer.
- `show policy-map statistics type qos` – This command displays the statistics of matched packets and bytes and dropped packets and bytes per class-map in table format.

Note: Packets dropped by the policer are counted in policy-map drops, as well as in queue red drops because the hardware doesn't support policer action to directly drop the packet. Packets that need to be dropped are marked red and are dropped at the queue.

Use the following command to obtain QoS statistics:

```
qos statistics
```

Use the following command to clear QoS statistics.

```
clear qos statistics
```

Drop counters verification

Drop counters with drop reason can be verified globally using the following command:

```
#show hardware-discard-counters
+-----------------------------------+---------+------------+------------+
| Registers                                   | Core 0      | Core 1      |
+-----------------------------------+---------+------------+------------+
IQM_QUEUE_ENQ_DISCARDED_PACKET_COUNTER        1596100
   Reason: DP_LEVEL_STATUS                    Y
EGQ_PQP_DISCARD_UNICAST_PACKET_COUNTER        59807
   Reason: SRC_EQUAL_DEST_INT                 Y
```

CHAPTER 4 LAG Egress Shaping or Policing on Sub-Interfaces

The LAG Egress Shaping or Policing on Sub-Interfaces feature introduces Hierarchical Quality of Service (HQoS) enhancements to enable unified bandwidth management for Link Aggregation Group (LAG) and Multi-Chassis LAG (MLAG) interfaces.

In earlier releases, LAG shaping was applied per member port, causing an overall bandwidth reduction when one or more LAG members failed. The new Unified Traffic Management (TM) model resolves this limitation by enabling a single, shared shaper across all active LAG members, ensuring consistent throughput and QoS behavior.

This enhancement provides two programming models:

- Multi-Unit TM per Member Model: Applied per member and per service (sub-interface) for multi-unit systems.
- Single-Unit Unified TM Model: Applied per LAG or per LAG-based service for single-unit systems.

Note: Both models provide enhanced control over bandwidth allocation, congestion avoidance, and policy enforcement for LAG and sub-interface–based traffic.

Feature Characteristics

- Provides egress shaping and policing on LAG interfaces and sub-interfaces.
- Supports HQoS for flexible bandwidth control and service differentiation.
- Offers two QoS programming models:
  - Per-Member Model: Individual TM instance per member or per service.
  - Unified Model: Single TM instance applied across the entire LAG or per service.
- Allows unified bandwidth control across all active LAG members to maintain consistent throughput.
- Supports QoS operations, including shaping, congestion avoidance (WRED), tail-drop, and header compensation.
- Enables policy-map binding and modification for both default and user-defined QoS profiles.
- Supports per-service (sub-interface) QoS configuration, allowing granular bandwidth control at VLAN or service levels.
- Provides dynamic event handling for LAG membership changes — automatically reprograms QoS mappings when members are added or removed.
- Offers counters and debug visibility for interfaces, policy maps, and scheduling elements.
- Compatible with both static and dynamic LAG configurations.
- Integrated with default and user-defined policy maps for traffic management.
- Supports hierarchical scheduling, allowing bandwidth enforcement at multiple levels (LAG and service).
- Maintains consistent QoS behavior across LAG and MLAG interfaces.

Benefits

- **Unified Bandwidth Management:** Prevents bandwidth degradation when a member link fails by maintaining a shared shaping policy.
- **Enhanced QoS Consistency:** Ensures uniform treatment of traffic across aggregated links.
- **Flexible Configuration Models:** Offers both per-member and unified programming options.
- **Dynamic Adaptability:** Handles membership changes in real time without affecting service continuity.

- **Efficient Resource Utilization:** Optimizes use of traffic management and scheduling elements.
- **Simplified QoS Administration:** Centralized configuration per LAG reduces operational complexity.
- **Scalable Design:** Supports thousands of sub-interfaces and policy instances with predictable performance.

Limitations

- Unified bandwidth model cannot be used when LAG spans multiple switching units.
- Swapping between programming models is not supported when:
- The LAG interface already has associated member ports.
- A policy map is already bound to the LAG interface.
- MLAG is supported only in active/standby mode; active/active unified shaping is not supported.
- Hash-based load balancing is used as the traffic distribution mechanism; unified TM does not alter hashing behavior.
- In this release, only the ‘Match Queue’ classification option is supported for both programming models.
- Shaper limit is not supported on unified shaper LAG interfaces when using default policy maps due to the inherent hierarchy of default queueing.
- Default policies such as ‘default-out-policy’ and ‘subif-default-out-policy’ do not apply shaping.
- To apply shaping correctly, the user must use a custom (user-defined) policy map on unified shaper LAG interfaces or sub-interfaces.
- Unified TM scheme can only be enabled before member ports are added.
- Migration between per-member and unified TM models is not supported once the LAG is active.
- Unified TM model is supported only on systems where the traffic manager operates in a single, unified instance.
- Resource scaling depends on available traffic management and scheduling elements; shaping precision may vary slightly at high rates.

Configuration Considerations

- Unified scheme can only be enabled before adding member ports.
- The LAG interface must be associated with a default policy-map before enabling unified scheme.

Configurations

When traffic is forwarded over a port-channel (single or multi-chassis), rate limits must be applied at the correct logical level (LAG interface or sub-interface) to ensure consistent, predictable bandwidth control across bundled links.

Topology

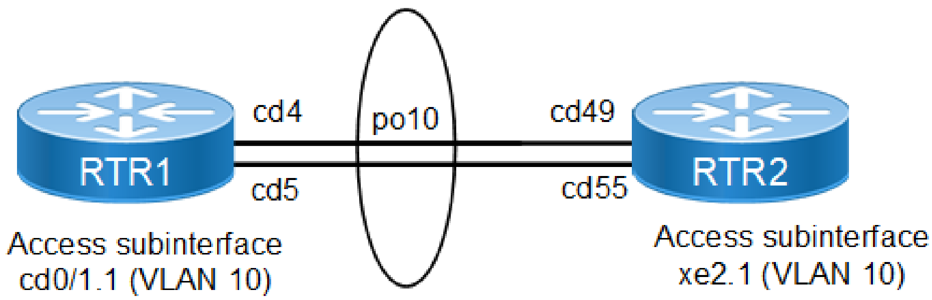
The topology consists of two network endpoints connected through a pair of intermediate network nodes. Traffic enters the topology through interface cd0/1 on the first node. This interface is configured as a sub-interface with VLAN encapsulation and mapped into a bridge domain.

From there, traffic is forwarded across a LAG formed using interfaces cd4 and cd5. These two physical links operate as a single logical port-channel, enabling the application of egress shaping or policing on the associated sub-interface.

On the second node, the corresponding LAG is created using interfaces cd49 and cd55. The aggregated traffic received over this port-channel is then forwarded toward the far end of the network through interface xe2.

This topology is used to validate that egress shaping or policing applied on a port-channel sub-interface is consistently enforced across all member links of the LAG, ensuring uniform rate control across the aggregated path.

Figure 4-1: Egress Shaping or Policing on Subinterface



Perform the following steps to configure egress shaping or policing on LAG sub-interfaces:

1. Configure the Port-Channel interface with a LAG shaper to ensure that shaping or policing is applied per LAG instead of per physical member interface.

```
(config)# interface po10
(config-if)# switchport
(config-if)# load-interval 30
(config-if)# lag-shaper unified unit 1
(config-if)# commit
```

Note: Based on the interface availability configure the lag-shaper unified unit zero or unit one.

2. Attach physical interfaces to the Port-Channel so the LAG operates as a single logical egress interface.

```
(config)# interface cd4
(config-if)# channel-group 10 mode active
(config-if)# commit
(config-if)# exit

(config)# interface cd5
(config-if)# channel-group 10 mode active
(config-if)# commit
```

3. Create the access sub-interface with VLAN encapsulation to identify customer/service traffic before it enters the bridge domain.

```
(config)# interface cd0/1.1 switchport
(config-if)# encapsulation dot1q 10
(config-if)# load-interval 30
(config-if)# commit
```

4. Apply port shaping at the Port-Channel sub-interface to ensure per-service bandwidth control over the LAG uplink.

```
(config)# interface po10.1 switchport
(config-if)# encapsulation dot1q 10
(config-if)# load-interval 30
(config-if)# shape rate 4 gbps
(config-if)# commit
```

5. Create queue classifications and shaping policies to enforce rate limits per queue on the sub-interface.

```
(config)# qos enable
(config)# qos statistics
(config)# commit
```

```
(config)# class-map type queuing C3
(config-cmap-que)# match queue 0
(config-cmap-que)# commit
(config-cmap-que)# exit

(config)# policy-map type queuing P3
(config-pmap-que)# class type queuing C3
(config-pmap-c-que)# shape 3 gbps
(config-pmap-c-que)# exit
(config-pmap-que)# commit
```

6. Attach the policy-map with queue shaping to the port-channel sub-interface to enforce egress rate limits on LAG traffic.

```
(config)# interface po10.1
(config-if)# service-policy type queuing output P3
(config-if)# commit
```

7. Map the access and uplink sub-interfaces into the bridge domain to enable L2 forwarding with shaping applied.

```
(config)# bridge-domain 5
(config-bridge-domain)# interface cd0/1.1
(config-bridge-domain)# interface po10.1
(config-bridge-domain)# commit
```

8. Repeat LAG and Sub-Interface Configuration on the Peer side:
   - The corresponding port-channel
   - Member interfaces
   - VLAN sub-interfaces
   - Bridge-domain mappings

Validation

```
show etherchannel summary
  Aggregator po10 100010
  Aggregator Type: Layer2
  Admin Key: 0010 - Oper Key 0010
     Link: cd4 (5023) sync: 1
     Link: cd5 (5024) sync: 1

show policy-map interface po10.1

Interface po10.1

Type Queuing policy-map : P3

Service-policy (queuing) output: P3
Interface Bandwidth 800000000 kbps
Port Shaper 4 gbps
-----------------------------------
Class-map (queuing): C3
 shape 3 gbps
 wfq-queue weight 1
 queue-limit 3749888 bytes/10 ms (default)
 match queue 0
```

```
      Output
         Total        : 8357663 packets, 12536494500 bytes
         Green        : 8357663 packets, 12536494500 bytes
         Yellow       : 0 packets, 0 bytes
      Dropped
         Total        : 2744779 packets, 4117168500 bytes
         Green        : 2744779 packets, 4117168500 bytes
         Yellow       : 0 packets, 0 bytes
         Red          : 0 packets, 0 bytes

Class-map (queuing): class-default-q
 shape 4000000 kbps (inherited)
 wfq-queue weight 1
 queue-limit 500000000 bytes/1000 ms (default)
 match queue 1
 match queue 2
 match queue 3
 match queue 4
 match queue 5
 match queue 6
 match queue 7

show interface po10.1 counters queue-stats
E - Egress, I - Ingress, Q-Size is in bytes
+--------------------+-------------------+-----------------+-------------------+-------
----------+-------------------+
|   Queue/Class-map  |      Q-Size       |     Tx pkts     |     Tx bytes      |  Dropped
pkts   |   Dropped bytes   |
+--------------------+-------------------+-----------------+-------------------+-------
----------+-------------------+
q0                 (E) 3749888       37957242        56935863000       12488411
18732616500
q1                  (E) 500000000      0               0                 0
0
q2                  (E) 500000000      0               0                 0
0
q3                  (E) 500000000      0               0                 0
0
q4                  (E) 500000000      0               0                 0
0
q5                  (E) 500000000      0               0                 0
0
q6                  (E) 500000000      0               0                 0
0
q7                  (E) 500000000      0               0                 0
0

show interface counters rate mbps
+------------------+-------------+-------------+--------------+-------------+
|     Interface    |    Rx mbps  |    Rx pps   |    Tx mbps   |    Tx pps   |
+------------------+-------------+-------------+--------------+-------------+
 cd0/1               3993.66       332804       0.00           0
 cd0/1.1             3880.44       323369       0.00           0
```

```
cd4             0.00          0          1502.30        125191
cd5             0.00          0          1502.29        125191
po10            0.00          0          3004.59        250382
po10.1          0.00          0          3004.59        250382
```

Implementation Examples

Unified Shaper for Single LAG Interface:

Applying a single shaping policy across all active member links of a LAG interface to ensure consistent bandwidth utilization, even during link failures.

A operator aggregates four 25G links into a single LAG (Port-channel10) to provide 100G aggregate bandwidth toward an access switch.To prevent bandwidth degradation when a member link goes down, a unified egress shaper is configured at the LAG interface level.

Per-Service (Sub-Interface) Shaping on LAG:

Implementing HQoS where different services (sub-interfaces) under the same LAG interface have individual shaping or policing profiles.

A service provider runs multiple VLAN-based services (For example: business internet and VoIP) over the same LAG. Each VLAN is mapped to a sub-interface, and each sub-interface requires a distinct egress shaping rate.

Commands

The following command is introduced as part of the OcNOS version 7.0.0.

lag-shaper unified

Use this command to enables the unified traffic management (TM) scheme on a LAG or MLAG interface. When enabled, bandwidth shaping is applied uniformly across all member ports of the LAG, preventing per-member bandwidth limitations and ensuring consistent bandwidth allocation.

On Qumran2 series platforms with dual switching units, the optional unit parameter specifies the switching unit to which the LAG is associated. On single-unit platforms, this parameter is ignored, and the LAG is automatically associated with the available unit.

For unit-to-port mapping details, use the `show interface hw-mapping` command.

Use the no parameter of this command to disable the unified traffic management (TM) scheme on a LAG or MLAG interface.

Command Syntax

```
lag-shaper unified (unit <0-1>)
```

Parameters

`unit`	Specifies the switching unit to which the LAG belongs (applicable only on multi-unit platforms).
`<0-1>`	0 specifies the first unit 1 specifies the second unit

Default

unit 0

Command Mode

Interface configuration mode.

Applicability

Introduced in OcNOS version 7.0.0.

Example

The following examples show to enable the unified TM scheme on a LAG interface, with and without specifying the switching unit.

```
# configure terminal
(config)# interface sa1
(config-if)# lag-shaper unified

(config)# interface sa1
(config-if)# lag-shaper unified unit 1

(config)# interface sa1
(config-if)# lag-shaper unified unit 0
```

Glossary

The following provides definitions for key terms used throughout this document.

Key Terms/Acronym	Description
Link Aggregation Group (LAG)	A logical interface formed by bundling multiple physical Ethernet links to provide increased bandwidth, load balancing, and redundancy.
Multi-Chassis Link Aggregation (MLAG)	An extension of LAG that allows physical links connected to different chassis to appear as a single logical LAG, providing high availability and redundancy.
Egress Traffic	Traffic that is transmitted out of an interface toward the network or downstream devices.
Egress Shaping	A traffic management mechanism that smooths outbound traffic by buffering packets and transmitting them at a configured rate to conform to bandwidth limits.
Egress Policing	A traffic control mechanism that enforces a maximum transmission rate on outbound traffic by dropping or marking packets that exceed the configured rate.
Traffic Management (TM)	A set of mechanisms, including shaping and policing, used to control bandwidth usage, prioritize traffic, and ensure predictable network performance.

CHAPTER 5 Rate Limiting BUM Traffic

To prevent the CPU from getting flooded with Broadcast, Unknown Unicast, and Multicast (BUM) traffic, rate limiting can be applied at the ingress interface. This configuration is called “storm control” and is independent of the QoS feature, and can be configured directly on the interface in the ingress direction.

Configure per Port Rate Limiting for BUM Traffic on Physical Interface

BUM rate limiting can be configured on the interface via following command:

```
storm-control (errdisable|((broadcast|multicast|dlf) (level LEVEL | <0-4294967294>
(kbps|mbps|gbps) burst-size <1-33292>)))
```

The `broadcast` option is for broadcast traffic, `multicast` option is for multicast traffic and `dlf` (Destination Lookup Failure) option is for unknown unicast traffic.

There are two ways to configure the rate limiting information. One is in percentage and the other is in absolute values. Use the level option to configure in percentage format where the link speed is used for rate calculation. Percentage value can be configured with up to four fractional digits in order to support kbps level rate limiting on 100 gbps ports, and the Absolute configuration option uses the input rate in kbps, mbps, or gbps.

Burst-size is an optional parameter, if storm control is not configured with burst-size, default value is calculated as 5ms of configured rate to limit the traffic.

An example of BUM rate limiting is shown below:

```
(config)#interface xe1
(config-if)#switchport
(config-if)#storm-control dlf 2 mbps
(config-if)#exit
```

In the configuration above, unknown unicast traffic received on `xe1` will be rate limited to 2 mbps. Burst-size is calculated as 10 Kbps.

Note:

- Packets marked with color red are dropped by default by Qumran devices. This default behavior can be modified with global command to disable red packet drop. However, traffic policing and storm control will not work if red packet drop is disabled.
- Burst-range in Qumran: 1-33292 Kbps.
- Storm discard notification is sent in case of packet discards but not based on configuration.

Storm Detection Time

Rate-limit setting should take place immediately after detecting storm. With existing burst size, rate limit is happening after few seconds, if burst size should be changed to a lower value so that rate-limiting happens immediately on the switch. Also, user have a provision to change burst size through CLI.

An example of BUM rate limiting with Burst-size configuration is shown below:

```
(config)#interface xe1
(config-if)#switchport
(config-if)#storm-control broadcast 600 kbps burst-size 15
(config-if)#exit
```

In the configuration above, broadcast traffic received on xe1 will be rate limited to 600 kbps and Burst-size is configured as 15 Kbps. If traffic sent is 700 kbps. Time taken to detect the storm should be 0.15 Seconds.

Displaying BUM rate limit information

Use the following command to verify BUM rate limit configuration:

```
show storm-control (INTERFACE-NAME|)
```

Note: Discard counters for BUM rate limiting is not supported on Qumran1.

```
#show storm-control
*The hardware applicable value is displayed
Port     BcastLevel(burst)       McastLevel(burst)   DlfLevel(burst)       Discards
ce5      100.000 kbps(0 Kbps)   100.0000%(0 Kbps)    1000 mbps(10 Kbps)    0
ce9/1    100.000 kbps(0 Kbps)   100.0000%(0 Kbps)    1000 mbps(10 Kbps)    0
```

Displaying BUM discards on Qumran2

Use the following command to verify BUM discards:

```
show storm-control (INTERFACE-NAME|)discards
```

Note: Discards for BUM traffic supports on Qumran2 only.

```
#show storm-control discards
Port     BcastLevel(burst)       McastLevel(burst)   DlfLevel(burst)       Discards
ce5      0                       0                   0                     0
ce9/1    0                       0                   0                     0
ce0      0                       0                   0                     0
```

Configure per Port Rate Limiting for BUM Traffic on Sub interface

BUM rate limiting can be configured on the sub interface through following command:

The l2-bc option is for broadcast traffic `l2-mc` option is for multicast traffic and `l2-unknown` option is for unknown unicast traffic.

To configure the rate limiting information, the Absolute configuration option uses the input rate in kbps, mbps, or gbps.

An example of BUM rate limiting:

```
(config)# class-map type qos match-all c1
(config-cmap-qos)# match traffic-type l2-mc
(config-cmap-qos)#exit
(config)#policy-map type qos p1
(config-pmap-qos)# class type qos c4
(config-pmap-c-qos)# police cir 100 kbps
(config-pmap-c-qos)# exit
(config-pmap-qos)#exit
(config)#class-map type qos match-all c2
(config-cmap-qos)# match traffic-type l2-bc
(config-pmap-c-qos)# exit
(config-cmap-qos)#exit
(config)#policy-map type qos p1
(config-pmap-qos)#class type qos c5
(config-pmap-c-qos)# police cir 100 kbps
(config-pmap-c-qos)# exit
(config-pmap-qos)#exit
(config)# class-map type qos match-all c3
```

```
(config-cmap-qos)# match traffic-type l2-unknown
(config-cmap-qos)#exit
(config)#policy-map type qos p1
(config-pmap-qos)#class type qos c3
(config-pmap-c-qos)# police cir 100 kbps
(config-pmap-c-qos)# exit
(config-pmap-qos)#exit
(config)#interface xe1.200 switchport
(config-if)#service-policy type qos input p1
(config-if)#exit
```

In the configuration above, unknown unicast, Broadcast and multicast traffic received on `xe1` will be rate limited to 100 mbps and total BUM is rate limited to 300mbps.

Note: Packets marked with color red are dropped by default by Qumran devices. This default behavior can be modified with global command to disable red packet drop. However, traffic policing and storm control will notwork if red packet drop is disabled.

Displaying BUM Rate Limit Information in the Interface Counters

Use the following command to verify BUM rate limit configuration:

Sending unknown, multicast and board cast traffic with rate 200mbps each and is rate limited to 100mbps, total of 300mbps.

Note: Discard counters for BUM rate limiting is not supported on Qumran.

```
#show interface counters rate mbps
+-------------------+--------------+-------------+--------------+-------------+
| Interface | Rx mbps | Rx pps | Tx mbps | Tx pps |
+-------------------+--------------+-------------+--------------+-------------+
xe2 0.00 5 302.00 37162
xe1 606.45 75806 0.00 0
xe1.200 600.45 49010 0.00 0
```

CHAPTER 6 Ingress Traffic Processing

Qumran can process packets based on the priority information of the packet. During the packet processing, there are several opportunities to influence the processing by the following configurations:

1. Derive internal priority and drop precedence from the packets IEEE 802.1p (PCP) value and drop eligibility indicator (DEI) value. The Priority Code Point (PCP) is a 3-bit field within an 802.1Q tagged frame that is used to convey the internal priority of the frame and DEI is a single bit that is used to convey the drop precedence of the frame.

   Note: The PCP field was formerly called 802.1p.

2. Derive internal priority and drop precedence from the packets DSCP value.
3. Force modification of internal priority and drop precedence using user defined ingress mapping profile.
4. Force the internal priority value based on classification configured using policy maps. This is used for setting a specific priority for L2, L3 or L4 traffic flow.
5. Packet priority and drop precedence will be based on PCP and DEI values for the traffic received on switch ports. Similarly, packet priority and drop precedence will be based on DSCP value for the IPv4 traffic received router ports. For MPLS traffic on label switched interface, packet priority is based on topmost label's EXP value.
6. When trust level on switch ports is modified to trust DSCP values, packet priority and drop precedence for IPv4 traffic will be based on DSCP value.
7. Assign internal priority to untagged traffic on switch port with port level untagged priority configuration.

Mapping Inbound Packet Priorities to Internal Priorities

Priority and drop precedence information is collected from various portions of the packet header:

- If a tagged packet is received on a switch port, derive a priority value from PCP and drop precedence DEI value.
- If a tagged packet is received on a switch port on which trust level is set to DSCP, derive the priority value from DSCP bits. Drop precedence will be derived from DEI bit value.
- If an untagged packet is received on a switch port, assign priority value based on port level untagged priority configuration.
- If an untagged packet is received on a switch port on which trust level is set to DSCP, derive the drop precedence value from DSCP bits.
- If IPv4 packet is received on a router port, derive a priority value and drop precedence by decoding the DSCP bits.
- If MPLS packet is received on a label switched port, derive a priority value and drop precedence by decoding EXP bits.
- The derived values for PCP and DSCP are mapped to a default map.

CHAPTER 7 Modifying Internal Priority at Ingress

After the packet priority is recognized as PCP, DSCP or EXP values, priority queue and drop precedence will be determined by one of the following methods:

- Default ingress decode mappings are configured as explained in Chapter 10, *Default QoS Mappings*.
- Global ingress mapping profiles are available for modification to configure non-default values on a Qumran device as explained in Chapter 11, *Configuring QoS*. Following are the global ingress profiles affecting priority queues and drop precedence configuration:

    `qos profile cos-to-queue default` – Configuration in this profile allows modifying default mapping of PCP to priority queue globally.

    `qos profile dscp-to-queue default` – Configuration in this profile allows modifying default mapping of DSCP to priority queue and drop precedence globally.

    `qos profile exp-to-queue default` – Configuration in this profile allows modifying default mapping of EXP to priority queue and drop precedence globally.

- User-defined ingress decode mapping profiles can be created and bound to port to override the global configuration:

    `qos profile cos-to-queue NAME` – Configuration in this profile allows creating user-defined mapping of PCP to priority queue and can be bound to port to take effect.

    `qos profile dscp-to-queue NAME` – Configuration in this profile allows creating user-defined mapping of DSCP to priority queue and drop precedence and can be bound to port to affect.

- User-defined PCP to TC profile map can be bound to VPLS instance or per-VLAN-based attachment circuit and this mapping will override the profiles mapping bound to port or the global mapping profile.
- User-defined PCP to TC profile map can be bound to VPWS VLAN based attachment circuits and this mapping will override the profile mapping bound to port or the global mapping profile.
- User-defined PCP to TC profile map can be bound to registration entries for provider edge ports and this mapping will override the profile mapping bound to port or the global mapping profile.
- QoS policy-maps can be configured as explained in Chapter 2, *Configuring a QoS Policy-map* with set queue action and bound to an ingress port will override the priority queue configuration for the traffic that matches the policy-map configuration. Policy-map configured with `policer` action resulting in color marking will override the drop precedence on the traffic as per configuration.
- Internal priority for untagged traffic received on switch ports will be set to zero by default.
- Untagged traffic can be assigned with a non-default priority queue with port level configuration:
  `qos untagged-priority <0-7>` – This configuration assigns priority queue for untagged traffic received on the configured switch port.

CHAPTER 8 Remarking Packet Priority at Ingress

Apart from altering the internal priority and drop precedence, Qumran device allows modification of packet priority for PCP and DSCP values by one of the following methods:

- Configuration of ingress dscp-to-queue mapping profile allows the modification of DSCP value of the packet. This mapping is applicable only to IPv4 traffic on router port.
- QoS policy-maps can be configured with set CoS and set DSCP actions which will modify PCP or DSCP value of the packet, respectively. Set CoS action is applicable on switch ports and tagged packets. Set DSCP action is applicable on router port where IPv4 traffic is received on the port. Remarking DSCP through policy-map configuration will override the mapping profile remarking.

Note: Qumran does not support encoding EXP value from packet priority of tagged (PCP) or IPv4 traffic (DSCP) through ingress processing. EXP value encoding is supported only through egress processing. See Chapter 11, *Configuring QoS* for Remarking configuration.

© 2026 IP Infusion Inc. Proprietary

CHAPTER 9 Remarking Packet Priority at Egress

In Qumran, tagged traffic PCP value on switch port and IPv4 traffic DSCP value on router port can be remarked through egress processing. When either the remark CoS or remark DSCP command is enabled on the egress port, one of the following remarkings takes place:

- Default egress encode mappings are configured as explained in Chapter 10, *Default QoS Mappings*.
- Global egress mapping profiles are available for modification to configure non-default values on a Qumran device as explained in Chapter 11, *Configuring QoS*. Following are the global egress profiles affecting packet priority on traffic at egress:

  `qos profile queue-color-to-cos default` — Configuration in this profile allows modifying default mapping of internal priority and drop precedence to packet priority (PCP) globally.

  `qos profile dscp-to-dscp default` — Configuration in this profile allows modifying default mapping of DSCP and drop precedence to packet priority (DSCP) globally.

- User-defined egress encode mapping profiles can be created and bound to egress port to override the global configuration:

  `qos profile queue-color-to-cos NAME` — Configuration in this profile allows creating user-defined mapping of internal priority and drop precedence to packet priority (PCP) and can be bound to egress port to take effect.

  `qos profile dscp-to-dscp NAME` — Configuration in this profile allows creating user-defined mapping of DSCP and drop precedence to packet priority (DSCP) and can be bound to egress port to take effect.

Qumran supports encapsulation mapping profile to mark EXP values derived from DSCP values for IPv4 traffic and internal priority and drop precedence for tagged traffic through egress processing. EXP value encapsulation will be determined by one of the following methods:

- Default egress encapsulation mappings are configured as explained in "*Default QoS Mappings*."
- Global egress EXP encapsulation mapping profile is available for modifying DSCP value to EXP encapsulation value for IPv4 traffic, and internal priority and drop precedence values to EXP encapsulation for tagged traffic. The command: `qos profile exp-encp default,` allows configuration of L3 DSCP to EXP mapping, and L2 queue, color to EXP mapping globally.
- User-defined egress EXP encapsulation mapping profiles can be created and bound to egress label switched port to override the global configuration: `qos profile exp-encap NAME` — configuration in this profile allows creating user-defined mapping of L3 DSCP to EXP and L2 queue, color to EXP encapsulation values and can be bound to an egress label switched port to take effect on network edge devices (Ingress LER).

CHAPTER 10 Default QoS Mappings

If a user defined profile map is not created or applied to ingress or egress traffic, Qumran uses a default map to assign PCP, DSCP, EXP priority, and drop precedence values. The following tables describe the default QoS mapping values:

Ingress decode mapping table:

- PCP, DEI-to-TC and DP table
- DSCP-to-TC and DP table
- EXP-to-TC and DP table

Egress encode mapping table:

- TC and DP-to-PCP table
- DSCP, DP-to-DSCP table
- EXP encapsulation table

Table 10-2 lists the default PCP, DEI-to-TC, and DP mappings.

Table 10-2: Default PCP, DEI-to-TC and DP

PCP (CoS)	drop eligibility (DE)	Traffic class (queue)	Drop precedence (color)
0 (000)	0	0	Green
0 (000)	1	0	Yellow
1 (001)	0	1	Green
1 (001)	1	1	Yellow
2 (010)	0	2	Green
2 (010)	1	2	Yellow
3 (011)	0	3	Green
3 (011)	1	3	Yellow
4 (100)	0	4	Green
4 (100)	1	4	Yellow
5 (101)	0	5	Green
5 (101)	1	5	Yellow
6 (110)	0	6	Green
6 (110)	1	6	Yellow
7 (111)	0	7	Green
7 (111)	1	7	Yellow

© 2026 IP Infusion Inc. Proprietary

Table 10-3 lists the default TC and DP-to-PCP mappings. This table is effective when egress remarking is enabled.

Table 10-3: Default TC and DP-to-PCP

Traffic class (queue)	Drop precedence (color)	Priority (CoS/PCP)	Drop eligibility (CFI)
0	Green	0	0
0	Yellow/red	0	1
1	Green	1	0
1	Yellow/red	1	1
2	Green	2	0
2	Yellow/red	2	1
3	Green	3	0
3	Yellow/red	3	1
4	Green	4	0
4	Yellow/red	4	1
5	Green	5	0
5	Yellow/red	5	1
6	Green	6	0
6	Yellow/red	6	1
7	Green	7	0
7	Yellow/red	7	1

Note: Table information with red color is applicable only when red packet drop is disabled on the device. Otherwise, red packets are dropped by default.

Table 10-4 lists the default DSCP-to-TC, DP mappings..

Table 10-4: Default DSCP-to-TC and DP

DSCP	Traffic class (queue)	Drop precedence (color)
0 (000000 - BE/CS0)	0	Green
1 (000001)	0	Green
2 (000010)	0	Green
3 (000011)	0	Green
4 (000100)	0	Green
5 (000101)	0	Green
6 (000110)	0	Green

Table 10-4: Default DSCP-to-TC and DP (Continued)

DSCP	Traffic class (queue)	Drop precedence (color)
7 (000111)	0	Green
8 (001000 - CS1)	1	Green
9 (001001)	1	Green
10 (001010 - AF11)	1	Green
11 (001011)	1	Green
12 (001100 - AF12)	1	Yellow
13 (001101)	1	Green
14 (001110 - AF13)	1	Yellow
15 (001111)	1	Green
16 (010000 - CS2)	2	Green
17 (010001)	2	Green
18 (010010 - AF21)	2	Green
19 (010011)	2	Green
20 (010100 - AF22)	2	Yellow
21 (010101)	2	Green
22 (010110 - AF23)	2	Yellow
23 (010111)	2	Green
24 (011000 - CS3)	3	Green
25 (011001)	3	Green
26 (011010 - AF31)	3	Green
27 (011011)	3	Green
28 (011100 - AF32)	3	Yellow
29 (011101)	3	Green
30 (011110 - AF33)	3	Yellow
31 (011111)	3	Green
32 (100000 - CS4)	4	Green
33 (100001)	4	Green
34 (100010 - AF41)	4	Green
35 (100011)	4	Green

© 2026 IP Infusion Inc. Proprietary

Table 10-4: Default DSCP-to-TC and DP (Continued)

DSCP	Traffic class (queue)	Drop precedence (color)
36 (100100 - AF42)	4	Yellow
37 (100101)	4	Green
38 (100110 - AF43)	4	Yellow
39 (100111)	4	Green
40 (101000 - CS5)	5	Green
41 (101001)	5	Green
42 (101010)	5	Green
43 (101011)	5	Green
44 (101100)	5	Green
45 (101101)	5	Green
46 (101110)	5	Green
47 (101111)	5	Green
48 (110000 - CS6)	6	Green
49 (110001)	6	Green
50 (110010)	6	Green
51 (110011)	6	Green
52 (110100)	6	Green
53 (110101)	6	Green
54 (110110)	6	Green
55 (110111)	6	Green
56 (111000 - CS7)	7	Green
57 (111001)	7	Green
58 (111010)	7	Green
59 (111011)	7	Green
60 (111100)	7	Green
61 (111101)	7	Green
62 (111110)	7	Green
63 (111111)	7	Green

© 2026 IP Infusion Inc. Proprietary

Table 10-5 lists the default DSCP and DP-to-DSCP mapping. This table is effective when egress remarking is enabled.

Table 10-5: Default DSCP and DP-to-DSCP

DSCP	Drop precedence (color)	Out-DSCP
0 (000000 - BE/CS0)	Green/yellow/red	0 (000000 - BE/CS0)
1 (000001)	Green/yellow/red	1 (000001)
2 (000010)	Green/yellow/red	2 (000010)
3 (000011)	Green/yellow/red	3 (000011)
4 (000100)	Green/yellow/red	4 (000100)
5 (000101)	Green/yellow/red	5 (000101)
6 (000110)	Green/yellow/red	6 (000110)
7 (000111)	Green/yellow/red	7 (000111)
8 (001000 - CS1)	Green/yellow/red	8 (001000 - CS1)
9 (001001)	Green/yellow/red	9 (001001)
10 (001010 - AF11)	Green	10 (001010 - AF11)
10 (001010 - AF11)	Yellow	12 (001100 - AF12)
10 (001010 - AF11)	Red	14 (001110 - AF13)
11 (001011)	Green/yellow/red	11 (001011)
12 (001100 - AF12)	Green/yellow	12 (001100 - AF12)
12 (001100 - AF12)	Red	14 (001110 - AF13)
13 (001101)	Green/yellow/red	13 (001101)
14 (001110 - AF13)	Green/yellow/red	14 (001110 - AF13)
15 (001111)	Green/yellow/red	15 (001111)
16 (010000 - CS2)	Green/yellow/red	16 (010000 - CS2)
17 (010001)	Green/yellow/red	17 (010001)
18 (010010 - AF21)	Green	18 (010010 - AF21)
18 (010010 - AF21)	Yellow	20 (010100 - AF22)
18 (010010 - AF21)	Red	22 (010110 - AF23)
19 (010011)	Green/yellow/red	19 (010011)
20 (010100 - AF22)	Green/yellow	20 (010100 - AF22)
20 (010100 - AF22)	Red	22 (010110 - AF23)

© 2026 IP Infusion Inc. Proprietary

Table 10-5: Default DSCP and DP-to-DSCP (Continued)

DSCP	Drop precedence (color)	Out-DSCP
21 (010101)	Green/yellow/red	21 (010101)
22 (010110 - AF23)	Green/yellow/red	22 (010110 - AF23)
23 (010111)	Green/yellow/red	23 (010111)
24 (011000 - CS3)	Green/yellow/red	24 (011000 - CS3)
25 (011001)	Green/yellow/red	25 (011001)
26 (011010 - AF31)	Green	26 (011010 - AF31)
26 (011010 - AF31)	Yellow	28 (011100 - AF32)
26 (011010 - AF31)	Red	30 (011110 - AF33)
27 (011011)	Green/yellow/red	27 (011011)
28 (011100 - AF32)	Green/yellow	28 (011100 - AF32)
28 (011100 - AF32)	Red	30 (011110 - AF33)
29 (011101)	Green/yellow/red	29 (011101)
30 (011110 - AF33)	Green/yellow/red	30 (011110 - AF33)
31 (011111)	Green/yellow/red	31 (011111)
32 (100000 - CS4)	Green/yellow/red	32 (100000 - CS4)
33 (100001)	Green/yellow/red	33 (100001)
34 (100010 - AF41)	Green	34 (100010 - AF41)
34 (100010 - AF41)	Yellow	36 (100100 - AF42)
34 (100010 - AF41)	Red	38 (100110 - AF43)
35 (100011)	Green/yellow/red	35 (100011)
36 (100100 - AF42)	Green/yellow	36 (100100 - AF42)
36 (100100 - AF42)	Red	38 (100110 - AF43)
37 (100101)	Green/yellow/red	37 (100101)
38 (100110 - AF43)	Green/yellow/red	38 (100110 - AF43)
39 (100111)	Green/yellow/red	39 (100111)
40 (101000 - CS5)	Green/yellow/red	40 (101000 - CS5)
41 (101001)	Green/yellow/red	41 (101001)
42 (101010)	Green/yellow/red	42 (101010)
43 (101011)	Green/yellow/red	43 (101011)

Table 10-5: Default DSCP and DP-to-DSCP (Continued)

DSCP	Drop precedence (color)	Out-DSCP
44 (101100)	Green/yellow/red	44 (101100)
45 (101101)	Green/yellow/red	45 (101101)
46 (101110)	Green/yellow/red	46 (101110)
47 (101111)	Green/yellow/red	47 (101111)
48 (110000 - CS6)	Green/yellow/red	48 (110000 - CS6)
49 (110001)	Green/yellow/red	49 (110001)
50 (110010)	Green/yellow/red	50 (110010)
51 (110011)	Green/yellow/red	51 (110011)
52 (110100)	Green/yellow/red	52 (110100)
53 (110101)	Green/yellow/red	53 (110101)
54 (110110)	Green/yellow/red	54 (110110)
55 (110111)	Green/yellow/red	55 (110111)
56 (111000 - CS7)	Green/yellow/red	56 (111000 - CS7)
57 (111001)	Green/yellow/red	57 (111001)
58 (111010)	Green/yellow/red	58 (111010)
59 (111011)	Green/yellow/red	59 (111011)
60 (111100)	Green/yellow/red	60 (111100)
61 (111101)	Green/yellow/red	61 (111101)
62 (111110)	Green/yellow/red	62 (111110)
63 (111111)	Green/yellow/red	63 (111111)

Note: Table information with red color is applicable only when red packet drop is disabled on the device. Otherwise, red packets are dropped by default.

Table 10-6 lists the default EXP-to-TC and DP mappings.

Table 10-6: Default EXP-to-TC and DP

EXP	Traffic class (queue)	Drop precedence (color)
0 (000)	0	Green
1 (001)	1	Green
2 (010)	2	Green
3 (011)	3	Green

Table 10-6: Default EXP-to-TC and DP (Continued)

EXP	Traffic class (queue)	Drop precedence (color)
4 (100)	4	Green
5 (101)	5	Green
6 (110)	6	Green
7 (111)	7	Green

Table 10-7 and Table 10-8 list the default EXP encapsulation mappings:

Table 10-7: Default EXP encapsulation table (L2 traffic)

Traffic class (queue)	Drop precedence (color)	EXP
0	Green/yellow/red	0
1	Green/yellow/red	1
2	Green/yellow/red	2
3	Green/yellow/red	3
4	Green/yellow/red	4
5	Green/yellow/red	5
6	Green/yellow/red	6
7	Green/yellow/red	7

Note: Table information with red color is applicable only when red packet drop is disabled on the device. Otherwise, red packets are dropped by default.

Table 10-8 displays the default EXP encapsulation table (L3 traffic).

Table 10-8: Default EXP encapsulation table (L3 traffic)

DSCP	EXP
0 - 7	0
8 - 15	1
16 - 23	2
24 - 31	3
32 - 39	4
40 - 47	5

Table 10-8: Default EXP encapsulation table (L3 traffic) (Continued)

DSCP	EXP
48 - 55	6
56 - 63	7

© 2026 IP Infusion Inc. Proprietary

CHAPTER 11 Configuring QoS

The configuration process involving several commands is described in the following chapter.

Configuring Ingress QoS Procedures

Following section explains the configuration details involved to achieve the ingress QoS treatment as explained in Chapter 7, *Modifying Internal Priority at Ingress* and Chapter 8, *Remarking Packet Priority at Ingress*.

The configuration steps involved are described below:

- Configuring trust level on switch ports
- Configuring internal priority for untagged traffic on switch ports
- Configuring ingress decode mapping profile
- Binding ingress decode mapping profile
- Configuring policy-map to modify internal or packet priority

Configuring trust level on switch ports

Switch ports support two trust levels. By default, trust level is based on PCP value. However, trust level based on DSCP value is configurable per port.

Trust level to DSCP value can be configured through the command:

```
trust dscp
```

When trust `DSCP` is configured on the port, internal priority will be derived from `DSCP` value of IPv4 packet based on default `dscp-to-queue` profile configuration. Drop precedence for untagged traffic will be derived from `DSCP` value of IPv4 packet based on `default dscp-to-queue` profile configuration. For tagged traffic, drop precedence will continue to be based on `DEI` bit on the packet.

Configuring internal priority for untagged traffic on switch ports

For tagged traffic, internal priority will be derived from PCP value of the packet. Untagged traffic will not carry PCP value and are assigned with internal priority zero by default. Internal priority can be assigned to untagged traffic with following port level configuration:

```
qos untagged-priority <0-7>
```

With untagged priority configured on switch port, internal priority will be selected for untagged traffic based on the configured value.

An example of the configuration is shown below:

```
(config)#interface xe1
(config-if)#switchport
(config-if)#qos untagged-priority 3
(config-if)#exit
```

In the example configuration, untagged traffic received on xe1 will be assigned with internal priority 3 and will be transmitted out of queue 3 at egress port.

Configuring Ingress Decode Mapping Profile

In order to modify the priority and drop precedence values used within the device, user-defined decode mapping profiles can be created or global decode mapping profile contents can be modified.

© 2026 IP Infusion Inc. Proprietary

Three types of decode mapping profiles are supported based on packet priority encoded in tagged IPv4 and in MPLS traffic.

Configuring PCP-to-TC Mapping Profile

User-defined profile can be created or global profile can be modified for PCP-to-TC through the following command:

```
qos profile cos-to-queue (NAME | default)
```

Inside this command PCP-to-TC mapping can be configured using the values:

```
cos <0-7> queue <0-7>
```

Here, CoS is the PCP value and the queue value is the traffic-class being configured.

An example of the configuration is shown below:

```
OcNOS(config)#qos profile cos-to-queue profile1
OcNOS(config-ingress-cos-map)#cos 3 queue 4
OcNOS(config-ingress-cos-map)#cos 2 queue 1
OcNOS(config-ingress-cos-map)#exit
```

While egressing out when remarking is enabled (On Qumran2 series platforms)

```
OcNOS(config)#qos profile cos-to-queue profile1
OcNOS(config-ingress-cos-map)#cos 3 queue 3 remark-queue 3
OcNOS(config-ingress-cos-map)#cos 1 queue 1 remark-queue 1
OcNOS(config-ingress-cos-map)#exit
```

Note: This mapping profile is applicable only to switch ports.

Configuring DSCP-to-TC, DP, and DSCP Mapping Profile

User-defined profiles can be created or global profiles can be modified for DSCP-to-TC, DP, and DSCP using the following command:

```
qos profile dscp-to-queue (NAME | default)
```

Inside this command DSCP-to-TC, DP, and DSCP mapping can be configured using the command:

```
dscp <0-63> queue <0-7> (color (green | yellow | red)|) (dscp <0-63>|)
```

An example configuration is shown below:

```
OcNOS(config)#qos profile dscp-to-queue profile2
OcNOS(config-ingress-dscp-map)#dscp 8 queue 2
OcNOS(config-ingress-dscp-map)#dscp 25 queue 4 color yellow
OcNOS(config-ingress-dscp-map)#exit
```

Note: This mapping profile is applicable only on router ports. However, the default mapping profile is applicable on switch ports as well, if trust level on the switch port is set to DSCP.

For IP, ingress stage remarking of DSCP values can be achieved by configuring the output DSCP in the command, however, this remarking can be overwritten by policy-map based remarking or by egress remarking.

For MPLS, ingress stage remarking of DSCP values is used to derive EXP value when it enters the MPLS network using queue-to-exp mapping.

An example configuration is shown below:

```
OcNOS(config)#qos profile dscp-to-queue profile1
OcNOS(config-ingress-dscp-map)#dscp 10 queue 5 dscp 24
OcNOS(config-ingress-dscp-map)#exit
```

In the example, configuration profile is bound to ingress router port, then traffic with DSCP 10 will be set with internal priority 5 and remarked DSCP value 24 will be used to calculate EXP value (Remark_DSCP/8= Queue) using queue-exp mapping at ingress node Egress Network interface (MPLS Network).

Example:

```
(config-ingress-dscp-map)#dscp 10 queue 3 dscp 24
```

Remark_DSCP = 24

EXP calculation:

(Remark_DSCP/8=EXP) = 24/8 = 3

Queue to EXP mapping always one-to-one when enters MPLS network, EXP value assigned to 3.

Refer to Table 10-7 and Table 10-8 for a list of the default EXP encapsulation mappings.

Configuring EXP-to-TC and DP Mapping Profile

Global profile can be modified for EXP-to-TC and DP through the following command:

```
qos profile exp-to-queue default
```

Inside this command, PCP-to-TC mapping can be configured using the command:

```
exp <0-7> queue <0-7> (color (green | yellow | red)|)
```

An example configuration is shown below:

```
(config)#qos profile exp-to-queue default
(config-ingress-exp-queue-map)#exp 3 queue 4 color yellow
(config-ingress-exp-queue-map)#exp 2 queue 1
(config-ingress-exp-queue-map)#exit
```

Note: This mapping profile is applicable only on label-switched router ports.

Binding Ingress Decode Mapping Profile

User-defined mapping profiles will be effective only when they are bound to ingress ports or any services.

User-defined profiles can be bound to port using the command:

```
qos map-profile (cos-to-queue | dscp-to-queue) NAME
```

An example of binding user-defined map to port is shown below:

```
(config)#interface xe1
(config-if)#switchport
(config-if)#qos map-profile cos-to-queue profile1
```

PCP-to-TC decode profiles can be bound to VPLS service instances or attachment circuits. Profile binding to these services can be achieved using the commands:

```
vpls-qos map-profile cos-to-queue NAME
vc-qos map-profile cos-to-queue NAME
```

An example of binding a PCP-to-TC mapping profile to a VPLS service is shown below:

```
(config)#mpls vpls vpls1 1
(config-vpls)#vpls-qos map-profile cos-to-queue cq-profile-1
```

An example of binding a PCP-to-TC profile to an attachment circuit is shown below:

```
(config)#interface xe1
(config-if)#switchport
(config-if)# mpls-vpls MPLS-VPLS service-template VPLS-10
(config-if-vpls)#vc-qos map-profile cos-to-queue cq-profile-2
```

Profile configured on the VPLS service is effective for all attachment circuits associated with VPLS service. However, if a profile is bound to an attachment circuit, then that profile takes higher priority for that attachment circuit.

Traffic received on VPLS service will be affected by QoS treatment by configurations in the following order:

1. Set queue through policy-map configuration. Refer below section for details.
2. PCP-to-TC profile configuration bound to attachment circuit.
3. PCP-to-TC profile configuration bound to VPLS service.
4. PCP-to-TC profile configuration bound to ingress port.
5. PCP-to-TC profile configuration.

Configuring Policy-map to Modify Internal or Packet Priority

QoS policy-map frame work can be used to modify the internal priority or remark the packet priority. For policy-map configuration, refer to Chapter 2, *Configuring a QoS Policy-map*. Set action has following two options:

- Modify internal priority - Internal priority or traffic class can be set using the command set queue <0-7> and this command will modify only the internal priority and the configuration is not carried forward to next device.
- Remark packet priority - Packet priority can be set at the ingress processing through set cos <0-7> and set dscp <0-63> to modify PCP value for tagged traffic received on switch port and DSCP value for IPv4 traffic received on router port. Both commands implicitly modify internal priority. Value for internal priority is derived from cos-to-queue profile bound to switch port and DSCP-to-queue profile bound to router port. If profile is not explicitly bound on the interface, internal priority will be derived from default cos-to-queue profile or default dscp-to-queue profile based on interface type. On VLAN interface, set cos will update only internal priority if the traffic is routed. Even if the traffic is routed to another VLAN interface, set cos will not set the packet priority on newly constructed VLAN header and it will be 0 or subject to egress remarking configuration. If traffic is switched, then set cos on VLAN interface will update both internal priority and packet priority. Qumran does not support remarking on MPLS traffic.

Configuration example for modifying traffic class is shown below:

```
(config)#policy-map p-1
(config-pmap-qos)#class c-1
(config-pmap-c-qos)#set queue 5
(config-pmap-c-qos)#exit
```

The commands `set cos` and `set dscp` can be configured similarly.

Configuring Egress QoS Procedures

Following section explains the configuration details involved to achieve the egress QoS treatment as explained in Chapter 9, *Remarking Packet Priority at Egress*.

Configuration steps involved are described below:

- Configuring egress remarking
- Configuring egress encode mapping profile
- Binding egress encode mapping profile
- Configuring and binding EXP encapsulation profile

Configuring Egress Remarking

Qumran supports remarking through egress processing. Remarking is configurable separately for tagged traffic and IPv4 traffic. Remarking can be enabled on the device through global configuration or per interface configuration.

Commands provide the flexibility to configure enable remarking globally but selectively disable on certain ports or enable only on certain ports.

An example of enabling egress remarking globally is shown below:

```
(config)#qos remark cos
```

This configuration takes effect on all egress switch ports and tagged traffic transmitted through the port will be egress remarked.

The syntax of the command is:

```
qos remark (cos | dscp)
```

An example of enabling egress remarking on an egress port is shown below:

```
(config)#interface xe1
(config-if)#switchport
(config-if)# qos remark cos enable
```

This configuration takes effect on xe1 and tagged traffic transmitted through xe1 will be egress remarked.

The syntax of the command is:

```
qos remark (cos | dscp) (enable | disable)
```

Configuring Egress Encode Mapping Profile

In order to modify the default encoding packet priority values when remarking is enabled on the device, a user-defined encode mapping profile can be created or global decode mapping profile content can be modified.

Two types of encode mapping profiles are supported in Qumran device.

Configuring TC, DP-to-PCP mapping profile

User-defined profile can be created or global profile can be modified for TC, DP-to-PCP using the following command:

```
qos profile queue-color-to-cos (NAME | default)
```

In this profile mode TC, DP-to-PCP mapping can be configured through the command:

```
queue <0-7> (color (green | yellow)| red) cos <0-7>
```

Here, the queue value is the traffic class, color is the drop precedence, and CoS is the PCP value to be remarked.

An example of a configuration is shown below:

```
(config)#qos profile queue-color-to-cos profile1
(config-egress-cos-map)#queue 4 cos 3
(config-egress-cos-map)#queue 1 color yellow cos 2
(config-egress-cos-map)#exit
```

Note: This mapping profile is applicable only on switch ports.

Configuring DSCP, DP to DSCP mapping profile

User-defined profiles can be created or global profiles can be modified for DSCP, DP-to-DSCP using the following command:

```
qos profile dscp-to-dscp (NAME | default)
```

Inside this profile mode DSCP, DP-to-DSCP mapping can be configured using the command:

```
dscp <0-63> (color (green | yellow)| red) dscp <0-63>
```

A configuration example is shown below:

```
(config)#qos profile dscp-to-dscp profile2
(config-egress-dscp-map)#dscp 8 dscp 7
```

```
(config-egress-dscp-map)#dscp 24 color yellow dscp 30
(config-egress-dscp-map)#exit
```

Note: This mapping profile is applicable only on router ports.

Note: When ingress DSCP-to-TC, DP, or DSCP profiles with ingress DSCP remarking is bound to an ingress router port, remarked DSCP values will be input for egress DSCP and DP-to-DSCP profiles on the egress interface.

An example of this case is shown below:

```
(config)#qos profile dscp-to-queue profile1
(config-ingress-dscp-map)#dscp 10 queue 3 dscp 24
```

If this profile is bound to ingress router port, then traffic with DSCP 10 will be set with internal priority 3 and DSCP will be remarked to 24 at ingress stage.

```
(config)#qos profile dscp-to-dscp profile2
(config-egress-dscp-map)#dscp 24 dscp 30
```

If the above profile is bound to egress router port, then the mapping entry will match the traffic with remarked DSCP 24 and will effectively update the DSCP value to 30 while egressing the traffic. As a result, traffic with DSCP value 10 will be remarked to DSCP value 30.

Binding egress encode mapping profile

For user-defined mapping profiles to work, they must be bound to egress ports and egress remarking must be enabled.

User-defined profiles can be bound to ports using the following command:

```
qos map-profile (queue-color-to-cos | dscp-to-dscp) NAME
```

An example of binding a user-defined map to a port is shown below:

```
(config)#interface xe1
(config-if)#qos map-profile dscp-to-dscp profile1
```

TC and DP-to-PCP encode profiles can be bound to a VPLS service instance or attachment circuit. Profile binding to these services can be achieved using the following commands:

```
vpls-qos map-profile queue-color-to-cos NAME
vc-qos map-profile queue-color-to-cos NAME
```

An example of binding PCP-to-TC mapping a profile to a VPLS service is shown below:

```
(config)# mpls-vpls MPLS-VPLS service-template VPLS-10
(config-vpls)#vpls-qos map-profile queue-color-to-cos qc-profile-1
```

An example of binding TC, DP-to-PCP profile to attachment circuit is shown below:

```
(config)#interface xe1
(config-if)#switchport
(config-if)#mpls-vpls vpls1 service-template st1
(config-if-vpls)#vc-qos map-profile queue-color-to-cos qc-profile-2
```

QoS egress remarking treatment on VPLS service has multiple configurations available and the priority of configuration is similar to the one explained in Binding Ingress Decode Mapping Profile.

Configuring and Binding EXP Encapsulation Profile

When a packet enters an MPLS network side from access side, the MPLS label will be encapsulated to the packet. Qumran supports inheriting packet priority to MPLS header from DSCP values for IPv4 traffic. For tagged traffic, packet priority is inherited by the MPLS header from the traffic class and drop precedence.

Qumran supports both global EXP encapsulation mapping profile and user-defined EXP encapsulation mapping profiles per egress port.

User-defined profiles can be created or global profile can be modified for EXP encapsulation using the following command:

```
qos profile exp-encap (NAME | default)
```

Inside this profile mode, DSCP to EXP mapping for L3 traffic and queue, color to EXP mapping for L2 traffic can be configured using the following ranges:

```
l3 dscp <0-63> exp <0-7>
l2 queue <0-7> (color (green|yellow|red)|) exp <0-7>
```

An example of binding a user-defined map to an egress port of ingress LER is shown below:

```
(config)#interface xe1
(config-if)#label-switching
(config-if)#qos map-profile exp-encap profile1
```

Note: This profile is applicable only on network facing ports of the ingress LER device.

CHAPTER 12 Displaying QoS Information

The following QoS information can be displayed:

- QoS Configuration Information – QoS configuration such as qos mapping profiles, class-maps and policy-maps can be verified by using show commands.
- QoS Packet and Byte Statistics – count of packets and bytes matching the match criteria configured in class-maps bound to ingress ports through policy-maps can be displayed when QoS statistics is enabled.

Displaying QoS Configuration Information

The QoS configurations that can be displayed are listed below:

- Mapping profile configuration
- Class-map configuration
- Policy-map configuration

Display Mapping Profile Configuration

Mapping profile configuration can be displayed using the command:

```
show qos-profile (type (cos-to-queue | dscp-to-queue | exp-to-queue | queue-color-to-
cos | dscp-to-dscp | dscp-to-exp) |) (PROFILE-NAME|)
```

Refer to the “*Configuration Guide*” for detailed information of the mapping profile output.

Display Class-map Configuration

Class-map configuration can be verified with the command `show class-map (type qos|NAME|)`.

Sample output for the class-map configuration is shown below:

```
#show class-map type qos

 Type qos class-maps
 ===================
      class-map c1
         match vlan 200

      class-map type qos match-any class-default
```

Display policy-map configuration

Policy-map configuration can be verified with the command `show policy-map type qos (NAME|)`.

Sample output for the policy-map configuration is shown below:

```
#show policy-map type qos

Type qos policy-maps
====================

policy-map type qos p1
 class type qos c1
  police cir 2 mbps
  exit
```

© 2026 IP Infusion Inc. Proprietary

Display Policy-map Configured on an Interface

Type qos policy-map configured on an interface is displayed by the command show policy-map interface INTERFACE-NAME type qos.

Sample output of this command is shown below:

```
#show policy-map interface xe1 type qos

Interface xe1
Type QoS statistics status : disabled

Service-policy (qos) input: parentPmap
----------------------------
Class-map (qos): vlan2 (match all)
 match vlan 2
 police cir 2 mbps

     Child Service-policy (qos) : childPmap
     ----------------------------
     Class-map (qos): class-default (match any)

     Class-map (qos): cos1 (match all)
      match cos 1
      police cir 1 mbps
```

To display a specific class within type qos policy-map configured on the interface use the command `show policy-map interface INTERFACE-NAME (class NAME|) type qos`.

Sample output of this command is shown below:

```
#show policy-map interface xe0 class cos1 type qos

Interface xe0

Type QoS statistics status : disabled

Class-map (qos): vlan2 (match all)
     Class-map (qos): cos1 (match all)
      match cos 1
      police cir 1 mbps
```

Displaying QoS packet and Byte Counters

QoS statistics can be enabled, displayed and cleared as described in following section:

- Enabling QoS packet and byte counters
- Displaying QoS packet and byte counters
- Clearing QoS packet and byte counters

Enabling QoS Packet and Byte Counters

Enable ingress and egress statistics using the command qos statistics as shown below.

```
(config)#qos statistics
```

The default for this command is disabled statistics.

Using `no` option statistics can be disabled.

Displaying QoS Packet and Byte Counters

Once statistics are enabled, type qos policy-map statistics and configurations can be displayed using the command `show policy-map interface INTERFACE-NAME type qos`.

A Sample output of this command is shown below:

```
#show policy-map interface xe0 type qos

Interface xe0

Type QoS statistics status : enabled

Class-map (qos): vlan2 (match all)
 match vlan 2
 police cir 2 mbps

     Class-map (qos): class-default (match any)
           matched      : 5503 packets, 8254500 bytes
           transmitted  : 92 packets, 138000 bytes
           dropped      : 5411 packets, 8116500 bytes

     Class-map (qos): cos1 (match all)
      match cos 1
      police cir 1 mbps
           matched      : 11010 packets, 16513500 bytes
           transmitted  : 94 packets, 141000 bytes
           dropped      : 10916 packets, 16372500 bytes
```

Statistics of type qos class-maps in the type qos policy-maps that are configured on the interfaces can be displayed using `show policy-map statistics (interface INTERFACE-NAME|) (class NAME|) type qos.`

A Sample output of this command is shown below:

```
#show policy-map statistics type qos
+----------------+-----------------+-----------------+-----------------+-------------+
|    Class-map   |    Match pkts   |   Match bytes   |   Dropped pkts  |   Dropped Bytes
|
+----------------+-----------------+-----------------+-----------------+-------------+
xe0
 vlan2
   cos1             9012               13516500          8935              13401000
xe2
 vlan2
   class-default    4452               6678000           4302              6453000


#show policy-map interface xe0 statistics type qos
+----------------+-----------------+-----------------+-----------------+------------+
```

```
|    Class-map   |    Match pkts   |   Match bytes   |   Dropped pkts  |   Dropped Bytes
|
+----------------+-----------------+-----------------+-----------------+------------+
 vlan2
   cos1             1073131           1609695000       1063993           1595988000


#show policy-map statistics class cos1 type qos
+----------------+-----------------+-----------------+-----------------+-----------+
|    Class-map   |    Match pkts   |   Match bytes   |   Dropped pkts  |   Dropped Bytes
|
+----------------+-----------------+-----------------+-----------------+-----------+
xe0
 vlan2
   cos1             1399290           2098933500       1387374           2081059500


#show policy-map int xe0 statistics class cos1 type qos
+----------------+-----------------+-----------------+-----------------+----------+
|    Class-map   |    Match pkts   |   Match bytes   |   Dropped pkts  |   Dropped Bytes
|
+----------------+-----------------+-----------------+-----------------+----------+
 vlan2
   cos1             1563686           2345527500       1550370           2325553500
```

Note: In order to check statistics, QoS statistics profile need to be enabled for Qumran devices. QoS can either use ingress-acl statistics profile or ingress-qos statistics profile. When ACL groups are configured on the same interface as QoS and both ACL and QoS need explicit counters, then ingress-qos statistics profile need to be configured along with ingress-acl statistics profile. However, this will have other limitations on statistics profiles. See the `hardware-profile statistics` command in the *System Management Guide* for details.

Clearing QoS Packet and Byte Counters

QoS statistics can be cleared using the command `clear qos statistics`. This clears both ingress QoS and egress queuing statistics.

CHAPTER 13 Configuring Egress Queues on Ports

Every physical port of a Qumran device has eight priority queues and every subinterface has four priority queues. These ports and subinterfaces can be applied with several egress QoS parameters – these will be discussed in the next sections. (For more about subinterfaces, see Chapter 22, *Subinterface Queuing*.) When the QoS feature is enabled, all priority queues of the ports are configured with certain default egress queuing parameters.

To customize the treatment on the priority queues, the queuing policy-map infrastructure needs to be used. The following section explains the basic configuration details involved to apply queue level treatment on a port.

Configuring the Default Queuing Policy-Map

When the QoS feature is enabled, all ports of theQumran device is supplied with a default policy-map of queuing type. The default policy-map is created with the name “`default-out-policy` which is reserved and modifying parameters in this policy-map is reflected on all ports that do not have customized queuing policy-maps. Customized queuing policy-maps can be created and bound to ports to treat ports differently from the default configuration.

The `default-out-policy` policy-map is created with the default eight classes and the default subinterface `subif-default-out-policy` policy-map as Qumran supports eight priority queues per port and four priority queues per subinterface. Once the policy-map is configured, priority queue class-maps can be configured with the following command:

```
class type queuing default (q0|q1|q2|q3|q4|q5|q6|q7)
```

Class-maps `qx` represent the respective priority queuing class-maps which can be configured with different queue level parameters.

Creating a Queuing Class-Map

```
class-map type queuing NAME
no class-map type queuing NAME
```

Matching criteria: Only a match queue is supported in a queuing class-map for a user-defined queuing policy-map.

```
(no|) match queue <0-7>
```

Note: The match queue range 0-7 is valid only for port queues classification.

For subinterface queues, the valid range is 0-3.

Creating a Queuing Policy-Map

The following is the command to create a customized default policy-map:

```
policy-map type queuing NAME
no policy-map NAME
```

Binding a Queuing Policy-map

Customized queuing policy-maps take affect only when the configuration is bound to a port. Queuing policy-maps can be bound to the port with the following command:

```
service-policy type queuing output NAME
```

Where `NAME` represents the name of the queuing policy-map.

CHAPTER 14 Congestion Avoidance

Congestion avoidance techniques monitor network traffic loads in an effort to anticipate and avoid congestion at common network bottlenecks. Congestion avoidance is achieved through packet dropping. Among the more commonly used congestion avoidance mechanisms, Weighted Random Early Detection (WRED) is optimum for high-speed transit networks.

Qumran supports two types of congestion avoidance mechanisms.

- Tail drop – this is the default congestion avoidance behaviour when WRED is not configured
- WRED (Weighted Random Early Detection): This is applied only when configured

Tail Drop

Tail drop treats all traffic equally and does not differentiate between classes of service. Queues get filled during period of congestion. When the output queue is full and tail drop is in effect, packets are dropped until the congestion is eliminated and the queue is no longer full.

Configuring Packet Drop Priority Using Tail Drop

Qumran supports configuring color independent tail drop per queue, where the default queue-limit is 62914560 bytes (~62MB). The maximum queue limit (629145600) is not guaranteed in case of congestion because this is shared memory.

```
queue-limit <1-629145600> (packets | bytes | kbytes | mbytes | ms | us)
```

no queue-limit

Tail drop configurable parameters are listed below:

- Threshold in bytes, kilobytes, megabytes, packets or millisecond format.
- Ranges of different units are as follow:
    - Packets: min 9 – max 614400
    - Bytes: min 9416 – max 629145600
    - Kilo-bytes: min 9 – max 614400
    - Mega-bytes: max 600
    - Milliseconds: max 50
    - Microseconds: max50000

An example of configuring per-priority queue Tail Drop is shown below:

```
(config)#policy-map type queuing default pq-taildrop
(config-pmap-que-def)#class type queuing default q3
(config-pmap-c-que-def)#queue-limit 1 mbytes
(config-pmap-c-que-def)#exit
```

© 2026 IP Infusion Inc. Proprietary

Queue Drop Counters Verification

Drop counters with drop reason can be verified globally by using the following command:

```
#show hardware-discard-counters
+-------------------------------------------+---------------+--------------+
| Registers                                 | Core 0        | Core 1       |
+-------------------------------------------+---------------+--------------+
IQM_QUEUE_ENQ_DISCARDED_PACKET_COUNTER        1596100
   Reason: VOQ_VOQ_MX_QSZ_STATUS               Y
EGQ_PQP_DISCARD_UNICAST_PACKET_COUNTER        59807
   Reason: SRC_EQUAL_DEST_INT                  Y
```

Weighted Random Early Discard (WRED)

Qumran assigns each port eight priority queues to buffer traffic level that exceeds the port's total bandwidth. When traffic congestion persists, packets are dropped randomly. As a result traffic of greater priority may be dropped instead of traffic with lower priority.

In order to protect higher priority traffic from being dropped in such a scenarios, Qumran supports monitoring traffic congestion and drop packets based on a Weighted Random Early Discard (WRED) algorithm. Early detection of traffic congestion helps in avoiding global synchronization.

This algorithm enables the system to calculate current average queue length and compares the value against the configured minimum and maximum threshold values. Configured weight is a factor of calculating average queue length. If the calculated average queue length is within the configured minimum threshold, then the packet is enqueued. If the queue length is more than the configured maximum threshold, the packet will be dropped. When the queue length increases above minimum threshold and is within the configured maximum threshold, decision to enqueue or drop the packet is taken based on the configured drop probability. Higher drop probability packets are dropped and lower drop probability packets are enqueued.

When the current average queue length is slightly above the minimum threshold, packet drop count will be lower and increases as the current queue length reaches towards maximum threshold. Drop probability configuration decides the fraction of packet drops when average queue length reaches maximum threshold.

This method allows the application to take protective measures and synchronize the lost sessions over a period of time and thus avoiding sudden surges of traffic. Configurable parameters should be effective enough to provide enough time for the application to take corrective measures. If the gap between minimum threshold and maximum threshold values is small, then the time required for average queue length to increase from minimum threshold to maximum threshold is less and the algorithm will be ineffective.

Figure 14-2 shows how weighted random early discard works.

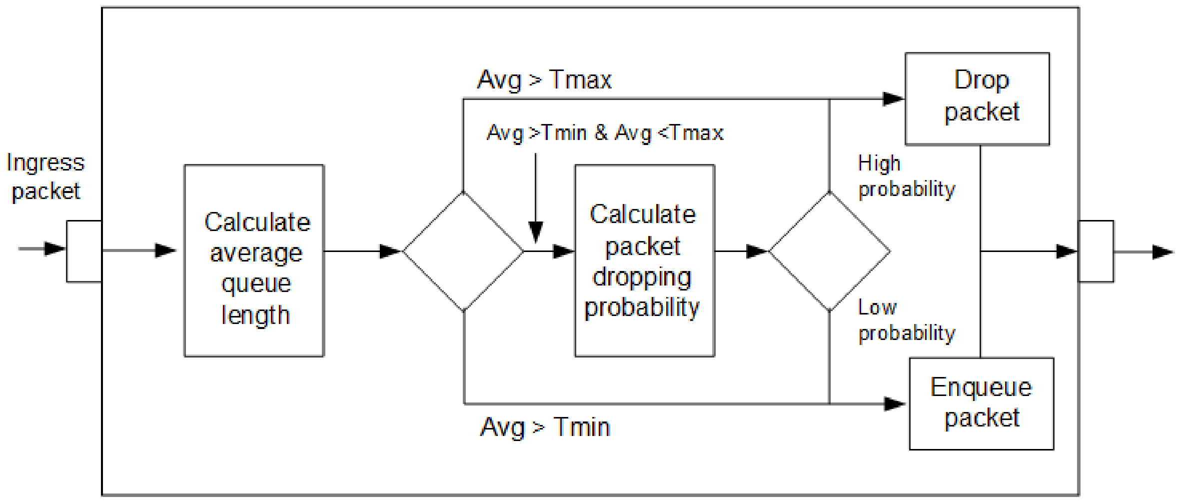


Figure 14-2: Weighted random early discard flow algorithm

Configuring Packet Drop Probability Using WRED

Qumran device supports both color independent per-priority queue WRED configuration and per-color, per-priority queue WRED configuration. One of the options is configurable per queue based on the requirement.

WRED configurable parameters are listed below:

- min_threshold: Minimal average queue size to apply WRED. Can be configured in bytes, kilobytes, megabytes, packets or millisecond format.
- max_thershold: Maximum average queue size to apply WRED. Can be configured in bytes, kilobytes, megabytes, packets or millisecond format.
- weight–factor of current size versus average size, in the calculation of new average size. Can be configured in range of 1 to 31 (optional).
- drop probability–maximum WRED drop probability applied at maximum threshold. Can be configured in percentage (optional).

At lower weight, average queue size will increase at the rate of instantaneous queue size, hence more packets will be dropped by WRED before queuing.

In this case, yellow packets are dropped more than green packets. At higher weight, average queue size will increase very slowly in comparison to instantaneous queue size, hence fewer packets will be dropped by WRED and more number packets will be queued. As a result we will observe similar results to tail drop.

Note: Formula to calculate new queue size:

NEW_AVG = OLD_AVG + (NEW_AVG-OLD_AVG) / (2 ^gain)

Formula to calculate drop probability:

Drop Probability = (AVRG_Q_SIZE-PQAVRGMINTH)/(PQAVRGMAXTH-PQAVRGMINTH) * 100%

It is recommended to use a weight value between 1 to 3 to achieve proper WRED functionality.

An example of configuring per-priority queue WRED is shown below:

```
(config)#policy-map type queuing default pq-wred
(config-pmap-que-def)#class type queuing default q3
(config-pmap-c-que-def)#random-detect 30 mbytes 500 mbytes drop-probability 50
(config-pmap-c-que-def)#exit
```

Both the minimum and maximum threshold must be configured in the same format. Thresholds can be configured in the range of 1 kilobyte to 600 megabytes or 50 milliseconds for millisecond format. In the above configuration, priority queue q3 is configured with a minimum threshold of 30 megabytes and a maximum threshold of 500 megabytes and a drop probability of 50%.

```
random-detect (green|yellow|red|)(min-threshold|)<1-629145600>
(packets|bytes|bytes|mbytes|ms|us) (max-threshold|)<1-629145600>
(packets|bytes|kbytes|mbytes|ms|us)(drop-probability <1-100>|)
```

An example of configuring per-color, per-priority queue WRED is shown below:

```
(config)#policy-map type queuing default pq-c-wred
(config-pmap-que-def)#class type queuing default q3
(config-pmap-c-que-def)#random-detect yellow 30 mbytes 100 mbytes drop-probability 10
(config-pmap-c-que-def)#random-detect green 100 mbytes 300 mbytes drop-probability 10
(config-pmap-c-que-def)#random-detect weight 3
(config-pmap-c-que-def)#exit
```

Configuration allows minimum threshold, maximum threshold and drop probability to be configured differently for green packets and yellow packets. In the above configuration priority queue q3 will be configured with minimum threshold of 30 megabytes and maximum threshold of 100 megabytes for yellow with drop probability for 10% and minimum threshold of 100 mbps and max threshold of 300 mbps for green with drop probability of 10% for all colors along with a weight value of 3.

See Chapter 13, *Configuring Egress Queues on Ports* for details about queuing property configuration.

Queue drop counters verification

Drop counters with drop reason can be verified globally by using the following command:

```
#show hardware-discard-counters
+--------------------------------------------+---------------+---------------+
| Registers                                  | Core 0        | Core 1        |
+--------------------------------------------+---------------+---------------+
IQM_QUEUE_ENQ_DISCARDED_PACKET_COUNTER        1596100
   Reason: VOQ_WRED_STATUS                     Y
EGQ_PQP_DISCARD_UNICAST_PACKET_COUNTER        59807
   Reason: SRC_EQUAL_DEST_INT                  Y
```

CHAPTER 15 Scheduling

Qumran can process all traffic if it is within the capacity of the device and all traffic will be forwarded as received. When the device reaches bandwidth constraint stage, traffic becomes subject to drop as described in Configuring Packet Drop Probability Using WRED or traffic scheduling as described in this section. Qumran classifies packets into one of eight internal priorities. Traffic scheduling allows to selectively forward traffic according to the forwarding queue that is mapped according to one of the following algorithm:

- Strict priority-based scheduling – This scheduling ensures the higher priority traffic is serviced ahead of lower priority traffic. As a result lower priority traffic may suffer from any access.
- WFQ (Weighted Fair Queuing) weight-based scheduling – In this scheduling, some weight based bandwidth is allocated to all queues. In this scheduling, egress traffic will be served based on the configured weight distribution.
- Mixed strict and weight based scheduling – This scheduling provides a mixture of strict priority for the higher priority queues and WFQ for the remaining priority queues. In this scheduling, strict priority should always be configured from highest to lower priority queues in sequential order and WFQ scheduling should be configured from lowest priority queues to higher priority queues in sequential order.

Configuring traffic scheduling

Traffic scheduling can be configured on a per port basis. It affects the outgoing traffic when bandwidth constraints occur. In Qumran device, all the eight queues of ports will be configured with strict priority scheduling by default when QoS feature is enabled. Strict priority level will correspond to the queue number.

Scheduling algorithms are configurable per priority queues using queuing policy-map infrastructure. The following section describes how to configure different types of scheduling:

- Configuring strict priority based traffic scheduling
- Calculating the values for WFQ Weight based traffic scheduling
- Configuring WFQ weight based traffic scheduling
- Configuring mixed strict priority and weight based scheduling

Configuring Strict Priority Based Traffic Scheduling

Qumran supports strict priority algorithm with 8 levels from 0 to 7. When QoS feature is enabled, all 8 default queues will be initialized with strict priority scheduling with level corresponding to queue number. Strict priority scheduling is configurable using the priority command on the default queuing class-maps in a queuing policy-map. Configuration takes effect when the queuing policy-map is bound to the egress interface. By default, all interfaces are configured with “default-out-policy” when QoS feature is enabled.

Below, is the command to configure strict priority based scheduling:

```
priority level <0-7>
```

An example of configuring “strict priority” is shown below:

```
(config)#policy-map type queuing default pq7-strict
(config-pmap-que-def)#class type queuing default q7
(config-pmap-c-que-def)#priority level 6
(config-pmap-c-que-def)#exit
```

Queues set with higher priority value will have higher priority. If more than one queue is set with strict priority scheduling with same level value, then there will be fair queuing between those queues

Default scheduling algorithm in Qumran device is WFQ weight based traffic scheduling with weight 1. Therefore, when strict priority configuration is removed, scheduling algorithm on that queue will be reset to WFQ with default weight 1.

© 2026 IP Infusion Inc. Proprietary

Calculating the Values for WFQ Weight Based Traffic Scheduling

Weighted Fair Queueing (WFQ) scheduling weight is calculated as a percentage of the port's total bandwidth using the formula:

$$\text{Weight percentage of } Q(x) = \frac{WQ(x)}{WQ0 + WQ1 + WQ2 + WQ3 + WQ4 + WQ5 + WQ6 + WQ7}$$

```
Bandwidth percentage of Q(x) = Total bandwidth of port * weight percentage of Q(x)
```

Where:

WQ(x) is the value of the priority queue for which weight is to be determined.

WQ0-WQ7 is the assigned weight values of the eight queues – for example, if the weight values q0 to q7 are assigned as 3, 10, 5, 7, 18, 4, 9 and 1, then the Weight value of q4 can be calculated using the formula:

$$\text{Weight percentage of } q4 = \frac{18}{3 + 10 + 5 + 7 + 18 + 4 + 9 + 1}$$

The weight of q4 is 31.6%. Q4 will get 31.6% of the port's total bandwidth.

Note: Due to a difference in hardware rate set values, a deviation of (<4%) is expected.

The example below explains how to derive weights based on bandwidth requirement on queues.

If the bandwidth requirement of Q0 is 5%, Q1 is 10%, Q2 is 10%, Q3 is 15% and remaining bandwidth for Q4, Q5, Q6 and Q7, then the weight for individual queue can be computed as:

Q0: 5, Q1: 10, Q2: 10, Q3: 15, Q4-Q7: (100 - 5 - 10 - 10 - 15) = 60

Equally dividing weight 60 among 4 queues will result in a weight of 15 per queue. So, the weights will be:

Q0: 5, Q1: 10, Q2: 10, Q3: 15, Q4: 15, Q5: 15, Q6: 15, Q7: 15

In this case, weight values can be simplified by dividing them with common denominator 5. So, the final set of weight values for q0 to q7 will be derived to be 1, 2, 3, 3, 3, 3, 3 and 3.

Configuring WFQ Weight Based Traffic Scheduling

Set the WFQ weight based scheduling with the following configuration:

```
(config)#policy-map type queuing default pq0-3-wfq
(config-pmap-que-def)#class type queuing default q0
(config-pmap-c-que-def)# wfq-queue weight 16
(config-pmap-c-que-def)#exit
```

where:

```
wfq-queue weight <1-127>
```

Weight values can be configured in the range of 1 to 127.

Configuring Mixed Strict Priority and Weight Based Traffic Scheduling

Qumran device supports mixed scheduling option where strict priority with same or different level values can be configured on few queues along with WFQ algorithm with same or different weights configured on other default queues. Queues configured with strict priority scheduling will have a higher weight over the queues with WFQ scheduling.

An example configuration is shown below:

```
(config)#policy-map type queuing default pq-mixed
```

```
(config-pmap-que-def)#class type queuing default q0
(config-pmap-c-que-def)# wfq-queue weight 4
(config-pmap-c-que-def)#exit
(config-pmap-que-def)#class type queuing default q1
(config-pmap-c-que-def)#priority level 3
(config-pmap-c-que-def)#exit
(config-pmap-que-def)#class type queuing default q2
(config-pmap-c-que-def)# wfq-queue weight 10
(config-pmap-c-que-def)#exit
(config-pmap-que-def)#class type queuing default q3
(config-pmap-c-que-def)#priority level 4
(config-pmap-c-que-def)#exit
(config-pmap-que-def)#class type queuing default q4
(config-pmap-c-que-def)# wfq-queue weight 30
(config-pmap-c-que-def)#exit
```

Classes 5 to 7 will be strict priority with corresponding level if these classes were not altered.

Follow Chapter 13, *Configuring Egress Queues on Ports* for complete configuration details of queuing property configuration.

Scheduling may be affected by resource exhaustion in case of multicast traffic.

Resource exhausted can be verified through drop reason as "`RESOURCE_ERROR_STATUS`."

Queue Drop Counters Verification

Drop counters with drop reasons can be verified globally using the command:

```
#show hardware-discard-counters
+-----------------------------------+--------+--------+
| Registers                         | Core 0 | Core 1 |
+-----------------------------------+--------+--------+
IQM_QUEUE_ENQ_DISCARDED_PACKET_COUNTER        1596100
   Reason:  RESOURCE_ERROR_STATUS               Y
           VOQ_MX_QSZ_STATUS                    Y
EGQ_PQP_DISCARD_UNICAST_PACKET_COUNTER        59807
      Reason: SRC_EQUAL_DEST_INT                Y
```

CHAPTER 16 Egress Port and Priority Rate Shaping

Rate shaping is a method of regulating traffic rate to ensure a certain level of network performance. The difference between policing and rate shaping is that policing drops the excess traffic. Shaping buffers the excess traffic and thus insures a uniform pattern of traffic egressing. Shaping is required when the nature of traffic is busty and needs to be smoothened.

Qumran supports configuring rate shaping per priority queue or per port.

Configuring Port-based Rate Shaping

With Port based rate shaping, total traffic can be limited to the shape rate within the limits of port bandwidth. Qumran supports per-port rate shaping configuration within a range of 52kbps to 1000gbps

Refer to Table 16-9 for shape rate granularity.

Table 16-9: Port-based shape rate granularity

Platform	Granularity in kbps
Q1 (QMX,QAX,QUX)	52
Q2A	515
Q2C/J2C	386

Note: Port based shaping is supported only on physical interfaces.

```
(config)#interface xe1
(config-if)#shape rate 200 mbps
(config-if)#exit
```

The general syntax is:

```
shape rate <1-1000000000> (kbps|mbps|gbps)
```

Configuring Priority-based Rate Shaping

Priority based rate shaping ensures a traffic shaping per priority queue traffic. Priority based rate shaping is configured on the queuing class-map representing priority queue. Shaping can be configured in absolute value or in percentage of bandwidth. Qumran supports per-queue rate shaping configuration within a range of 469kbps to 483gbps.

Refer to Table 16-10 for shape rate granularity.

Table 16-10: Priority-based shape rate granularity

Platform	Granularity in kbps
Q1 (QMX,QAX,QUX)	469 for lower ranges 1.56% for higher ranges.
J2C/Q2C	2604
Q2A	1562

© 2026 IP Infusion Inc. Proprietary

The following example shows a sample configuration of priority based rate shaping:

```
(config)#policy-map type queuing default pq
(config-pmap-que-def)#class type queuing default q3
(config-pmap-c-que-def)#shape 10 mbps
(config-pmap-c-que-def)#exit
```

The general syntax is:

```
shape <1-483000000> (kbps|mbps|gbps) | percent <1-100>)
```

See Chapter 13, *Configuring Egress Queues on Ports* for details about queuing property configuration.

Queue drop counters verification

Drop counters with drop reasons can be verified globally by using the command:

```
#show hardware-discard-counters
+-----------------------------------+--------+--------+
| Registers                         | Core 0 | Core 1 |
+-----------------------------------+--------+--------+
IQM_QUEUE_ENQ_DISCARDED_PACKET_COUNTER        1596100
   Reason: VOQ_MX_QSZ_STATUS                   Y
EGQ_PQP_DISCARD_UNICAST_PACKET_COUNTER        59807
      Reason: SRC_EQUAL_DEST_INT                  Y
```

CHAPTER 17 Display Queuing Information

The following queuing information can be displayed:

- Queuing Configuration Information
- Queuing Packet and Byte Statistics

Displaying queuing configuration information

The queuing configurations that can be displayed are:

- Policy-map configuration
- Interface level queuing configuration

Display Policy-map Configuration

Use the following command to verify configurations on policy-map:

```
show policy-map (type queuing|NAME|)
```

`NAME` is an optional parameter which allows verifying a particular policy-map. Qumran supports only default queuing policy-maps.

Sample output for queuing policy-map configuration is shown below:

```
#show policy-map pq1

Type queuing policy-maps
========================

policy-map type queuing default pq1
 class type queuing default q0
  shape percent 60
  wfq-queue weight 20
  exit
 class type queuing default q1
  shape percent 20
  wfq-queue weight 50
  exit
 class type queuing default q2
  priority level 2
  exit
 class type queuing default q3
  priority level 3
  exit
 class type queuing default q4
  priority level 4
  exit
 class type queuing default q5
  priority level 5
  exit
 class type queuing default q6
  priority level 6
```

```
  exit
 class type queuing default q7
  priority level 7
  exit
```

Interface level queuing configuration

The following command shows the configuration on eight priority queues of an interface:

```
show queuing interface INTERFACE-NAME
```

Sample output of interface based queue configuration is shown below:

```
#show queuing interface xe1

Egress Queuing for Ethernet xe1 [System]
------------------------------------------------------
L0   L1   L2   Group    PrioLevel  Shape      Bandwidth
------------------------------------------------------
q0             -        -          60 percent     -
q1             -        -          20 percent     -
q2             -        High       -              -
q3             -        High       -              -
q4             -        High       -              -
q5             -        High       -              -
q6             -        High       -              -
q7             -        High       -              -
```

Display Type Queuing Policy-map Configuration and Statistics on an Interface

Type queuing policy-map configurations and statistics can be displayed using the command:

```
show policy-map interface INTERFACE-NAME type queuing
```

Sample output is shown below:

```
#show policy-map interface xe0 type queuing

Interface xe0
Type Queuing policy-map : pq1

Class-map (queuing): class-default-q
 shape 1000000 kbps (inherited)
 wfq-queue weight 1
 queue-limit 1253376 bytes/10 ms (default)
 match queue 0
      Output
         Total        : 14303226 packets, 21454839000 bytes
         Green        : 14303226 packets, 21454839000 bytes
         Yellow       : 0 packets, 0 bytes
 match queue 1
 match queue 2
 match queue 3
 match queue 4
```

© 2026 IP Infusion Inc. Proprietary

```
 match queue 5
 match queue 6
 match queue 7


Class-map (queuing): service3
 match service-template v3
 shape 1000000 kbps (inherited)
 wfq-queue weight 1

     Class-map (queuing): class-default-q
      shape 1000000 kbps (inherited)
      wfq-queue weight 1

     Class-map (queuing): data
      shape 1000000 kbps (inherited)
      wfq-queue weight 1
      queue-limit 1253376 bytes/10 ms (default)
      match queue 0
      match queue 1
           Output
              Total        : 4717105 packets, 7075660500 bytes
              Green        : 4717122 packets, 7075684500 bytes
              Yellow       : 0 packets, 0 bytes
           Rate            : 118443.625 kbps
      match queue 2
```

Configurations and statistics of a particular class in the type queuing policy-map on an interface can be displayed using the command `show policy-map interface INTERFACE-NAME (class NAME|) type queuing`

Sample output is shown below:

```
#show policy-map interface xe2 class service3

Interface xe2
Type Queuing policy-map : pq1

Class-map (queuing): service3
 match service-template v3
 shape 1000000 kbps (inherited)
 wfq-queue weight 1

     Class-map (queuing): class-default-q
      shape 1000000 kbps (inherited)
      wfq-queue weight 1
      queue-limit 1253376 bytes/10 ms (default)
      match queue 2
           Output
              Total        : 111200 packets, 166801500 bytes
              Green        : 111209 packets, 166813500 bytes
              Yellow       : 0 packets, 0 bytes
```

```
      Rate            : 59141.176 kbps
 match queue 3

Class-map (queuing): data
 shape 1000000 kbps (inherited)
 wfq-queue weight 1
 queue-limit 1253376 bytes/10 ms (default)
 match queue 0
 match queue 1
      Output
         Total        : 222450 packets, 333678000 bytes
         Green        : 222466 packets, 333702000 bytes
         Yellow       : 0 packets, 0 bytes
      Rate            : 118287.391 kbps
```

CHAPTER 18 Display Queue Level Packet and Byte Counters

Queue level statistics can be displayed and cleared as described in the following section:

- Displaying queuing class configuration and queue packet and byte counters
- Clearing queue level packet and byte counters

Queue level statistics can be displayed using the command:

Display queue level statistics using the commands:

show policy-map, show policy-map interface, and show interface IFNAME counters queue-stats.

Sample output is shown below:

```
#show policy-map statistics type queuing
+-----------------+-----------------+-----------------+-----------------+---------------+
|     Class-map   |    Total pkts    |   Total bytes    |   Dropped pkts   |   Dropped Bytes
|
+-----------------+-----------------+-----------------+-----------------+---------------+
xe0
 q1                1142              1713000           133978            200965500
 q3                1138              1707000           66451             99675000
xe2
 q1                1133              1699500           134476            201715500
 q2                1155              1732500           66655             99984000
#show policy-map statistics class q1 type queuing
+-----------------+-----------------+-----------------+-----------------+---------------+
|     Class-map   |    Total pkts    |   Total bytes    |   Dropped pkts   |   Dropped Bytes
|
+-----------------+-----------------+-----------------+-----------------+---------------+
xe0
 q1                5110              7665000           598853            898278000
xe2
 q1                5084              7627500           598821            898231500
#show policy-map int xe0 statistics type queuing
+-----------------+-----------------+-----------------+-----------------+---------------+
|     Class-map   |    Total pkts    |   Total bytes    |   Dropped pkts   |   Dropped Bytes
|
+-----------------+-----------------+-----------------+-----------------+---------------+
 q1                5943              8914500           697242            1045861500
 q3                5925              8887500           345696            518542500
#show policy-map int xe0 statistics class q1 type queuing
+-----------------+-----------------+-----------------+-----------------+---------------+
|     Class-map   |    Total pkts    |   Total bytes    |   Dropped pkts   |   Dropped Bytes
|
+-----------------+-----------------+-----------------+-----------------+---------------+
 q1                7314              10971000          858974            1288461000
```

Sample output for interface based queue statistics is shown below:

```
#show interface xe1 counters queue-stats
E - Egress, I - Ingress, Q-Size is in bytes
+---+------+--------+---------+---------+-----------+
```

```
| Q | Q-Sz | Tx pkt | Tx byte | Drp pkt | Drop byte |
+---+------+--------+---------+---------+-----------+
q0   629160 100      12000       0         0
q0   629160 0        0           0         0
q0   629160 0        0           0         0
q0   629160 0        0           0         0
q0   629160 0        0           0         0
q0   629160 0        0           0         0
```

Display Queue Level Instantaneous Transmission Rate

The instantaneous rate at which packets are de-queued/transmitted from the queues can be displayed using

```
show policy-map (interface INTERFACE-NAME|) statistics (class CLASS-NAME|) type queuing
(rate (kbps|mbps|gbps)|)
```

Sample output of the command is shown below:

```
#show policy-map statistics type queuing rate mbps
+-------------------------+-----------------+
|             Class-map    | Rate (in mbps) |
+-------------------------+-----------------+
xe0
 q1                                1.031
 q3                                1.031
xe2
 q1                                1.031
 q2                                0.824


#show policy-map interface xe0 statistics type queuing rate mbps
+-------------------------+----------------+
|             Class-map    | Rate (in mbps)|
+-------------------------+----------------+
 q1                                0.700
 q3                                1.168


#show policy-map interface xe0 statistics class q1 type queuing rate mbps
+------------------------+-----------------+
|             Class-map   | Rate (in mbps) |
+------------------------+-----------------+
 q1                                0.937
```

Clearing Queue Level Packet and Byte Counters

Queue statistics can be cleared using the command clear qos statistics and clear interface counters.

© 2026 IP Infusion Inc. Proprietary

CHAPTER 19 VLAN Service Queuing (VLAN Shaping)

Each port in a Qumran device can use eight default priority queues.Enabling the QoS feature configures all port priority queues with default egress queuing parameters. To customize the priority queue treatment, use the queuing policy-map infrastructure.

Service queuing maps services to specific VLANs and shapes each VLAN based traffic. Within each VLAN, queues can be grouped and shaped independently.

With service queuing, OcNOS supports up to three levels of hierarchical queuing (HQoS). The port default queues continue to operate with a single-level scheduler.

Configuring VLAN Shaping

The following section explains how to configure basic infrastructure functionality for queuing per services on an interface. These queues support all the possible QoS treatments through egress queuing policy-map configurations. Services can be mapped using a `service-template` or match vlan commands. When a service in a class is matched inside a policy and attached to an interface, four new queues are created for these services. Users can create a maximum 3-level scheduling hierarchy for each service using these policy-maps.

A policy-map attached to an interface is referred to as an `L0` level policy-map. Each child policy-map added is at an incremented level, i.e., the child of an L0-level policy-map is an L1-level policy-map, and the child of an L1-level is an L2 policy-map. This user-defined policy-map supports a maximum 3-level hierarchy.

Class-default-q is a self-created class map within a policy map. There are two types of class-default-q class maps (CMAPs):

- When a policy is applied to an interface, class-default-q represents the port default queues. If the operator wants to shape or apply TailDrop or Weighted Random Early Detection (WRED) properties on the port queues, they must apply them to the class-default-q CMAP at Level 0.
- When a child policy is attached to a CMAP with a service match criterion (match service-template/VLAN), the class-default-q CMAP in the child policy represents the queues left unmatched in the rest of the CMAPs of the child policy.

Configuring a Queuing Policy-map

When the QoS feature is enabled, the Qumran device applies a default policy-map of queuing type to all ports. This default policy-map, named `default-out-policy`, is reserved, and modifying its parameters affects all ports without a customized queuing policy-map. Users can create customized queuing policy-maps and bind them to ports to treat ports differently from default configuration.

Creating a user-defined queuing policy-map

Qumran board allows the creation of customized policy-maps, where users can configure four new queues for each service. If a priority queue class-map is not configured, the default behavior for these new queues is weighted fair queuing. Weighted fair queuing is also the default behavior between new queues and port default queues.

Use the command class-map type queuing (match-any|match-all|) NAME to create a class-map.

These class-maps match services for which new queues will be created. Users can also create empty class-maps to establish hierarchies and group services accordingly.

Matching criteria of a queuing class-map

A queuing class-map can use four matching criteria:

1. **Queue:** Selects the queue. Use the command `match queue <0-7>`.

Note:

- For port queues, the valid range is 0-7.
- For service queues, the valid range is 0-3.

2. **Service-template:** Classifies based on service. Use the command `match service-template NAME` and `no match service-template`.

3. **VLAN:** Lists VLAN IDs.Use the command match vlan

Note: VLAN matching criteria support applies to customer VLANs on the CE port in Provider Bridge only.

4. **Interface:** Matches based on interface name. Use the commands `match interface IFNAME` and `no match interface`.

Note: The interface name can be a VLAN interface or a sub-interface.

To create a customized policy-map, use the command policy-map {NAME | (type (queuing|queuing default) NAME)}.

After configuring the policy-map, set up the queuing class-maps using the command class type queuing.

Binding/Unbinding a QoS Policy-map on an Interface

Customized queuing policy-maps take affect only when the configuration is bound to a port. To bind or unbind a queuing policy-map to an out-port, use the command service-policy type queuing.

Binding/Unbinding a QoS Policy-map as a Child Service Policy

To create a hierarchy, configure a policy-map as a child policy. Bind a queuing policy-map to a policy-map as a child policy with the command `(no|) service-policy NAME`. Here, `NAME` represents the name of the queuing policy-map.

Note: Attach the child policy to a parent policy-map within a class.

Note: Match all the required classes to the child policy map to police the traffic according to the parent policy map policer.

For example:

```
class-map type queuing data
 match queue 0
!
class-map type queuing service1
 match service-template ETH-2016
!
class-map type queuing service2
 match service-template ETH-2017
!
class-map type queuing signal
 match queue 3
!
class-map type queuing voice
 match queue 1
```

```
!
policy-map type queuing configPolicy1
 class type queuing class-default-q
  exit
 class type queuing data
  exit
 class type queuing signal
  exit
 class type queuing voice
  exit
!
policy-map type queuing customer1
 class type queuing service1
  service-policy configPolicy1
class type queuing class-default-q
  exit
!
interface xe11
 service-policy type queuing output customer1
```

Here, “customer1” is L0-level policy-map and “configPolicy1” is L1-level policy-map. Policy-map “customer1” is having a class-default-q which is having port default queues. Policy-map “configPolicy1” is having class-default-q which is representing remaining queue i.e. queue 2 as queue3 is mapped to class signal, queue1 is mapped to class voice and queue0 is mapped to class data.

WRED and taildrop configuration is applicable only in the class in which queues are mapped. If in a policy-map having a class matching the service is not having any child policy-map, then all the new queues will be mapped to the same class and WRED and taildrop configuration is valid for this class. If the class matching service is having child policy matching queues, then WRED and taildrop is valid for the child service-policy only.

Policy-map having classes matching the queues can only be configured as a child service-policy inside a class matching service or at L0 class-default-q. It cannot be attached on an interface directly

If the user-defined child service policy is applied matching queues in L0 class-default-q which are mapping port queues, supported match queue range is 0-7.

For service queues, valid range is 0-3 as only 4 new queues are created for each service.Since the queues created are 4, 8 traffic classes are mapped implicitly to 4 queues as shown in Table 19-11.

Table 19-11: Traffic class to queue mapping

Traffic class	Queue
0	0
1, 2, 3	1
4, 5	2
6, 7	3

Ingress mapping profile like cos-to-queue, dscp-to-queue, and exp-to-queue actually maps packet fields (cos/dscp/exp) to 8 traffic classes. These traffic classes are mapped one-to-one when we have 8 queues in case of physical port and to 4 queues in case of services as shown above.

Until the child service-policy is applied on L0 class-default-q (port queues), port queues will follow default mapping profiles.

QoS feature must be enabled to configure policy-maps. This infrastructure contains two entities - class-map and policy-map. Class-map holds the match criteria and class-maps can be bound to policy-map to configure QoS treatment for the matching traffic.

The following section explains the basic configuration details involved to apply queue level treatment on the port.

Note: Please refer to *MPLS configuration guide* for service-template configurations.

```
(config)# class-map type queuing customer1
(config-cmap-que)# match service-template customer1Vlan
(config-cmap-que)#exit
(config)# class-map type queuing customer2
(config-cmap-que)# match service-template customer2Vlan
(config-cmap-que)#exit
(config)# class-map type queuing customer3
(config-cmap-que)# match service-template customer3Vlan
(config-cmap-que)#exit
(config)# class-map type queuing customer4
(config-cmap-que)# match service-template customer4Vlan
(config-cmap-que)#exit

(config)#class-map type queuing data
(config-cmap-que)# match queue 0
(config-cmap-que)# exit
(config)#class-map type queuing voice
(config-cmap-que)# match queue 1
(config-cmap-que)#exit
(config)# class-map type queuing signaling
(config-cmap-que)# match queue 3
(config-cmap-que)# exit

(config)#class-map type queuing area1
(config-cmap-que)# exit
(config)#class-map type queuing area2
(config-cmap-que)# exit

(config)#policy-map type queuing traffic_policy
(config-pmap-que)# class type queuing data
(config-pmap-c-que)# shape 10 mbps
(config-pmap-c-que)# exit
(config-pmap-que)# class type queuing voice
(config-pmap-c-que)# shape 2 mbps
(config-pmap-que)# class type queuing signaling
(config-pmap-c-que)# shape 1 mbps
(config-pmap-c-que)# exit

(config-pmap-que)#policy-map type queuing area1_policy
(config-pmap-que)# class type queuing customer1
(config-pmap-c-que)# shape 12 mbps
(config-pmap-c-que)# service-policy traffic_policy
(config-pmap-c-que)# exit
(config-pmap-que)# class type queuing customer2
(config-pmap-c-que)# shape 12 mbps
(config-pmap-c-que)# exit

(config-pmap-que)#policy-map type queuing area2_policy
```

```
(config-pmap-que)# class type queuing customer3
(config-pmap-c-que)# shape 12 mbps
(config-pmap-c-que)# service-policy traffic_policy
(config-pmap-c-que)# exit
(config-pmap-que)# class type queuing customer4
(config-pmap-c-que)# shape 12 mbps
(config-pmap-c-que)# exit

(config-pmap-que)#policy-map type queuing xe16_policy

(config-pmap-que)# class type queuing area1
(config-pmap-c-que)# shape 100 mbps
(config-pmap-c-que)# service-policy area1_policy
(config-pmap-c-que)# exit
(config-pmap-que)# class type queuing area2
(config-pmap-c-que)# shape 100 mbps
(config-pmap-c-que)# service-policy area2_policy
(config-pmap-c-que)# exit

(config-pmap-que)#interface xe16
(config-if)# service-policy type queuing output xe16_policy
(config-if)#exit
```

Displaying Policy-map Configuration

The following is an example of `show policy-map interface` command:

```
(config)#show policy-map interface xe16

Interface xe16
Type QoS statistics status : enabled

Service-policy (queuing) output: xe16_policy
Interface Bandwidth 1000000 kbps
-----------------------------------
Class-map (queuing): area1
 shape 100 mbps
 wfq-queue weight 1

     Service-policy (queuing) output: area1_policy
     -----------------------------------
     Class-map (queuing): customer1
      match service-template customer1Vlan
      shape 12 mbps
      wfq-queue weight 1

          Service-policy (queuing) output: traffic_policy
          -----------------------------------
          Class-map (queuing): class-default-q
           shape 12000 kbps (inherited)
           wfq-queue weight 1
           queue-limit 15040 bytes/10 ms (default)
```

```
           match queue 1

          Class-map (queuing): data
           shape 10 mbps
           wfq-queue weight 1
           queue-limit 12544 bytes/10 ms (default)
           match queue 0
           Output
         Total        : 66681 packets, 66681000 bytes
         Green        : 66681 packets, 66681000 bytes
         Yellow       : 0 packets, 0 bytes

          Class-map (queuing): signaling
           shape 1 mbps
           wfq-queue weight 1
           queue-limit 9472 bytes/76 ms (default)
           match queue 3

          Class-map (queuing): voice
           shape 2 mbps
           wfq-queue weight 1

           queue-limit 9472 bytes/38 ms (default)
           match queue 2
     Class-map (queuing): customer2
      match service-template customer2Vlan
      shape 12 mbps
      wfq-queue weight 1
      queue-limit 15040 bytes/10 ms (default)
      match queue 0
      match queue 1
      match queue 2
      match queue 3

Class-map (queuing): area2
 shape 100 mbps
 wfq-queue weight 1

     Service-policy (queuing) output: area2_policy
     -----------------------------------
     Class-map (queuing): customer3
      match service-template customer3Vlan
      shape 12 mbps
      wfq-queue weight 1
      queue-limit 15040 bytes/10 ms (default)
      match queue 0
      match queue 1
      match queue 2
      match queue 3
```

```
     Class-map (queuing): customer4
      match service-template customer4Vlan
      shape 12 mbps
      wfq-queue weight 1

          Service-policy (queuing) output: traffic_policy
          -----------------------------------
          Class-map (queuing): class-default-q
           shape 12000 kbps (inherited)
           wfq-queue weight 1
           queue-limit 15040 bytes/10 ms (default)
           match queue 1

          Class-map (queuing): data

           shape 10 mbps
           wfq-queue weight 1
           queue-limit 12544 bytes/10 ms (default)
           match queue 0
          Class-map (queuing): signaling
           shape 1 mbps
           wfq-queue weight 1
           queue-limit 9472 bytes/76 ms (default)
           match queue 3

          Class-map (queuing): voice
           shape 2 mbps
           wfq-queue weight 1
           queue-limit 9472 bytes/38 ms (default)
           match queue 2

Class-map (queuing): class-default-q
 shape 1000000 kbps (inherited)
 wfq-queue weight 1
 queue-limit 1253376 bytes/10 ms (default)
 match queue 0
 match queue 1
 match queue 2
 match queue 3
 match queue 4
 match queue 5
 match queue 6
 match queue 7
```

QoS Configuration on User-defined Policy-map

All the queuing configurations such as WRED, taildrop, WFQ, shaping are same for user-defined policy as they are in default-policy-map except the priority queue configuration.

In default-policy-map, max priority supported is 8 i.e. 0-7, while in user-defined policy-map, max priority level is 4, i.e.0-3.

Priority class will always have priority over weighted class in default policy. But in user-defined policy, when all the 4 priorities are assigned with weighted classes, priority 0 class will be in fair queuing with the total weighted queues. If 3 or less than 3 priority class are present with weighted classes, than priority class will have priority over weighted class.

For example:

```
(config-pmap-que)#policy-map type queuing area2_policy
 (config-pmap-que)# class type queuing customer1
 (config-pmap-c-que)# priority level 0
 (config-pmap-c-que)# service-policy traffic_policy
 (config-pmap-que)# class type queuing customer2
 (config-pmap-c-que)# priority level 1
 (config-pmap-c-que)# service-policy traffic_policy
 (config-pmap-que)# class type queuing customer3
 (config-pmap-c-que)# priority level 2
 (config-pmap-c-que)# service-policy traffic_policy
 (config-pmap-que)# class type queuing customer4
 (config-pmap-c-que)# priority level 3
 (config-pmap-c-que)# service-policy traffic_policy
 (config-pmap-que)# class type queuing customer5
 (config-pmap-c-que)# wfq-queue weight 1
 (config-pmap-c-que)# service-policy traffic_policy
  (config-pmap-que)# class type queuing customer6
 (config-pmap-c-que)# wfq-queue weight 2
 (config-pmap-c-que)# service-policy traffic_policy
```

In this case, customer4 will have the highest priority, while 50% of the remaining bandwidth after distributing among priority classes will be used by Customer1, and the remaining 50% will be shared by Customer5 and Customer6 (FQ between priority 0 and weighted class).

If there are only three priority classes, for example:

```
(config-pmap-que)#policy-map type queuing area2_policy
(config-pmap-que)# class type queuing customer1
(config-pmap-c-que)# priority level 0
(config-pmap-c-que)# service-policy traffic_policy
(config-pmap-que)# class type queuing customer2
(config-pmap-c-que)# priority level 1
(config-pmap-c-que)# service-policy traffic_policy
(config-pmap-que)# class type queuing customer3
(config-pmap-c-que)# priority level 2
(config-pmap-c-que)# service-policy traffic_policy
(config-pmap-que)# class type queuing customer4
(config-pmap-c-que)# wfq-queue weight 1
(config-pmap-c-que)# service-policy traffic_policy
(config-pmap-que)# class type queuing customer5
(config-pmap-c-que)# wfq-queue weight 2
(config-pmap-c-que)# service-policy traffic_policy
```

Here, Customer3 has the highest priority. Customer1 has priority 0 and will have priority over Custimer4 and Customer5.

© 2026 IP Infusion Inc. Proprietary

Configuration Considerations

- Max 3 level of user defined hierarchy is supported.
- Class-default-q is a self-created class map as part a policy map. It cannot be created nor be destroyed. It will be displayed (on whichever level applicable) only when user access it. Executing command "no class-default-q", will un-configure all the configurations of class-default-q.
- User can configure all queuing parameters like weight, priority, queue-limit, wred and shape in a class inside policy.
- Queue-limit and wred is only applicable in class matching queues.
- Same service should not be matched twice in the same hierarchy at any level.
- Policy-map with classes matching queues can only be attached to the policy-map having classes with match service or L0 class-default-q class.
- Child service policy is not allowed in class matching queues unless it is L0 class-default-q.
- If a service-policy is configured as a child policy in any hierarchy, then it cannot be attached on an interface directly and vice-versa.
- All the classes inside a policy-map should have same matching criteria but not same matching criteria value.
- Class-default-q class will only be present at L0 level or at the last level in a hierarchy. Class-default-q class will be matching port default queues if the class is of L0. Class-default-a class will be matching remaining queues in the newly created queue bundle if not matched in a class.
- Valid match queue range for classifying port-queues is 0-7. For service queue it is 0-3.
- Update is possible in the policy-map except the update of match criteria. Once the class with some match criteria is used in a policy-map, it cannot be updated.
- Max 4 priority queues are supported in non-default queuing policy-map.
- VLAN shaping is only supported for L2VPN, L3VPN, and provider-bridging services.
- Attaching user-defined policy will impact the ongoing traffic which may leads to session flaps if any, configured on that interface.

Service Queuing refers to mapping services to specific vlans and shaping each vlan based traffic. Within the vlan, queues can be grouped and shaped independently.

Matching of traffic can be based on different parameters such as service-template, VLAN, sub interface/VLAN interface.

Below, are different configuration for L2VPN, VPLS/VPWS services, L3VPN with sub and VLAN interfaces, and a provider bridge configuration.

L2VPN-VPLS

Figure 19-3 displays a six node topology configured with end to end connectivity from Router 1 to Router 6. The end to end connectivity is established by configuring OSPF, iBGP and LDP configuration in all the routers. We should be able to ping each device from other device from topology. Configure L2VPN- VPLS services on RTR1 and RTR6 and create Qos configuration on RTR6 and verify service queuing.

© 2026 IP Infusion Inc. Proprietary

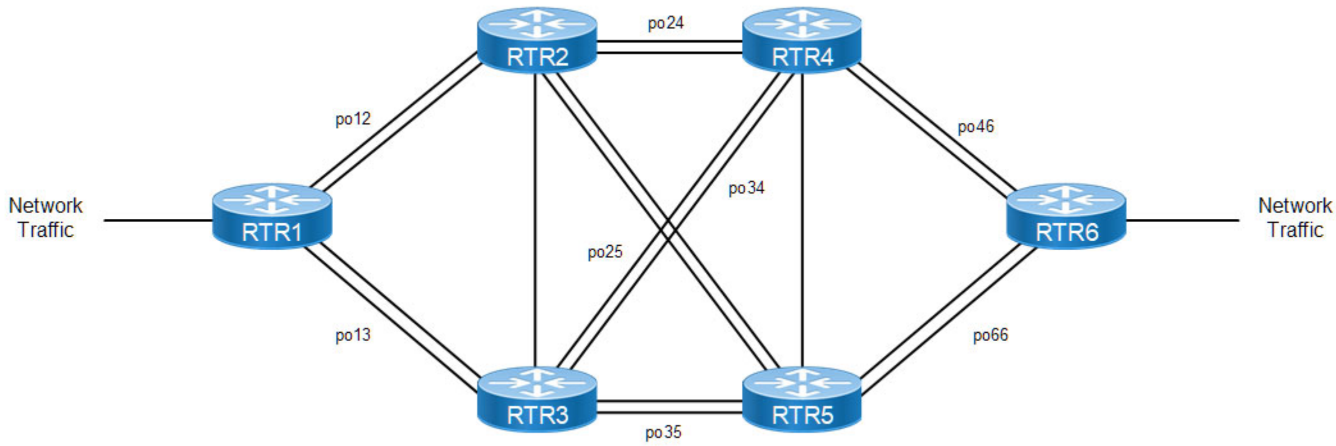


Figure 19-3: Simple Topology

Configure Sub-interface on RTR1

1. Create VPLS Instance V30.

```
RTR1(config)#mpls vpls V30 30
RTR1(config-vpls)#service-tpid dot1.ad
RTR1(config-vpls)#signaling ldp
RTR1(config-vpls-sig)#vpls-type vlan
RTR1(config-vpls-sig)#vpls-peer 6.6.6.6
RTR1(config-vpls-sig)#exit-signaling
RTR1(config-vpls)#exit
```

2. Create VPLS Instance V40.

```
RTR1(config)#mpls vpls V40 40
RTR1(config-vpls)#service-tpid dot1.ad
RTR1(config-vpls)#signaling ldp
RTR1(config-vpls-sig)#vpls-type vlan
RTR1(config-vpls-sig)#vpls-peer 6.6.6.6
RTR1(config-vpls-sig)#exit-signaling
RTR1(config-vpls)#exit
```

3. Configure sub-interface xe13.101.

```
RTR1(config)#interface xe13.101 switchport
RTR1(config-if)#load-interval 30
RTR1(config-if)#encapsulation dot1q 30 inner-dot1q 2030
RTR1(config-if)#rewrite pop
RTR1(config-if)#access-if-vpls
RTR1(config_if_vpls)#mpls-vpls V30
RTR1(config_if_vpls)#exit-if-vpls
```

4. Configure sub-interface xe13.102.

```
RTR1(config)#interface xe13.102 switchport
RTR1(config-if)#load-interval 30
RTR1(config-if)#encapsulation dot1q 40 inner-dot1q 2040
RTR1(config-if)#rewrite pop
RTR1(config-if)#access-if-vpls
RTR1(config_if_vpls)#mpls-vpls V40
RTR1(config_if_vpls)#exit-if-vpls
```

Configure Sub-interface on RTR6

1. Create VPLS instance V30.

```
RTR6(config)#mpls vpls V30 30
RTR6(config-vpls)#service-tpid dot1.ad
RTR6(config-vpls)#signaling ldp
RTR6(config-vpls-sig)#vpls-type vlan
RTR6(config-vpls-sig)#vpls-peer 1.1.1.1
RTR6(config-vpls-sig)#exit-signaling
RTR6(config-vpls)#exit
```

2. Create VPLS instance V40.

```
RTR6(config)#mpls vpls V40 40
RTR6(config-vpls)#service-tpid dot1.ad
RTR6(config-vpls)#signaling ldp
RTR6(config-vpls-sig)#vpls-type vlan
RTR6(config-vpls-sig)#vpls-peer 1.1.1.1
RTR6(config-vpls-sig)#exit-signaling
RTR6(config-vpls)#exit
```

3. Configure sub-interface xe13.102.

```
RTR1(config)#interface xe13.102 switchport
RTR1(config-if)#load-interval 30
RTR1(config-if)#encapsulation dot1q 40 inner-dot1q 2040
RTR1(config_if)#rewrite pop
RTR1(config-if)#access-if-vpls
RTR1(config_if_vpls)#mpls-vpls V40
```

4. Configure class maps for queuing.

```
RTR6(config)#class-map type queuing que0
RTR6(config-cmap-que)#match queue 0
RTR6(config-cmap-que)#class-map type queuing que1
RTR6(config-cmap-que)#match queue 1
RTR6(config-cmap-que)#class-map type queuing que2
RTR6(config-cmap-que)#match queue 2
RTR6(config-cmap-que)#class-map type queuing que3
RTR6(config-cmap-que)#match queue 3
RTR6(config-cmap-que)#class-map type queuing service30
RTR6(config-cmap-que)#match service-template VPLS-30
RTR6(config-cmap-que)#class-map type queuing service40
RTR6(config-cmap-que)#match service-template VPLS-40
```

5. Create and configure policy maps.

```
RTR6(config)#policy-map type queuing queue
RTR6(config-pmap-que)#class type queuing que0
RTR6(config-pmap-c-que)#exit
RTR6(config-pmap-que)#class type queuing que1
RTR6(config-pmap-que)#shape 100 mbps
RTR6(config-pmap-c-que)#exit
RTR6(config-pmap-que)#class type queuing que2
RTR6(config-pmap-que)#shape 100 mbps
RTR6(config-pmap-c-que)#exit
RTR6(config-pmap-que)#class type queuing que3
RTR6(config-pmap-c-que)#exit
RTR6(config-pmap-que)#policy-map type queuing service
RTR6(config-pmap-que)#class type queuing service30
RTR6(config-pmap-que)#shape 1000 mbps
```

```
RTR6(config-pmap-c-que)#service-policy queue
RTR6(config-pmap-c-que)#exit
RTR6(config-pmap-que)#class type queuing service40
RTR6(config-pmap-c-que)#service-policy queue
RTR6(config-pmap-c-que)#exit
```

6. Configure interface xe13.

```
RTR6(config)#interface xe13
RTR6(config-if)#service-policy type queuing output service
RTR6(config-if)#shape rate 3000 mbps
RTR6(config-if)#exit
```

Validation

RTR6

```
RTR6#show policy-map statistics type queuing rate mbps
+--------------------------------++
|  Class-map|Rate (in mbps) |
+--------------------------------++
xe13
q0  956.796
q2  1032.543
q4  1044.794
q6  1030.280
RTR6#
RTR6#show running-config qos qos enable
!
class-map type queuing que0 match queue 0
!
class-map type queuing que1 match queue 1
!
class-map type queuing que2 match queue 2
!
class-map type queuing que3 match queue 3
!
class-map type queuing service30 match service-template VPLS-30
!
class-map type queuing service40 match service-template VPLS-40
!
!
policy-map type queuing queue class type queuing que0
exit
class type queuing que1 exit
class type queuing que2 exit
class type queuing que3 exit
!
policy-map type queuing service class type queuing service30
service-policy queue


exit
class type queuing service40 service-policy queue
```

```
exit
!
interface xe13
service-policy type queuing output service
!

RTR6#show policy-map statistics type queuing rate mbps
+--------------------------------++
|  Class-map|Rate (in mbps) |
+--------------------------------++

que0
(q0)
510.806
que1
(q1)
514.803
que2
(q2)
507.316
que3
(q3)
521.820
ervice40
que0 (q0)438.860
que1
(q1)
530.446
que2
(q2)
518.338
que3
(q3)
517.863
```

L2VPN- VPWS

Figure 19-4 displays a six node topology configured ith end to end connectivity from Router 1 to Router 6. The end to end connectivity is established by configuring OSPF, iBGP and LDP configuration in all the routers. We should be able to ping each device from other device from topology. Configure L2VPN- VPWS services on RTR1 and RTR6 and create Qos configuration on RTR6 and verify service queuing.

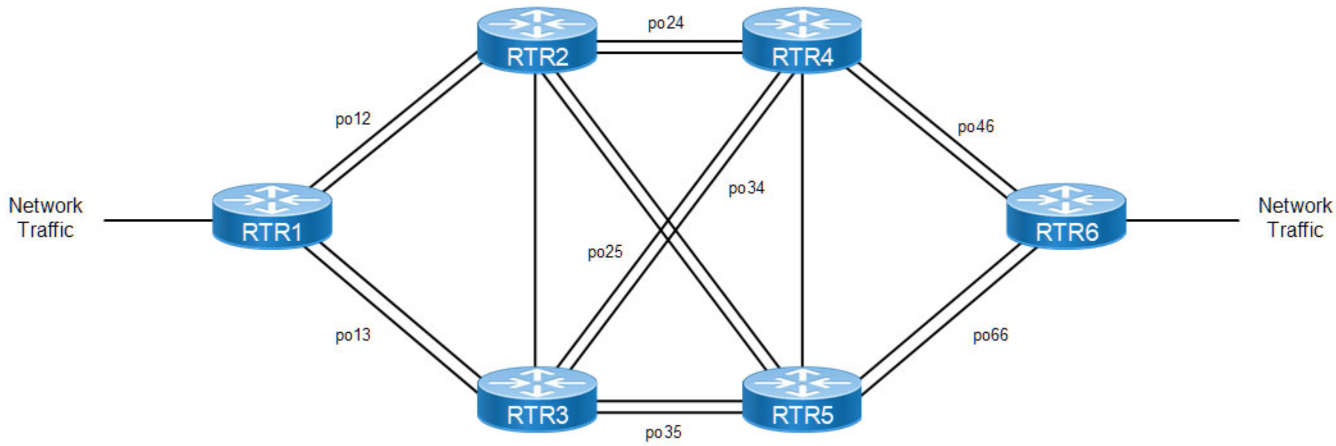


Figure 19-4: Simple Topology

Configure Sub-interface on RTR1

1. Create MPLS L2 circuits.
   - For VPWS10:

```
RTR1(config)#mpls l2-circuit VPWS10 10 6.6.6.6
RTR1(config-pseudowire)#exit
```

   - For VPWS20:

```
RTR1(config)#mpls l2-circuit VPWS20 20 6.6.6.6
RTR1(config-pseudowire)#exit
```

2. Configure sub-interface xe14.101.

```
RTR1(config)#interface xe14.101 switchport
RTR1(config-if)#encapsulation dot1q 101
RTR1(config-if)#load-interval 30
RTR1(config-if)#access-if-vpws
RTR1(config-acc-if-vpws)#mpls-l2-circuit VPWS10 primary
RTR1(config-acc-if-vpws)#exit
```

3. Configure sub-interface xe14.102.

```
RTR1(config)#interface xe14.102 switchport
RTR1(config-if)#encapsulation dot1q 102
RTR1(config-if)#load-interval 30
RTR1(config-if)#access-if-vpws
RTR1(config-acc-if-vpws)#mpls-l2-circuit VPWS20 primary
RTR1(config-acc-if-vpws)#exit
```

Configure Sub-interface on RTR6

1. Create MPLS L2 circuits.
   - For VPWS10:

```
RTR6(config)#mpls l2-circuit VPWS10 10 1.1.1.1
RTR6(config-pseudowire)#exit
```

   - For VPWS20:

```
RTR6(config)#mpls l2-circuit VPWS20 20 1.1.1.1
RTR6(config-pseudowire)#exit
```

2. Configure sub-interface xe14.101.

```
RTR6(config)#interface xe14.101 switchport
```

```
RTR6(config-if)#encapsulation dot1q 101
RTR6(config-if)#load-interval 30
RTR6(config-if)#access-if-vpws
RTR6(config-acc-if-vpws)#mpls-l2-circuit VPWS10 primary
RTR6(config-acc-if-vpws)#exit
```

3. Configure sub-interface xe14.102.

```
RTR6(config)#interface xe14.102 switchport
RTR6(config-if)#encapsulation dot1q 102
RTR6(config-if)#load-interval 30
RTR6(config-if)#access-if-vpws
RTR6(config-acc-if-vpws)#mpls-l2-circuit VPWS20 primary
RTR6(config-acc-if-vpws)#exit
```

4. Enable QoS.

```
RTR6(config)#qos enable
```

5. Create class maps for queuing.

```
RTR6(config)#class-map type queuing que0
RTR6(config-cmap-que)#match queue 0
RTR6(config-cmap-que)#class-map type queuing que1
RTR6(config-cmap-que)#match queue 1
RTR6(config-cmap-que)#class-map type queuing que2
RTR6(config-cmap-que)#match queue 2
RTR6(config-cmap-que)#class-map type queuing que3
RTR6(config-cmap-que)#match queue 3
RTR6(config-cmap-que)#class-map type queuing service10
RTR6(config-cmap-que)#match service-template VPWS-10
RTR6(config-cmap-que)#class-map type queuing service20
RTR6(config-cmap-que)#match service-template VPWS-20
```

6. Create and configure policy maps.

```
RTR6(config)#policy-map type queuing queue
RTR6(config-pmap-que)#class type queuing que0
RTR6(config-pmap-c-que)#exit
RTR6(config-pmap-que)#class type queuing que1
RTR6(config-pmap-que)#exit
RTR6(config-pmap-que)#class type queuing que2
RTR6(config-pmap-que)#priority level 0
RTR6(config-pmap-c-que)#exit
RTR6(config-pmap-que)#class type queuing que3
RTR6(config-pmap-c-que)#exit
```

7. Create service policy.

```
RTR6(config)#policy-map type queuing service
RTR6(config-pmap-que)#class type queuing service10
RTR6(config-pmap-que)#priority level 0
RTR6(config-pmap-c-que)#service-policy queue
RTR6(config-pmap-c-que)#exit
RTR6(config-pmap-que)#class type queuing service20
RTR6(config-pmap-que)#service-policy queue
RTR6(config-pmap-que)#exit
```

8. Configure interface xe13.

```
RTR6(config)#interface xe13
RTR6(config-if)#service-policy type queuing output service
RTR6(config-if)#shape rate 1000 mbps
RTR6(config-if)#exit
```

Validation

RTR6

```
RTR6#show policy-map statistics type queuing rate mbps
+--------------------------------++
|  Class-map|Rate (in mbps) |
+--------------------------------++
xe13
q0
406.907
q2
413.234
q4
411.703

q6  415.307

RTR6#show running-config qos qos enable
!
class-map type queuing que0 match queue 0
!
class-map type queuing que1 match queue 1
!
class-map type queuing que2 match queue 2
!
class-map type queuing que3 match queue 3
!
class-map type queuing service10 match service-template VPWS-10
!
class-map type queuing service20 match service-template VPWS-20
!
!
policy-map type queuing queue class type queuing que0
exit
class type queuing que1 exit
class type queuing que2 exit
class type queuing que3 exit
!
policy-map type queuing service class type queuing service10
service-policy queue exit
class type queuing service20 service-policy queue
exit
!
interface xe13
service-policy type queuing output service

RTR6#show policy-map statistics type queuing rate mbps
+--------------------------------++
|  Class-map|Rate (in mbps) |
+--------------------------------++
```

```
xe0

q7

0.178
xe1

q7

0.170
xe2

q7

0.178
xe3

q7

0.186
xe6

q7

0.170
xe7
```

```
q7


0.186
xe8





q7


0.170
xe9





q7


0.178
xe13






class-default-q
(q0)
207.461
class-default-q
(q2)
207.031
class-default-q
(q4)
207.158
class-default-q
(q6)
197.740
service10





que0 (q0)


107.698
que1 (q1)
```

```
106.338
que2 (q2)


103.133
que3 (q3)


105.985
service20




que0
(q0)
103.376
que1
(q1)
102.938
que2
(q2)
101.485
que3
(q3)
101.537




RTR6#show policy-map statistics type queuing rate mbps
+--------------------------------++
|  Class-map|Rate (in mbps) |
+--------------------------------++
xe13
class-default-q
(q0)
80.431
class-default-q
(q2)
79.903
class-default-q
(q4)
79.780
class-default-q
(q6)
79.605
service10
```

que0 (q0)

100.546
que1 (q1)

105.640
que2 (q2)

103.802
que3 (q3)

103.724
service20

que0
(q0)
70.014
que1
(q1)
69.665
que2
(q2)
108.427
que3
(q3)
70.090

RTR1

`RTR1#configure terminal`	Enter into configuration mode
`RTR1(config)#mpls l2-circuit VPWS10 10 6.6.6.6`	Create MPLS l2 circuit with VPWS10 with ID 10 and end point 6.6.6.6
`(config-pseudowire)#exit`	Exit pseudowire config mode.
`RTR1(config)#mpls l2-circuit VPWS20 20 6.6.6.6`	Create MPLS l2 circuit with VPWS20 with ID 20 and end point 6.6.6.6
`(config-pseudowire)#exit`	Exit pseudowire config mode.

`RTR1(config)#service-template VPWS-10`	Create service template VPWS-10
`RTR1(config-svc)#match outer-vlan 10`	Match outer vlan 10
`RTR1(config-svc)#rewrite ingress translate 1000 outgoing-tpid dot1.q`	Rewrite ingress translate 1010 with outgoing tpid as dot1q
`RTR1(config-svc)#exit`	Exit
`RTR1(config)#`	
`RTR1(config-svc)#service-template VPWS-20`	Create service template VPWS-20
`RTR1(config-svc)# match outer-vlan 20`	Match outer vlan 20
`RTR1(config-svc)# rewrite ingress translate 1020 outgoing-tpid dot1.q`	Rewrite ingress translate 1020 with outgoing tpid as dot1q
`RTR1(config-svc)#exit`	Exit
`RTR1(config)#interface xe14`	Configure interface xe14
`RTR1(config-if)#switchport`	Configure as layer 2 port
`RTR1(config-if)#load-interval 30`	Configure load interval as 30
`RTR1(config-if)#mpls-l2-circuit VPWS10 service-template VPWS-10`	Attach l2circuit VPWS10 with service template VPWS-10
`RTR1(config-if)#mpls-l2-circuit VPWS20 service-template VPWS-20`	Attach l2circuit VPWS20 with service template VPWS-20
`RTR1(config-if)#exit`	exit

RTR6

`RTR6#configure terminal`	Enter into configuration mode
`RTR6(config)#mpls l2-circuit VPWS10 10 1.1.1.1`	Create MPLS l2 circuit with VPWS10 with ID 10 and end point 1.1.1.1
`(config-pseudowire)#exit`	Exit pseudowire config mode.
`RTR6(config)#mpls l2-circuit VPWS20 20 1.1.1.1`	Create MPLS l2 circuit with VPWS20 with ID 20 and end point 1.1.1.1
`(config-pseudowire)#exit`	Exit pseudowire config mode.
`RTR6(config)#service-template VPWS-10`	Create service template VPWS-10
`RTR6(config-svc)# match outer-vlan 10`	Match outer vlan 10
`RTR6(config-svc)# rewrite ingress translate 1000 outgoing-tpid dot1.q`	Rewrite ingress translate 1010 with outgoing tpid as dot1q
`RTR6(config-svc)#exit`	Exit
`RTR6(config-svc)#service-template VPWS-20`	Create service template VPWS-20
`RTR6(config-svc)#match outer-vlan 20`	Match outer VLAN 20
`RTR6(config-svc)#rewrite ingress translate 1020 outgoing-tpid dot1.q`	Rewrite ingress translate 1020 with outgoing tpid as dot1q
`RTR6(config-svc)#exit`	Exit
`RTR6(config)#interface xe14`	Configure interface xe14
`RTR6(config-if)#switchport`	Configure as layer 2 port
`RTR6(config-if)#load-interval 30`	Configure load interval as 30
`RTR6(config-if)#mpls-l2-circuit VPWS10 service-template VPWS-10`	Attach l2circuit VPWS10 with service template VPWS-10

`RTR6(config-if)#mpls-l2-circuit VPWS20 service-template VPWS-20`	Attach l2circuit VPWS20 with service template VPWS-20
`RTR6(config-if)#exit`	exit
`RTR6(config)#qos enable`	Enable Qos
`RTR6(config)#class-map type queuing que0`	Create class map of type queuing with name que0
`RTR6(config-cmap-que)# match queue 0`	Match for queue 0
`RTR6(config-cmap-que)#class-map type queuing que1`	Create class map of type queuing with name que1
`RTR6(config-cmap-que)#match queue 1`	Match for queue 1
`RTR6(config-cmap-que)#class-map type queuing que2`	Create class map of type queuing with name que2
`RTR6(config-cmap-que)#match queue 2`	Match for queue 2
`RTR6(config-cmap-que)#class-map type queuing que3`	Create class map of type queuing with name que3
`RTR6(config-cmap-que)#match queue 3`	Match for queue 3
`RTR6(config-cmap-que)#class-map type queuing service10`	Create class map of type queuing with name VPWS-10
`RTR6(config-cmap-que)#match service-template VPWS-10`	Match for service template VPWS-10
`RTR6(config-cmap-que)#class-map type queuing service20`	Create class map of type queuing with name VPWS-20
`RTR6(config-cmap-que)#match service-template VPWS-20`	Match for service template VPWS-20
`RTR6(config-cmap-que)#policy-map type queuing queue`	Create policy map with name queue
`RTR6(config-pmap-que)#class type queuing que0`	Add class map que0 to above policy map
`RTR6(config-pmap-c-que)#exit`	Exit
`RTR6(config-pmap-que)#class type queuing que1`	Add class map que1 to above policy map
`RTR6(config-pmap-c-que)#exit`	Exit
`RTR6(config-pmap-que)#class type queuing que2`	Add class map que2 to above policy map
`RTR6(config-pmap-que)#priority level 0`	configure Priority level 0
`RTR6(config-pmap-c-que)#exit`	Exit
`RTR6(config-pmap-que)#class type queuing que3`	Add class map que3 to above policy map
`RTR6(config-pmap-c-que)#exit`	Exit
`RTR6(config-pmap-que)#policy-map type queuing service`	Create policy map with name service
`RTR6(config-pmap-que)#class type queuing service10`	Add class map service10 to above policy map
`RTR6(config-pmap-que)#priority level 0`	Change priority level 0
`RTR6(config-pmap-c-que)#service-policy queue`	Add policy map queue as service-policy
`RTR6(config-pmap-c-que)#exit`	Exit

`RTR6(config-pmap-que)#class type queuing service20`	Add class map service20 to above policy map
`RTR6(config-pmap-c-que)#service-policy queue`	Add policy map queue as service-policy
`RTR6(config-pmap-c-que)#exit`	Exit
`RTR6(config-pmap-que)#interface xe13`	Configure interface xe13
`RTR6(config-if)#service-policy type queuing output service`	Attach the policy map service to the interface
`RTR6(config-if)#shape rate 1000 mbps`	Shape rate traffic to 1000 mbps
`RTR6(config-if)#exit`	Exit

Validation

RTR6:

```
RTR6#show policy-map statistics type queuing rate mbps
+---------------------------------+--------------------+
|            Class-map            |    Rate (in mbps)  |
+---------------------------------+--------------------+
xe13
 q0                                  406.907
 q2                                  413.234
 q4                                  411.703
 q6                                  415.307

RTR6#show running-config qos
qos enable
!
class-map type queuing que0
 match queue 0
!
class-map type queuing que1
 match queue 1
!
class-map type queuing que2
 match queue 2
!
class-map type queuing que3
 match queue 3
!
class-map type queuing service10
 match service-template VPWS-10
!
class-map type queuing service20
 match service-template VPWS-20
!
!
policy-map type queuing queue
 class type queuing que0
  exit
```

```
 class type queuing que1
  exit
 class type queuing que2
  exit
 class type queuing que3
  exit
!
policy-map type queuing service
 class type queuing service10
  service-policy queue
  exit
 class type queuing service20
  service-policy queue
  exit
!
interface xe13
 service-policy type queuing output service
!

RTR6#show policy-map statistics type queuing rate mbps
+---------------------------------+--------------------+
|            Class-map            |    Rate (in mbps)  |
+---------------------------------+--------------------+
xe0
 q7                                 0.178
xe1
 q7                                 0.170
xe2
 q7                                 0.178
xe3
 q7                                 0.186
xe6
 q7                                 0.170
xe7
 q7                                 0.186
xe8
 q7                                 0.170
xe9
 q7                                 0.178
xe13
 class-default-q (q0)               207.461
 class-default-q (q2)               207.031
 class-default-q (q4)               207.158
 class-default-q (q6)               197.740
 service10
   que0 (q0)                        107.698
   que1 (q1)                        106.338
   que2 (q2)                        103.133
   que3 (q3)                        105.985
 service20
```

```
   que0 (q0)                      103.376
   que1 (q1)                      102.938
   que2 (q2)                      101.485
   que3 (q3)                      101.537


RTR6#show running-config qos
qos enable
!
class-map type queuing que0
 match queue 0
!
class-map type queuing que1
 match queue 1
!
class-map type queuing que2
 match queue 2
!
class-map type queuing que3
 match queue 3
!
class-map type queuing service10
 match service-template VPWS-10
!
class-map type queuing service20
 match service-template VPWS-20
!
!
policy-map type queuing queue
 class type queuing que0
  exit
 class type queuing que1
  exit
 class type queuing que2
  priority level 0
  exit
 class type queuing que3
  exit
!
policy-map type queuing service
 class type queuing service10
  priority level 0
  service-policy queue
  exit
 class type queuing service20
  service-policy queue
  exit
!
interface xe13
 service-policy type queuing output service
```

```
 shape rate 1000 mbps
!

RTR6#show policy-map statistics type queuing rate mbps
+--------------------------------+--------------------+
|            Class-map           |    Rate (in mbps)  |
+--------------------------------+--------------------+
xe13
 class-default-q (q0)             80.431
 class-default-q (q2)             79.903
 class-default-q (q4)             79.780
 class-default-q (q6)             79.605
 service10
   que0 (q0)                      100.546
   que1 (q1)                      105.640
   que2 (q2)                      103.802
   que3 (q3)                      103.724
 service20
   que0 (q0)                      70.014
   que1 (q1)                      69.665
   que2 (q2)                      108.427
   que3 (q3)                      70.090
```

L3VPN- Sub/VLAN interfaces

Figure 19-5 displays a six node topology configured with end to end connectivity from Router 1 to Router 6. The end to end connectivity is established by configuring OSPF, iBGP and LDP configuration in all the routers. We should be able to ping each device from other device from topology. Configure L3VPN with sub interface and VLAN interface configurations on RTR1 and RTR6 and create Qos configuration on RTR6 and verify service queuing.

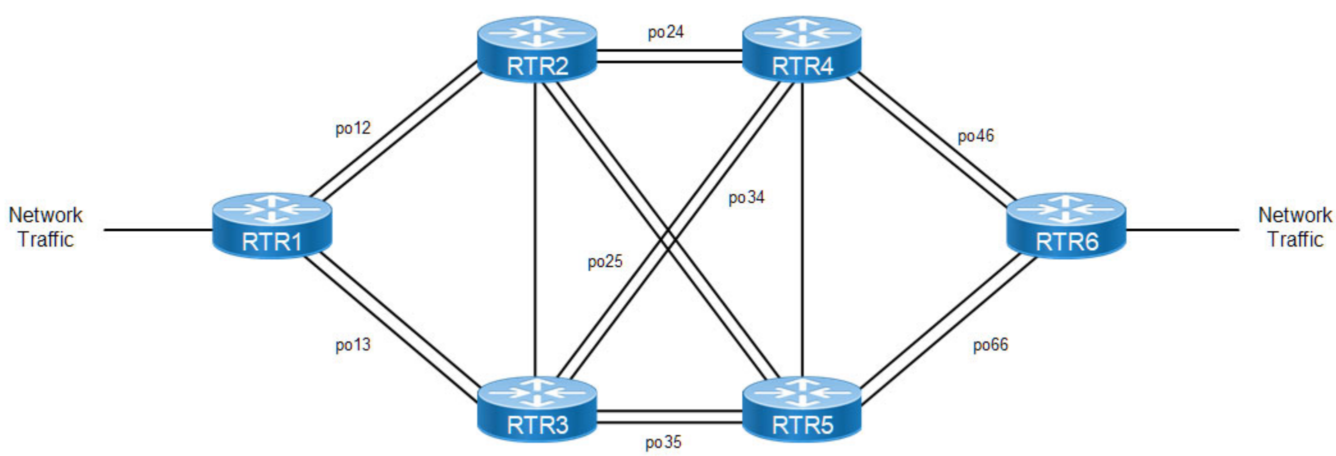


Figure 19-5: Simple Topology

RTR1

`RTR1(config)#ip vrf 3`	Create new vrf with name 3
`RTR1(config-vrf)#rd 2:3`	Create route distinguisher for vrf 3
`RTR1(config-vrf)#route-target both 1:3`	Create route target for vrf 3

`RTR1(config-vrf)#ip vrf 4`	Create new vrf with name 4
`RTR1(config-vrf)#rd 2:4`	Create route distinguisher for vrf 4
`RTR1(config-vrf)#route-target both 1:4`	Create route target for vrf 4
`RTR1(config-vrf)#ip vrf 5`	Create new vrf with name 5
`RTR1(config-vrf)#rd 2:5`	Create route distinguisher for vrf 5
`RTR1(config-vrf)#route-target both 1:5`	Create route target for vrf 5
`RTR1(config-vrf)#ip vrf 6`	Create new vrf with name 6
`RTR1(config-vrf)#rd 2:6`	Create route distinguisher for vrf 6
`RTR1(config-vrf)#route-target both 1:6`	Create route target for vrf 6
`RTR1(config-vrf)#exit`	Exit
`RTR1(config)#interface lo.3`	Configure loopback lo.3
`RTR1(config-if)#ip vrf forwarding 3`	Attach VRF 3 to loopback lo.3 interface
`RTR1(config-if)#interface lo.4`	Configure loopback lo.4
`RTR1(config-if)#ip vrf forwarding 4`	Attach VRF 4 to loopback lo.4 interface
`RTR1(config-if)#interface lo.5`	Configure loopback lo.5
`RTR1(config-if)#ip vrf forwarding 5`	Attach VRF 5 to loopback lo.5 interface
`RTR1(config-if)#interface lo.6`	Configure loopback lo.6
`RTR1(config-if)#ip vrf forwarding 6`	Attach vrf 6 to loopback lo.6 interface
`RTR1(config-if)#exit`	Exit
`RTR1(config)#interface ce52.203`	Create sub interface for ce52 port
`RTR1(config-if)#ip vrf forwarding 3`	Attach vrf 3 to sub int ce52.203
`RTR1(config-if)#ip address 201.203.1.1/24`	Assign ip address 201.203.1.1/24
`RTR1(config-if)#encapsulation dot1q 203`	Encapsulate dot1q VLAN with id 203
`RTR1(config-if)#interface ce52.204`	Create sub interface for ce52 port
`RTR1(config-if)#ip vrf forwarding 4`	Attach vrf 4 to sub int ce52.204
`RTR1(config-if)#ip address 201.204.1.1/24`	Assign ip address 201.204.1.1/24
`RTR1(config-if)#encapsulation dot1q 204`	Encapsulate dot1q VLAN with id 204
`RTR1(config-if)#interface vlan1.205`	Create VLAN interface with id 205
`RTR1(config-if)#ip vrf forwarding 5`	Attach VRF 5 to vlan1.205
`RTR1(config-if)#ip address 201.205.1.1/24`	Assign IP address of 201.205.1.1/24
`RTR1(config-if)#mtu 9216`	Configure MTU of size 9216
`RTR1(config-if)#interface vlan1.206`	Create VLAN interface with id 206
`RTR1(config-if)#ip vrf forwarding 6`	Attach VRF 6 to vlan1.206
`RTR1(config-if)#ip address 201.206.1.1/24`	Assign ip address of 201.206.1.1/24
`RTR1(config-if)#mtu 9216`	Configure MTU of size 9216
`RTR1(config)#router bgp 64512`	Configure router BGP 66512
`RTR1(config-router)#address-family vpnv4 unicast`	Enter into vpnv4 unicast address family
`RTR1(config-router-af)#neighbor 6.6.6.6 activate`	Activate neighbor on vpnv4 unicast
`RTR1(config-router-af)#exit-address-family`	Exit

`RTR1(config-router)#address-family ipv4 vrf 3`	Enter into ipv4 vrf3 address family
`RTR1(config-router-af)#redistribute connected`	Redistribute connected networks into vrf 3
`RTR1(config-router-af)#exit-address-family`	Exit
`RTR1(config-router)#address-family ipv4 vrf 4`	Enter into ipv4 vrf4 address family
`RTR1(config-router-af)#redistribute connected`	Redistribute connected networks into vrf 4
`RTR1(config-router-af)#exit-address-family`	Exit
`RTR1(config-router)#address-family ipv4 vrf 5`	Enter into ipv4 vrf5 address family
`RTR1(config-router-af)#redistribute connected`	Redistribute connected networks into vrf 5
`RTR1(config-router-af)#exit-address-family`	Exit
`RTR1(config-router)#address-family ipv4 vrf 6`	Enter into ipv4 vrf6 address family
`RTR1(config-router-af)#redistribute connected`	Redistribute connected networks into vrf 6
`RTR1(config-router-af)#exit-address-family`	Exit
`RTR1(config-router)#exit`	Exit
`RTR1(config)#interface ce52`	Configure interface ce52
`RTR1(config-if)#load-interval 30`	Configure load interval 30
`RTR1(config-if)#interface ce50`	Configure interface ce50
`RTR1(config-if)#switchport`	Configure port as layer 2 port
`RTR1(config-if)#bridge-group 1`	Configure interface in bridge group 1
`RTR1(config-if)#switchport mode trunk`	Configure interface as trunk port
`RTR1(config-if)#switchport trunk allowed vlan add 205, 206`	Allow only VLANs 205 and 206
`RTR1(config-if)#load-interval 30`	Configure load interval 30

RTR6

`RTR6(config)#ip vrf 3`	Create new vrf with name 3
`RTR6(config-vrf)#rd 1:3`	Create route distinguisher for VRF 3
`RTR6(config-vrf)#route-target both 1:3`	Create route target for VRF 3
`RTR6(config-vrf)#ip vrf 4`	Create new VRF with name 4
`RTR6(config-vrf)#rd 1:4`	Create route distinguisher for VRF 4
`RTR6(config-vrf)#route-target both 1:4`	Create route target for VRF 4
`RTR6(config-vrf)#ip vrf 5`	Create new VRF with name 5
`RTR6(config-vrf)#rd 1:5`	Create route distinguisher for VRF 5
`RTR6(config-vrf)#route-target both 1:5`	Create route target for VRF 5
`RTR6(config-vrf)#ip vrf 6`	Create new VRF with name 6
`RTR6(config-vrf)#rd 1:6`	Create route distinguisher for VRF 6
`RTR6(config-vrf)#route-target both 1:6`	Create route target for VRF 6

RTR6(config-vrf)#exit	Exit
RTR6(config)#interface lo.3	Configure loopback lo.3
RTR6(config-if)#ip vrf forwarding 3	Attach vrf 3 to loopback lo.3 interface
RTR6(config-if)#interface lo.4	Configure loopback lo.4
RTR6(config-if)#ip vrf forwarding 4	Attach vrf 4 to loopback lo.4 interface
RTR6(config-if)#interface lo.5	Configure loopback lo.5
RTR6(config-if)#ip vrf forwarding 5	Attach vrf 5 to loopback lo.5 interface
RTR6(config-if)#interface lo.6	Configure loopback lo.6
RTR6(config-if)#ip vrf forwarding 6	Attach VRF 6 to loopback lo.6 interface
RTR6(config-if)#interface xe13.103	Create sub interface for xe13 port
RTR6(config-if)#ip vrf forwarding 3	Attach VRF 3 to sub int xe13.103
RTR6(config-if)#ip address 101.103.1.1/24	Assign ip address 101.103.1.1/24
RTR6(config-if)#encapsulation dot1q 103	Encapsulate dot1q VLAN with id 103
RTR6(config-if)#interface xe13.104	Create sub interface for xe13.104 port
RTR6(config-if)#ip vrf forwarding 4	Attach VRF 4 to sub int xe13.104
RTR6(config-if)#ip address 101.104.1.1/24	Assign IP address 101.104.1.1/24
RTR6(config-if)#encapsulation dot1q 104	Encapsulate dot1q VLAN with id 104
RTR6(config-if)#interface vlan1.105	Create VLAN interface with id 105
RTR6(config-if)#ip vrf forwarding 5	Attach VRF 5 to vlan1.105
RTR6(config-if)#ip address 101.105.1.1/24	Assign ip address of 101.105.1.1/24
RTR6(config-if)#mtu 9216	Configure MTU of size 9216
RTR6(config-if)#interface vlan1.106	Create VLAN interface with id 106
RTR6(config-if)#ip vrf forwarding 6	Attach VRF 6 to vlan1.106
RTR6(config-if)#ip address 101.106.1.1/24	Assign IP address of 101.106.1.1/24
RTR6(config-if)#mtu 9216	Configure mtu of size 9216
RTR6(config-if)#exit	Exit
RTR6(config)#router bgp 64512	Configure BGP 64512
RTR6(config-router)#address-family vpnv4 unicast	Enter into vpnv4 unicast address family
RTR6(config-router-af)#neighbor 1.1.1.1 activate	Activate neighbor on vpnv4 unicast
RTR6(config-router-af)#exit-address-family	Exit
RTR6(config-router)#address-family ipv4 vrf 3	Enter into ipv4 vrf3 address family
RTR6(config-router-af)#redistribute connected	Redistribute connected networks into VRF 3
RTR6(config-router-af)#exit-address-family	Exit
RTR6(config-router)#address-family ipv4 vrf 4	Enter into ipv4 vrf4 address family
RTR6(config-router-af)#redistribute connected	Redistribute connected networks into vrf 4
RTR6(config-router-af)#exit-address-family	Exit

`RTR6(config-router)#address-family ipv4 vrf 5`	Enter into ipv4 vrf5 address family
`RTR6(config-router-af)#redistribute connected`	Redistribute connected networks into vrf 5
`RTR6(config-router-af)#exit-address-family`	Exit
`RTR6(config-router)#address-family ipv4 vrf 6`	Enter into ipv4 vrf6 address family
`RTR6(config-router-af)#redistribute connected`	Redistribute connected networks into vrf 6
`RTR6(config-router-af)#exit-address-family`	Exit
`RTR6(config-router)#exit`	Exit
`RTR6(config)#interface xe13`	Configure interface xe13
`RTR6(config-if)#load-interval 30`	Configure load interval 30
`RTR6(config-if)#interface xe14`	Configure interface xe14
`RTR6(config-if)#switchport`	Configure port as layer 2 port
`RTR6(config-if)#bridge-group 1`	Configure interface in bridge group 1
`RTR6(config-if)#switchport mode trunk`	Configure interface as trunk port
`RTR6(config-if)#switchport trunk allowed vlan add 105,106`	Allow only vlans105 and 106
`RTR6(config-if)#load-interval 30`	Configure load interval 30
`RTR6(config-if)#exit`	Exit

Validation

Validation for sub interfaces

RTR6:

```
RTR6#show running-config qos
qos enable
!
class-map type queuing que0
 match queue 0
!
class-map type queuing que1
 match queue 1
!
class-map type queuing que2
 match queue 2
!
class-map type queuing que3
 match queue 3
!
class-map type queuing xe13.103
 match interface xe13.103
!
class-map type queuing xe13.104
 match interface xe13.104
```

```
!
!
policy-map type queuing queuePolicy
 class type queuing que0
  exit
 class type queuing que1
  exit
 class type queuing que2
  exit
 class type queuing que3
  exit
!
policy-map type queuing service
 class type queuing xe13.103
  service-policy queuePolicy
  exit
 class type queuing xe13.104
  service-policy queuePolicy
  exit
interface xe13
 service-policy type queuing output service
!

RTR6#
RTR6#show policy-map statistics type queuing rate mbps
+---------------------------------+--------------------+
|            Class-map            |    Rate (in mbps)  |
+---------------------------------+--------------------+
xe13
 xe13.103
   que0 (q0)                        499.345
   que1 (q1)                        507.020
   que2 (q2)                        510.231
   que3 (q3)                        503.312
 xe13.104
   que0 (q0)                        502.730
   que1 (q1)                        497.862
   que2 (q2)                        508.641
   que3 (q3)                        489.681
RTR6#

RTR6# show running-config qos
qos enable
!
class-map type queuing que0
 match queue 0
!
class-map type queuing que1
 match queue 1
!
```

```
class-map type queuing que2
 match queue 2
!
class-map type queuing que3
 match queue 3
!
class-map type queuing xe13.103
 match interface xe13.103
!
class-map type queuing xe13.104
 match interface xe13.104
!
!
!
policy-map type queuing queuePolicy
 class type queuing que0
  exit
 class type queuing que1
  exit
 class type queuing que2
  exit
 class type queuing que3
  exit
!
policy-map type queuing service
 class type queuing xe13.103
  service-policy queuePolicy
  exit
 class type queuing xe13.104
  priority level 0
  service-policy queuePolicy
  exit
!
interface xe13
 service-policy type queuing output service
 shape rate 4000 mbps
!
RTR6#show policy-map statistics type queuing rate mbps
+--------------------------------+--------------------+
|            Class-map           |    Rate (in mbps)  |
+--------------------------------+--------------------+
xe13
 xe13.103
   que0 (q0)                      82.403
   que1 (q1)                      83.355
   que2 (q2)                      82.881
   que3 (q3)                      83.354
 xe13.104
   que0 (q0)                      498.243
   que1 (q1)                      502.950
```

```
   que2 (q2)                     510.054
   que3 (q3)                     494.921

Validation for vlan interfaces
RTR6:
RTR6#show policy-map statistics type queuing rate mbps
+---------------------------------+--------------------+
|            Class-map            |    Rate (in mbps)  |
+---------------------------------+--------------------+
xe13
 q0                              1988.796
 q1                              1976.734
 q5                              1997.010
 q6                              2022.276
 q7                              0.012

RTR6#show running-config qos
qos enable
!
class-map type queuing que0
 match queue 0
!
class-map type queuing que1
 match queue 1
!
class-map type queuing que2
 match queue 2
!
class-map type queuing que3
 match queue 3
!
class-map type queuing vlan1.105
 match interface vlan1.105
!
class-map type queuing vlan1.106
 match interface vlan1.106
!
!
policy-map type queuing queuePolicy
 class type queuing que0
  shape 100 mbps
  exit
 class type queuing que1
  exit
 class type queuing que2
  shape 150 mbps
  exit
 class type queuing que3
  exit
!
```

```
policy-map type queuing service
 class type queuing vlan1.105
  service-policy queuePolicy
  exit
 class type queuing vlan1.106
  wfq-queue weight 4
  service-policy queuePolicy
  exit
!
interface xe13
 service-policy type queuing output service
 shape rate 6000 mbps
!
RTR6#  show policy-map statistics type queuing rate mbps
+---------------------------------+--------------------+
|             Class-map           |    Rate (in mbps)  |
+---------------------------------+--------------------+
xe13
 vlan1.105
   que0 (q0)                         99.882
   que1 (q1)                         508.907
   que2 (q2)                         150.103
   que3 (q3)                         505.349
 vlan1.106
   que0 (q0)                         100.148
   que1 (q1)                         505.652
   que2 (q2)                         149.896
   que3 (q3)                         501.783
```

Provider Bridge

Figure 19-6 displays a six node topology configured with end to end connectivity from Router 1 to Router 6. The end to end connectivity is established by configuring OSPF, iBGP and LDP configuration in all the routers. We should be able to ping each device from other device from topology.

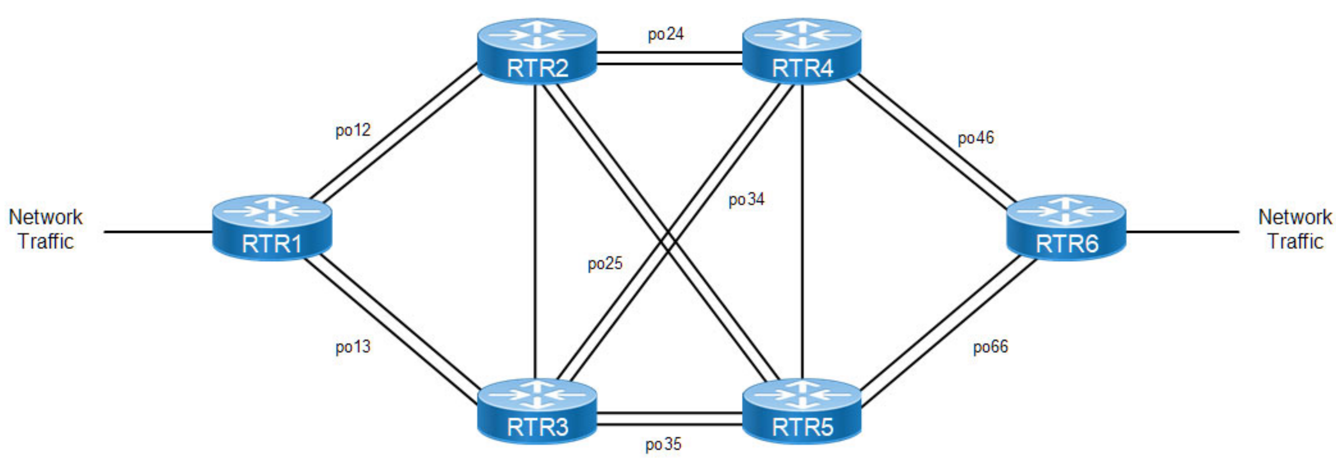


Figure 19-6: Simple Topology

RTR1

`RTR1#configure terminal`	Enter into configure terminal mode
`RTR1(config)#bridge 1 protocol provider-rstp edge`	Configure bridge 1 protocol as provider RSTP edge
`RTR1(config)#vlan database`	Configure VLAN database
`RTR1(config-vlan)#vlan 201-300 type customer bridge 1 state enable`	Configure customer VLANs from 201-300
`RTR1(config-vlan)#vlan 301-400 type service point-point bridge 1 state enable`	Configure service VLANs from 301-400
`RTR1(config-vlan)#cvlan registration table map1 bridge 1`	Create cvlan registration table
`RTR1(config-cvlan-registration)#cvlan 201 svlan 301`	Map 201 cvlan to 301 svlan
`RTR1(config-cvlan-registration)#cvlan 202 svlan 302`	Map 202 cvlan to 302 svlan
`RTR1(config-cvlan-registration)#cvlan 203 svlan 303`	Map 203 cvlan to 303 svlan
`RTR1(config-cvlan-registration)#cvlan 204 svlan 304`	Map 204 cvlan to 304 svlan
`RTR1(config-cvlan-registration)#cvlan 205 svlan 305`	Map 205 cvlan to 305 svlan
`RTR1(config-cvlan-registration)#interface po12`	Configure interface po12
`RTR1(config-if)#switchport`	Configure as layer 2 port
`RTR1(config-if)#bridge-group 1`	Configure interface in bridge group 1
`RTR1(config-if)#switchport mode provider-network`	Configure switchport mode as provider network
`RTR1(config-if)#switchport provider-network allowed vlan add 301-400`	Configure allowed VLAN from 301-400
`RTR1(config-if)#load-interval 30`	Configure load interval 30
`RTR1(config-if)#interface po13`	Configure interface po13
`RTR1(config-if)#switchport`	Configure as layer 2 port
`RTR1(config-if)#bridge-group 1`	Configure interface in bridge group 1
`RTR1(config-if)#switchport mode provider-network`	Configure switchport mode as provider network
`RTR1(config-if)#switchport provider-network allowed vlan add 301-400`	Configure allowed vlan from 301-400
`RTR1(config-if)#load-interval 30`	Configure load interval 30
`RTR1(config-if)#interface xe1`	Configure xe1
`RTR1(config-if)#channel-group 12 mode active`	Add xe1 to po12
`RTR1(config-if)#interface xe2`	Configure xe2
`RTR1(config-if)#channel-group 12 mode active`	Add xe2 to po12
`RTR1(config-if)#interface xe3`	Configure xe3

© 2026 IP Infusion Inc. Proprietary

RTR1(config-if)#channel-group 13 mode active	Add xe3 to po13
RTR1(config-if)#interface xe4	Configure xe4
RTR1(config-if)#channel-group 13 mode active	Add xe4 to po13
RTR1(config-if)#interface xe15	Configure interface xe15
RTR1(config-if)#switchport	Configure as layer 2 port
RTR1(config-if)#bridge-group 1	Configure interface in bridge group 1
RTR1(config-if)#switchport mode customer-edge trunk	Configure switchport mode as customer edge trunk
RTR1(config-if)#switchport customer-edge trunk allowed vlan add 201-300	Configure allowed vlan from 201-300
RTR1(config-if)#switchport customer-edge vlan registration map1	Attach VLAN registration table map 1 to xe15
RTR1(config-if)#load-interval 30	Configure load interval 30
RTR1(config-if)#exit	Exit

RTR2

RTR2#configure terminal	Configure terminal
RTR2(config)#bridge 1 protocol provider-rstp	Configure bridge 1 protocol as provider rstp edge
RTR2(config)#vlan database	VLAN database
RTR2(config-vlan)#vlan 301-400 type service point-point bridge 1 state enable	Configure service vlans from 301-400
RTR2(config-vlan)#interface po12	Configure interface po12
RTR2(config-if)#switchport	Configure as layer 2 port
RTR2(config-if)#bridge-group 1	Configure interface in bridge group 1
RTR2(config-if)#switchport mode provider-network	Configure switchport mode as provider network
RTR2(config-if)#switchport provider-network allowed vlan add 301-400	Configure allowed vlan from 301-400
RTR2(config-if)#load-interval 30	Configure load interval 30
RTR2(config-if)#interface po24	Configure interface po24
RTR2(config-if)#switchport	Configure as layer 2 port
RTR2(config-if)#bridge-group 1	Configure interface in bridge group 1
RTR2(config-if)#switchport mode provider-network	Configure switchport mode as provider network
RTR2(config-if)#switchport provider-network allowed vlan add 301-400	Configure allowed vlan from 301-400
RTR2(config-if)#load-interval 30	Configure load interval 30
RTR2(config-if)#interface po25	Configure interface po25
RTR2(config-if)#switchport	Configure as layer 2 port
RTR2(config-if)#bridge-group 1	Configure interface in bridge group 1
RTR2(config-if)#switchport mode provider-network	Configure switchport mode as provider network

© 2026 IP Infusion Inc. Proprietary

RTR2(config-if)#switchport provider-network allowed vlan add 301-400	Configure allowed vlan from 301-400
RTR2(config-if)#load-interval 30	Configure load interval 30
RTR2(config-if)#interface xe1	Configure xe1
RTR2(config-if)#channel-group 12 mode active	Add xe1 to po12
RTR2(config-if)#interface xe2	Configure xe2
RTR2(config-if)#channel-group 12 mode active	Add xe2 to po12
RTR2(config-if)#interface xe8	Configure xe8
RTR2(config-if)#channel-group 25 mode active	Add xe8 to po25
RTR2(config-if)#interface xe9	Configure xe9
RTR2(config-if)#channel-group 25 mode active	Add xe9 to po25
RTR2(config-if)#interface xe14	Configure xe14
RTR2(config-if)#channel-group 24 mode active	Add xe14 to po24
RTR2(config-if)#interface xe15	Configure xe15
RTR2(config-if)#channel-group 24 mode active	Add xe15 to po24
RTR2(config)#exit	Exit

RTR3

RTR3#config term	Configure terminal
RTR3(config)#bridge 1 protocol provider-rstp	Configure bridge 1 protocol as provider rstp edge
RTR3(config)#vlan database	Configure VLAN database
RTR3(config-vlan)#vlan 301-400 type service point-point bridge 1 state enable	Configure service VLANS from 301-400
RTR3(config-vlan)#interface po13	Configure interface po13
RTR3(config-if)# switchport	Configure as layer 2 port
RTR3(config-if)# bridge-group 1	Configure interface in bridge group 1
RTR3(config-if)#switchport mode provider-network	Configure switchport mode as provider network
RTR3(config-if)#switchport provider-network allowed vlan add 301-400	Configure allowed vlan from 301-400
RTR3(config-if)#load-interval 30	Configure load interval 30
RTR3(config-if)#interface po34	Configure interface po34
RTR3(config-if)#switchport	Configure as layer 2 port
RTR3(config-if)#bridge-group 1	Configure interface in bridge group 1
RTR3(config-if)#switchport mode provider-network	Configure switchport mode as provider network
RTR3(config-if)#switchport provider-network allowed vlan add 301-400	Configure allowed vlan from 301-400

`RTR3(config-if)#load-interval 30`	Configure load interval 30
`RTR3(config-if)#interface po35`	Configure interface po35
`RTR3(config-if)#switchport`	Configure as layer 2 port
`RTR3(config-if)#bridge-group 1`	Configure interface in bridge group 1
`RTR3(config-if)#switchport mode provider-network`	Configure switchport mode as provider network
`RTR3(config-if)#switchport provider-network allowed vlan add 301-400`	Configure allowed vlan from 301-400
`RTR3(config-if)#load-interval 30`	Configure load interval 30
`RTR3(config-if)#interface xe3`	Configure xe3
`RTR3(config-if)#channel-group 13 mode active`	Add xe3 to po13
`RTR3(config-if)#interface xe4`	Configure xe4
`RTR3(config-if)# channel-group 13 mode active`	Add xe4 to po13
`RTR3(config-if)#interface xe7`	Configure xe7
`RTR3(config-if)#channel-group 34 mode active`	Add xe7 to po34
`RTR3(config-if)#interface xe8`	Configure xe8
`RTR3(config-if)#channel-group 34 mode active`	Add xe8 to po34
`RTR3(config-if)#interface xe14`	Configure xe14
`RTR3(config-if)#channel-group 35 mode active`	Add xe14 to po35
`RTR3(config-if)#interface xe15`	Configure xe15
`RTR3(config-if)#channel-group 35 mode active`	Add xe15 to po35
`RTR3(config-if)#exit`	Exit

RTR4

`RTR4#config term`	Configure terminal
`RTR4(config)#bridge 1 protocol provider-rstp`	Configure bridge 1 protocol as provider rstp edge
`RTR4(config)#vlan database`	Configure vlan database
`RTR4(config-vlan)#vlan 301-400 type service point-point bridge 1 state enable`	Configure service vlans from 301-400
`RTR4(config-vlan)#interface po24`	Configure interface po24
`RTR4(config-if)#switchport`	Configure as layer 2 port
`RTR4(config-if)#bridge-group 1`	Configure interface in bridge group 1
`RTR4(config-if)#switchport mode provider-network`	Configure switchport mode as provider network
`RTR4(config-if)#switchport provider-network allowed vlan add 301-400`	Configure allowed VLAN from 301-400
`RTR4(config-if)#load-interval 30`	Configure load interval 30
`RTR4(config-if)#interface po34`	Configure interface po34

RTR4(config-if)#switchport	Configure as layer 2 port
RTR4(config-if)#bridge-group 1	Configure interface in bridge group 1
RTR4(config-if)#switchport mode provider-network	Configure switchport mode as provider network
RTR4(config-if)#switchport provider-network allowed vlan add 301-400	Configure allowed vlan from 301-400
RTR4(config-if)#load-interval 30	Configure load interval 30
RTR4(config-if)#interface po46	Configure interface po46
RTR4(config-if)#switchport	Configure as layer 2 port
RTR4(config-if)#bridge-group 1	Configure interface in bridge group 1
RTR4(config-if)#switchport mode provider-network	Configure switchport mode as provider network
RTR4(config-if)#switchport provider-network allowed vlan add 301-400	Configure allowed vlan from 301-400
RTR4(config-if)#load-interval 30	Configure load interval 30
RTR4(config-if)#interface xe0	Configure xe0
RTR4(config-if)#channel-group 46 mode active	Add xe0 to po46
RTR4(config-if)#interface xe1	Configure xe1
RTR4(config-if)#channel-group 46 mode active	Add xe1 to po46
RTR4(config-if)#interface xe2	Configure xe2
RTR4(config-if)#channel-group 46 mode active	Add xe2 to po46
RTR4(config-if)#interface xe3	Configure xe3
RTR4(config-if)#channel-group 46 mode active	Add xe3 to po46
RTR4(config-if)#interface xe7	Configure xe7
RTR4(config-if)#channel-group 34 mode active	Add xe7 to po34
RTR4(config-if)#interface xe8	Configure xe8
RTR4(config-if)#channel-group 34 mode active	Add xe8 to po34
RTR4(config-if)#interface xe14	Configure xe14
RTR4(config-if)#channel-group 24 mode active	Add xe14 to po24
RTR4(config-if)#interface xe15	Configure xe15
RTR4(config-if)#channel-group 24 mode active	Add xe15 to po24
RTR4(config-if)#end	End

RTR5

RTR5#config term	Configure terminal
RTR5(config)#bridge 1 protocol provider-rstp	Configure bridge 1 protocol as provider rstp edge

© 2026 IP Infusion Inc. Proprietary

`RTR5(config)#vlan database`	Configure vlan database
`RTR5(config-vlan)#vlan 301-305 type service point-point bridge 1 state enable`	Configure service vlans from 301-305
`RTR5(config-vlan)#interface po25`	Configure interface po25
`RTR5(config-if)#switchport`	Configure as layer 2 port
`RTR5(config-if)#bridge-group 1`	Configure interface in bridge group 1
`RTR5(config-if)#switchport mode provider-network`	Configure switchport mode as provider network
`RTR5(config-if)#switchport provider-network allowed vlan add 301-400`	Configure allowed vlan from 301-400
`RTR5(config-if)#load-interval 30`	Configure load interval 30
`RTR5(config-if)#interface po35`	Configure interface po35
`RTR5(config-if)#switchport`	Configure as layer 2 port
`RTR5(config-if)#bridge-group 1`	Configure interface in bridge group 1
`RTR5(config-if)#switchport mode provider-network`	Configure switchport mode as provider network
`RTR5(config-if)#switchport provider-network allowed vlan add 301-400`	Configure allowed vlan from 301-400
`RTR5(config-if)#load-interval 30`	Configure load interval 30
`RTR5(config-if)#interface po56`	Configure interface po56
`RTR5(config-if)#switchport`	Configure as layer 2 port
`RTR5(config-if)#bridge-group 1`	Configure interface in bridge group 1
`RTR5(config-if)#switchport mode provider-network`	Configure switchport mode as provider network
`RTR5(config-if)#switchport provider-network allowed vlan add 301-400`	Configure allowed vlan from 301-400
`RTR5(config-if)#load-interval 30`	Configure load interval 30
`RTR5(config-if)#interface xe2`	Configure xe2
`RTR5(config-if)#channel-group 25 mode active`	Add xe2 to po25
`RTR5(config-if)#interface xe3`	Configure xe3
`RTR5(config-if)#channel-group 25 mode active`	Add xe3 to po25
`RTR5(config-if)#interface xe6`	Configure xe6
`RTR5(config-if)#channel-group 56 mode active`	Add xe6 to po56
`RTR5(config-if)#interface xe7`	Configure xe7
`RTR5(config-if)#channel-group 56 mode active`	Add xe7 to po56
`RTR5(config-if)#interface xe8`	Configure xe8
`RTR5(config-if)#channel-group 56 mode active`	Add xe8 to po56
`RTR5(config-if)#interface xe9`	Configure xe9
`RTR5(config-if)#channel-group 56 mode active`	Add xe9 to po56

© 2026 IP Infusion Inc. Proprietary

`RTR5(config-if)#interface xe14`	Configure xe14
`RTR5(config-if)#channel-group 35 mode active`	Add xe14 to po35
`RTR5(config-if)#interface xe15`	Configure xe15
`RTR5(config-if)#channel-group 35 mode active`	Add xe15 to po35
`RTR5(config-if)#end`	End

RTR6

`RTR6#config terminal`	Configure terminal
`RTR6(config)#bridge 1 protocol provider-rstp edge`	Configure bridge 1 protocol as provider rstp edge
`RTR6(config)#vlan database`	Configure vlan database
`RTR6(config-vlan)#vlan 201-300 type customer bridge 1 state enable`	Configure customer VLANS from 201-300
`RTR6(config-vlan)#vlan 301-400 type service point-point bridge 1 state enable`	Configure service VLANs from 301-400
`RTR6(config-vlan)#cvlan registration table map1 bridge 1`	Create cvlan registration table
`RTR6(config-cvlan-registration)#cvlan 201 svlan 301`	Map 201 cvlan to 301 svlan
`RTR6(config-cvlan-registration)#cvlan 202 svlan 302`	Map 202 cvlan to 302 svlan
`RTR6(config-cvlan-registration)#cvlan 203 svlan 303`	Map 203 cvlan to 303 svlan
`RTR6(config-cvlan-registration)#cvlan 204 svlan 304`	Map 204 cvlan to 304 svlan
`RTR6(config-cvlan-registration)#cvlan 205 svlan 305`	Map 205 cvlan to 305 svlan
`RTR6(config-cvlan-registration)#interface po46`	Configure interface po46
`RTR6(config-if)#switchport`	Configure as layer 2 port
`RTR6(config-if)#bridge-group 1`	Configure interface in bridge group 1
`RTR6(config-if)#switchport mode provider-network`	Configure switchport mode as provider network
`RTR6(config-if)#switchport provider-network allowed vlan add 301-400`	Configure allowed vlan from 301-400
`RTR6(config-if)#load-interval 30`	Configure load interval 30
`RTR6(config-if)#interface po56`	Configure interface po56
`RTR6(config-if)#switchport`	Configure as layer 2 port
`RTR6(config-if)#bridge-group 1`	Configure interface in bridge group 1
`RTR6(config-if)#switchport mode provider-network`	Configure switchport mode as provider network
`RTR6(config-if)#switchport provider-network allowed vlan add 301-400`	Configure allowed vlan from 301-400
`RTR6(config-if)#load-interval 30`	Configure load interval 30

© 2026 IP Infusion Inc. Proprietary

`RTR6(config-if)#interface xe0`	Configure xe0
`RTR6(config-if)#channel-group 46 mode active`	Add xe0 to po46
`RTR6(config-if)#interface xe1`	Configure xe1
`RTR6(config-if)#channel-group 46 mode active`	Add xe1 to po46
`RTR6(config-if)#interface xe2`	Configure xe2
`RTR6(config-if)#channel-group 46 mode active`	Add xe2 to po46
`RTR6(config-if)#interface xe3`	Configure xe3
`RTR6(config-if)#channel-group 46 mode active`	Add xe3 to po46
`RTR6(config-if)#interface xe6`	Configure xe6
`RTR6(config-if)#channel-group 56 mode active`	Add xe6 to po56
`RTR6(config-if)#interface xe7`	Configure xe7
`RTR6(config-if)#channel-group 56 mode active`	Add xe7 to po56
`RTR6(config-if)#interface xe8`	Configure xe8
`RTR6(config-if)#channel-group 56 mode active`	Add xe8 to po56
`RTR6(config-if)#interface xe9`	Configure xe9
`RTR6(config-if)#channel-group 56 mode active`	Add xe9 to po56
`RTR6(config-if)#interface xe15`	Configure interface xe15
`RTR6(config-if)#switchport`	Configure as layer 2 port
`RTR6(config-if)#bridge-group 1`	Configure interface in bridge group 1
`RTR6(config-if)#switchport mode customer-edge trunk`	Configure switchport mode as customer edge trunk
`RTR6(config-if)#switchport customer-edge trunk allowed vlan add 201-300`	Configure allowed vlan from 201-300
`RTR6(config-if)#switchport customer-edge vlan registration map1`	Attach vlan registration table map 1 to xe15
`RTR6(config-if)#load-interval 30`	Configure load interval 30
`RTR6(config)#class-map type queuing que0`	Create Class map type queuing que0
`RTR6(config-cmap-que)#match queue 0`	Match for queue 0
`RTR6(config-cmap-que)#class-map type queuing que1`	Create Class map type queuing que0
`RTR6(config-cmap-que)#match queue 1`	Match for queue 1
`RTR6(config-cmap-que)#class-map type queuing que2`	Create Class map type queuing que1
`RTR6(config-cmap-que)#match queue 2`	Match for queue 2
`RTR6(config-cmap-que)#class-map type queuing que3`	Create Class map type queuing que2
`RTR6(config-cmap-que)#match queue 3`	Match for queue 3

RTR6(config-cmap-que)#class-map type queuing vlan201	Create class map type queuing vlan201
RTR6(config-cmap-que)#match vlan 201	Match for VLAN 201
RTR6(config-cmap-que)#class-map type queuing vlan202	Create class map type queuing vlan202
RTR6(config-cmap-que)#match vlan 202	Match for VLAN 202
RTR6(config-cmap-que)#class-map type queuing vlan203	Create class map type queuing vlan203
RTR6(config-cmap-que)#match vlan 203	Match for VLAN 203
RTR6(config-cmap-que)#class-map type queuing vlan204	Create class map type queuing vlan204
RTR6(config-cmap-que)#match vlan 204	Match for VLAN 204
RTR6(config-cmap-que)#policy-map type queuing queue	Create policy map of type queuing with name queue
RTR6(config-pmap-que)#class type queuing que0	Attach class map que0
RTR6(config-pmap-c-que)#exit	Exit
RTR6(config-pmap-que)#class type queuing que1	Attach class map que1
RTR6(config-pmap-c-que)#exit	Exit
RTR6(config-pmap-que)#class type queuing que2	Attach class map que2
RTR6(config-pmap-c-que)#exit	Exit
RTR6(config-pmap-que)#class type queuing que3	Attach class map que3
RTR6(config-pmap-c-que)#exit	Exit
RTR6(config-pmap-que)#policy-map type queuing vlan	Create policy map of type queuing with name vlan
RTR6(config-pmap-que)#class type queuing vlan201	Attach class map vlan201
RTR6(config-pmap-c-que)#service-policy queue	Attach service policy queue
RTR6(config-pmap-c-que)#exit	Exit
RTR6(config-pmap-que)#class type queuing vlan202	Attach class map vlan202
RTR6(config-pmap-c-que)#shape 100 mbps	Shape traffic to 100 mbps
RTR6(config-pmap-c-que)#service-policy queue	Attach service policy queue
RTR6(config-pmap-c-que)#exit	Exit
RTR6(config-pmap-que)#class type queuing vlan203	Attach class map vlan203
RTR6(config-pmap-c-que)#service-policy queue	Attach service policy queue
RTR6(config-pmap-c-que)#exit	Exit
RTR6(config-pmap-que)#class type queuing vlan204	Attach class map vlan204

`RTR6(config-pmap-c-que)#service-policy queue`	Attach service policy queue
`RTR6(config-pmap-c-que)#exit`	Exit
`RTR6(config-pmap-que)#interface xe13`	Configure interface xe13
`RTR6(config-if)#service-policy type queuing output vlan`	Attach service policy vlan to interface xe13
`RTR6(config-if)#shape rate 1000 mbps`	Shape rate 1000 mbps
`RTR6(config-if)#exit`	Exit

Validation

RTR6:

```
RTR6#show running-config qos
qos enable
!
RTR6#show policy-map statistics type queuing rate mbps
+---------------------------------+--------------------+
|             Class-map           |    Rate (in mbps)  |
+---------------------------------+--------------------+
xe13
 q0                                 407.399
 q2                                 415.419
 q4                                 398.478
 q6                                 411.140


RTR6#show running-config qos
qos enable
!
class-map type queuing que0
 match queue 0
!
class-map type queuing que1
 match queue 1
!
class-map type queuing que2
 match queue 2
!
class-map type queuing que3
 match queue 3
!
class-map type queuing vlan201
 match vlan 201
!
class-map type queuing vlan202
 match vlan 202
!
class-map type queuing vlan203
```

```
 match vlan 203
!
class-map type queuing vlan204
 match vlan 204
!
!
policy-map type queuing queue
 class type queuing que0
  exit
 class type queuing que1
  exit
 class type queuing que2
  exit
 class type queuing que3
  exit
!
policy-map type queuing vlan
 class type queuing vlan201
  service-policy queue
  exit
 class type queuing vlan202
  service-policy queue
  exit
 class type queuing vlan203
  service-policy queue
  exit
 class type queuing vlan204
  service-policy queue
  exit
!
interface xe13
 service-policy type queuing output vlan
!

RTR6#show policy-map statistics type queuing rate mbps
+--------------------------------+--------------------+
|            Class-map           |    Rate (in mbps)  |
+--------------------------------+--------------------+
xe13
 vlan201
   que0 (q0)                      98.957
   que1 (q1)                      106.502
   que2 (q2)                      103.779
   que3 (q3)                      100.639
 vlan202
   que0 (q0)                      102.953
   que1 (q1)                      96.453
   que2 (q2)                      103.694
   que3 (q3)                      100.486
 vlan203
```

```
   que0 (q0)                      100.919
   que1 (q1)                      95.551
   que2 (q2)                      103.445
   que3 (q3)                      105.183
 vlan204
   que0 (q0)                      101.593
   que1 (q1)                      101.609
   que2 (q2)                      102.030
   que3 (q3)                      102.992

RTR6#show running-config qos
qos enable
!
class-map type queuing que0
 match queue 0
!
class-map type queuing que1
 match queue 1
!
class-map type queuing que2
 match queue 2
!
class-map type queuing que3
 match queue 3
!
class-map type queuing vlan201
 match vlan 201
!
class-map type queuing vlan202
 match vlan 202
!
class-map type queuing vlan203
 match vlan 203
!
class-map type queuing vlan204
 match vlan 204
!
!
policy-map type queuing queue
 class type queuing que0
  exit
 class type queuing que1
  exit
 class type queuing que2
  exit
 class type queuing que3
  exit
!
policy-map type queuing vlan
 class type queuing vlan201
```

```
  service-policy queue
  exit
 class type queuing vlan202
  shape 100 mbps
  service-policy queue
  exit
 class type queuing vlan203
  service-policy queue
  exit
 class type queuing vlan204
  service-policy queue
  exit
!
interface xe13
 service-policy type queuing output vlan
 shape rate 1000 mbps
!

RTR6#show policy-map statistics type queuing rate mbps
+---------------------------------+--------------------+
|            Class-map            |    Rate (in mbps)  |
+---------------------------------+--------------------+
xe13
 vlan201
   que0 (q0)                       75.224
   que1 (q1)                       74.814
   que2 (q2)                       75.476
   que3 (q3)                       75.298
 vlan202
   que0 (q0)                       25.327
   que1 (q1)                       25.068
   que2 (q2)                       24.684
   que3 (q3)                       25.379
 vlan203
   que0 (q0)                       75.943
   que1 (q1)                       75.859
   que2 (q2)                       75.492
   que3 (q3)                       75.115
 vlan204
   que0 (q0)                       75.513
   que1 (q1)                       75.295
   que2 (q2)                       75.907
   que3 (q3)                       76.048
```

CHAPTER 20 Queue Compensation

The size of packets transmitted may vary from the size of packets queued in ingress and egress queues. Contributing factors are:

1. Ethernet Overhead

   Fixed size, typically 20-bytes

2. Internal DRAM CRC

   Fixed size, 2-byte CRC

3. Packet editing resulting from Packet Processing

   Size may vary per queue or per packet

   Network header termination (i.e., Layer-3 link layer termination from routing, MPLS/IP tunnel termination, VLAN Tag removal)

   Network header encapsulation (i.e., Layer-3 link layer encapsulation, MPLS/IP tunnel initiation, Vlan tag addition)

These bytes need to be adjusted to achieve the proper egress rate. Hence, compensation is used to adjust this byte difference in order to achieve the expected egress rate.

The default compensation for all queues is set to -22 internally (Ethernet overhead + internal CRC). This is as per the internal headers in the packet pipeline.

When an attachment-circuit is created on a port the compensation is updated as shown in Table 20-12.

Table 20-12: Compensation updates

AC ingress operation	Compensation
POP	-38
NONE/Translate	-42 (defaults)
PUSH	-46

Note: It is the operator's responsibility to update the compensation if required per application.

The user can configure compensation on the class-maps matching services for service queues, and class-map for “class-default-q” for port queues.

CLI: `compensation <-64-64>`

For example:

```
class-map type queuing data
 match queue 0
!
class-map type queuing service1
 match service-template ETH-2016
!
class-map type queuing service2
 match service-template ETH-2017
!
class-map type queuing signal
 match queue 3
```

```
!
class-map type queuing voice
 match queue 1
!
policy-map type queuing configPolicy1
 class type queuing class-default-q
  exit
 class type queuing data
  exit
 class type queuing signal
  exit
 class type queuing voice
  exit
!
policy-map type queuing customer1
 class type queuing service1
  compensation -20
  service-policy configPolicy1
 class type queuing class-default-q
  compensation -20
  exit
!
interface xe11
 service-policy type queuing output customer1
```

© 2026 IP Infusion Inc. Proprietary

CHAPTER 21 Hierarchical Traffic Policing

Basic traffic policing is explained in Chapter 3, *Traffic Policing*, which covers single-level policing. The Qumran device support two level hierarchical policing in addition to single level traffic policing (Hierarchical policing is also in compliance with RFC 2697 and RFC 4115).

Hierarchical policing ensures traffic policing in two levels in serial mode. Traffic is policed with a “child policer” configuration first, and then policed at a second level, the “parent policer” configuration. Thus, traffic is treated with two levels of policing.

Hierarchical policing will be useful when traffic to be policed per service and additionally to be policed per customers of that particular service. Hierarchical policing is useful when multiple streams (such as voice, data) of customer traffic to be policed and additionally, traffic to be policed per customer. Configuration considerations for hierarchical policing are same as mentioned in Chapter 3, *Traffic Policing*.

Configuring Hierarchical Traffic Policing

Command to configure policing on a class remains same as explained in Chapter 3, *Traffic Policing*.

In addition to the commands explained in Chapter 1, *Introduction* and Chapter 2, *Configuring a QoS Policy-map*, binding a child policy-map to a parent class is done using the following command:

```
service-policy NAME
```

Notice that `NAME` represents the name of the child qos policy-map. This command is configurable on the class mode as shown in the example below:

```
(config)#policy-map Pmap-Parent-1
(config-pmap-qos)#class Cmap-Parent-1
(config-pmap-c-qos)#service-policy Pmap-Child-1
(config-pmap-c-qos)#exit
```

An example of creating parent and child class-maps and policy-maps, and configuring them for hierarchical policing is shown below:

Note: the qos-policer TCAM group must be enabled before binding a parent policy-map to an interface. For more information about the hardware filter groups, see the `hardware-profile filter` and `show hardware-profile filters` commands in the *System Management Guide.*

In the following example, traffic streams on VLAN ID 10, where one stream with CoS value 1 and another stream with CoS value 2 (received on interface xe1) are policed to a total of 40 mbps with 10 mbps of traffic being marked green for stream1 at 20 mbps, and traffic marked green for stream2. The remaining 10 mbps of traffic from stream1 and stream2 is marked yellow. Any remaining traffic will be dropped at ingress.

```
(config)#qos enable
(config)#class-map CUST-10
(config-cmap-qos)#match vlan 10
(config-cmap-qos)#exit
(config)#class-map CUST-10-C1
(config-cmap-qos)#match cos 1
(config-cmap-qos)#exit
(config)#class-map CUST-10-C2
(config-cmap-qos)#match cos 2
(config-cmap-qos)#exit
(config)#policy-map PC-CUST-10
(config-pmap-qos)#class CUST-10-C1
```

```
(config-pmap-c-qos)#police cir 10 mbps eir 30 mbps
(config-pmap-c-qos)#exit
(config-pmap-qos)#class CUST-10-C2
(config-pmap-c-qos)#police cir 20 mbps eir 20 mbps
(config-pmap-c-qos)#exit
(config-pmap-qos)#class class-default
(config-pmap-c-qos)#police cir 10 mbps eir 30 mbps
(config-pmap-c-qos)#exit
(config-pmap-qos)#exit
(config)#policy-map P-CUST-10
(config-pmap-qos)#class CUST-10
(config-pmap-c-qos)#police cir 40 mbps
(config-pmap-c-qos)#service-policy PC-CUST-10
(config-pmap-c-qos)#exit
(config-pmap-qos)#exit
(config)#hardware-profile filter qos-policer enable
(config)#interface xe1
(config-if)#service-policy type qos input P-CUST-10
(config-if)#exit
```

Configuring Hierarchical Policing per Attachment Circuit

In this section, a configuration example is provided for the use-case of configuring traffic policing per attachment circuit with additional traffic policing based on the class of traffic. The configuration helps in providing overall traffic policing per attachment circuit and additionally ensures that different classes of traffic are policed. Total CIR configurations for child classes must be ensured to match the police rate of parent and EIR values for child can be configured to take the available bandwidth when one or more child traffic is not received at any point in time.

Topology

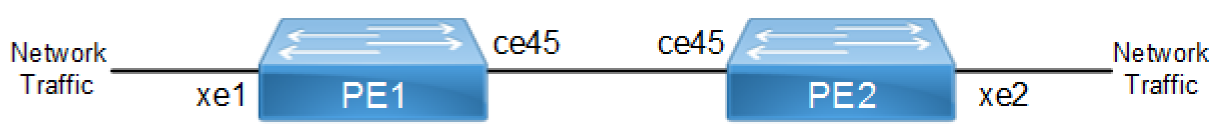


Figure 21-7: Topology for per-AC hierarchical policing

PE1

`PE1#configure terminal`	Enter configure terminal mode
`PE1(config)#hardware-profile filter ingress-ipv6 enable`	Enabling Ingress IPv6 group for IPv6 feature support
`PE1(config)#hardware-profile filter qos-policer enable`	Enabling Ingress extended QOS group for hierarchical policer
`PE1(config)#qos enable`	Enabling QoS
`PE1(config)#qos statistics`	Enabling Qos statistics
`PE1(config)#mpls l2-circuit VPWS-VLAN-200 2000200 21.21.21.21 agi abc saii xyz taii mmm`	Create an instance of an MPLS layer 2 virtual circuit

(config-pseudowire)#exit	Exit pseudowire config mode.
PE1(config)#mpls l2-circuit VPWS-VLAN-300 2000300 21.21.21.21	Create an instance of an MPLS layer 2 virtual circuit
(config-pseudowire)#exit	Exit pseudowire config mode.
PE1(config)#mpls lsp-model uniform	Configure the MPLS LSP model as Uniform.
PE1(config)#service-template VPWS-Vlan200-Service	Enabling Service Template
PE1(config-svc)#match outer-vlan 200	Match outer vlan 200
PE1(config-svc)#rewrite ingress translate 100 outgoing-tpid dot1.q	Enabling command to configure a match VLAN action for a service template
PE1(config-svc)#exit	Exit mode
PE1(config)#service-template VPWS-Vlan300-Service	Enabling Service Template
PE1(config-svc)#match outer-vlan 300	Match outer vlan 300<null>
PE1(config-svc)#rewrite ingress translate 4000 outgoing-tpid dot1.q	Enabling command to configure a match VLAN action for a service template
PE1(config-svc)#exit	Exit mode
PE1(config)#class-map c0	Enter Class-map mode
PE1(config-cmap-qos)#match cos 0	Matching cos value inside class map
PE1(config-cmap-qos)#exit	Exit mode
PE1(config)#class-map c1	Enter Class-map mode
PE1(config-cmap-qos)#match cos 1	Matching cos value inside class map
PE1(config-cmap-qos)#exit	Exit mode
PE1(config)#class-map c2	Enter Class-map mode
PE1(config-cmap-qos)#match cos 2	Matching cos value inside class map
PE1(config-cmap-qos)# exit	Exit mode
PE1(config)#class-map c3	Enter Class-map mode
PE1(config-cmap-qos)#match cos 3	Matching cos value inside class map
PE1(config)#class-map cp1	Enter Class-map mode
PE1(config-cmap-qos)#match vlan 200	Matching vlan inside class map
PE1(config-cmap-qos)#exit	Exit mode
PE1(config)#class-map cp2	Enter Class-map mode
PE1(config-cmap-qos)#match vlan 300	Matching vlan inside class map
PE1(config-cmap-qos)#exit	Exit mode
PE1(config)#policy-map child	Enter policy-map mode
PE1(config-pmap-qos)#class c0	Assign Class c0 to Policy-map child
PE1(config-pmap-c-qos)#police cir 50 mbps eir 150 mbps	Police packets @ Committed information rate 50 mbps
PE1(config-pmap-c-qos)#exit	Exit mode
PE1(config-pmap-qos)#class c1	Assign Class c1 to Policy-map child
PE1(config-pmap-c-qos)#police cir 50 mbps eir 150 mbps	Police packets @ Committed information rate 50 mbps
PE1(config-pmap-c-qos)#exit	Exit mode

© 2026 IP Infusion Inc. Proprietary

PE1(config-pmap-qos)#class c2	Assign Class c2 to Policy-map child
PE1(config-pmap-c-qos)#police cir 50 mbps eir 150 mbps	Police packets @ Committed information rate 50 mbps
PE1(config-pmap-c-qos)#exit	Exit mode
PE1(config-pmap-qos)#class c3	Assign Class c3 to Policy-map child
PE1(config-pmap-c-qos)#police cir 50 mbps eir 150 mbps	Police packets @ Committed information rate 50 mbps
PE1(config-pmap-c-qos)#exit	Exit mode
PE1(config-pmap-qos)#policy-map parent	Enter policy-map mode
PE1(config-pmap-qos)#class cp1	Assign Class cp1 to Policy-map parent
PE1(config-pmap-c-qos)#police cir 200 mbps	Police packets @ Committed information rate 200 mbps
PE1(config-pmap-c-qos)#service-policy child	Attaching child policy to Parent
PE1(config-pmap-c-qos)#exit	Exit mode
PE1(config-pmap-qos)#class cp2	Assign Class cp2 to Policy-map child
PE1(config-pmap-c-qos)#police cir 200 mbps	Police packets @ Committed information rate 200 mbps
PE1(config-pmap-c-qos)#service-policy child	Attaching child policy to Parent
PE1(config-pmap-c-qos)#exit	Exit mode
PE1(config-pmap-qos)#exit	Exit from Policy Map
PE1(config)#router ldp	Enter the Router LDP mode.
PE1(config-router)#targeted-peer ipv4 21.21.21.21	Configure targeted peer
PE1(config-router-targeted-peer)#<null>exit-targeted-peer-mode	Exit Targeted Peer Mode
PE1(config-router)#transport-address ipv4 11.11.11.11	Configure transport address
PE1(config-router)#exit	Exit mode
PE1(config)#interface ce45	Entering Interface Mode
PE1(config-if)#ip address 1.1.1.1/24	Assigning IP Address to interface
PE1(config-if)#label-switching	Enabling Label Switching on the interface.
PE1(config-if)#enable-ldp ipv4	Enabling LDP
PE1(config-if)#exit	Exit mode
PE1(config)#interface lo	Entering loopback Interface Mode
PE1(config-if)#ip address 11.11.11.11/32 secondary	Assigning IP Address to interface
PE1(config-if)#exit	Exit mode
PE1(config)#interface xe2	Entering Interface Mode
PE1(config-if)#switchport	Enabling the Interface as Switchport
PE1(config-if)#mpls-l2-circuit VPWS-VLAN-200 service-template VPWS-Vlan200-Service	Bind the interface to the VC and add the service-template.
PE1(config-if)#mpls-l2-circuit VPWS-VLAN-300 service-template VPWS-Vlan300-Service	Bind the interface to the VC and add the service-template.
PE1(config-if)#service-policy type qos input parent	Assign service-policy to interface on in-direction
PE1(config-if)#exit	Exit mode

PE1(config)#router ospf	Configure the routing process and The Process ID should be a unique positive integer identifying the routing process
PE1(config-router)#network 1.1.1.0/24 area 0.0.0.0	Define the interface on which OSPF runs and associate the area ID (0) with the interface.
PE1(config-router)#network 11.11.11.11/32 area 0.0.0.0	Define the interface on which OSPF runs and associate the area ID (0) with the interface.
PE1(config-router)#exit	Exit Router Ospf MOde

PE2 Configuration

PE2#configure terminal	Enter configure terminal mode
PE2(config)#hardware-profile filter ingress-ipv6 enable	Enabling Ingress IPv6 group for IPv6 feature
PE2(config)#hardware-profile filter qos-policer enable	Enabling Ingress extended QOS group for hierarchical policer
PE2(config)#qos enable	Enabling QoS
PE2(config)#qos statistics	Enabling Qos statistics
PE2(config)#mpls l2-circuit VPWS-VLAN-200 2000200 11.11.11.11 agi abc saii xyz taii mmm	Create an instance of an MPLS layer 2 virtual circuit
(config-pseudowire)#exit	Exit pseudowire config mode.
PE2(config)#mpls l2-circuit VPWS-VLAN-300 2000300 11.11.11.11	Create an instance of an MPLS layer 2 virtual circuit
(config-pseudowire)#exit	Exit pseudowire config mode.
PE2(config)#mpls lsp-model uniform	Configure the MPLS LSP model as Uniform.
PE2(config)#service-template VPWS-Vlan200-Service	Enabling Service Template
PE2(config-svc)#match outer-vlan 200	Match outer vlan 200
PE2(config-svc)#rewrite ingress translate 100 outgoing-tpid dot1.q	Enabling command to configure a match VLAN action for a service template
PE2(config-svc)#exit	Exit mode
PE2(config)#service-template VPWS-Vlan300-Service	Enabling Service Template
PE2(config-svc)#match outer-vlan 300	Match outer vlan 300
PE2(config-svc)#rewrite ingress translate 4000 outgoing-tpid dot1.q	Enabling command to configure a match VLAN action for a service template
PE2(config-svc)#exit	Exit mode
PE2(config)#class-map c0	Enter Class-map mode
PE2(config-cmap-qos)# match cos 0	Matching cos value inside class map
PE2(config-cmap-qos)#exit	Exit mode
PE2(config)#class-map c1	Enter Class-map mode
PE2(config-cmap-qos)#match cos 1	Matching cos value inside class map
PE2(config-cmap-qos)#exit	Exit mode
PE2(config)#class-map c2	Enter Class-map mode
PE2(config-cmap-qos)#match cos 2	Matching cos value inside class map

`PE2(config-cmap-qos)#exit`	Exit mode
`PE2(config)#class-map c3`	Enter Class-map mode
`PE2(config-cmap-qos)#match cos 3`	Matching cos value inside class map
`PE2(config-cmap-qos)#exit`	Exit mode
`PE2(config)#class-map cp1`	Enter Class-map mode
`PE2(config-cmap-qos)#match vlan 200`	Matching vlan inside class map
`PE2(config-cmap-qos)#exit`	Exit mode
`PE2(config)#class-map cp2`	Enter Class-map mode
`PE2(config-cmap-qos)#match vlan 300`	Matching vlan inside class map
`PE2(config-cmap-qos)#exit`	Exit mode
`PE2(config)#policy-map child`	Enter policy-map mode
`PE2(config-pmap-qos)#class c0`	Assign Class c0 to Policy-map child
`PE2(config-pmap-c-qos)#police cir 50 mbps eir 150 mbps`	Police packets @ Committed information rate 50 mbps
`PE2(config-pmap-c-qos)#exit`	Exit mode
`PE2(config-pmap-qos)#class c1`	Assign Class c1 to Policy-map child
`PE2(config-pmap-c-qos)#police cir 50 mbps eir 150 mbps`	Police packets @ Committed information rate 50 mbps
`PE2(config-pmap-c-qos)#exit`	Exit mode
`PE2(config-pmap-qos)#class c2`	Assign Class c2 to Policy-map child
`PE2(config-pmap-c-qos)#police cir 50 mbps eir 150 mbps`	Police packets @ Committed information rate 50 mbps
`PE2(config-pmap-c-qos)#exit`	Exit mode
`PE2(config-pmap-qos)#class c3`	Assign Class c3 to Policy-map child
`PE2(config-pmap-c-qos)#police cir 50 mbps eir 150 mbps`	Police packets @ Committed information rate 50 mbps
`PE2(config-pmap-c-qos)#exit`	Exit mode
`PE2(config-pmap-qos)#policy-map parent`	Enter policy-map mode
`PE2(config-pmap-qos)#class cp1`	Assign Class cp1 to Policy-map parent
`PE2(config-pmap-c-qos)#police cir 200 mbps`	Police packets @ Committed information rate 200 mbps
`PE2(config-pmap-c-qos)#service-policy child`	Attaching child policy to Parent
`PE2(config-pmap-c-qos)#exit`	Exit mode
`PE2(config-pmap-qos)#class cp2`	Assign Class cp2 to Policy-map child
`PE2(config-pmap-c-qos)#police cir 200 mbps`	Police packets @ Committed information rate 200 mbps
`PE2(config-pmap-c-qos)#service-policy child`	Attaching child policy to Parent
`PE2(config-pmap-c-qos)#exit`	Exit mode
`PE2(config-pmap-qos)#exit`	Exit from Policy Map
`PE2(config)#router ldp`	Enter the Router LDP mode
`PE2(config-router)#targeted-peer ipv4 11.11.11.11`	Configure targeted peer
`PE2(config-router-targeted-peer)#exit-targeted-peer-mode`	Exit Targeted Peer Mode

`PE2(config-router)# transport-address ipv4 21.21.21.21`	Configure transport address
`PE2(config-router)#exit`	Exit mode
`PE2(config)#interface ce45`	Entering Interface Mode
`PE2(config-if)#ip address 1.1.1.2/24`	Assigning IP Address to interface
`PE2(config-if)#label-switching`	Enabling Label Switching on the interface.
`PE2(config-if)#enable-ldp ipv4`	Enabling LDP
`PE2(config-if)#exit`	Exit mode
`PE2(config)#interface lo`	Entering loopback Interface Mode
`PE2(config-if)#ip address 21.21.21.21/32 secondary`	Assigning IP Address to interface
`PE2(config-if)#exit`	Exit interface mode
`PE2(config)#interface xe1`	Entering Interface Mode
`PE2(config-if)#switchport`	Enabling the Interface as Switch port
`PE2(config-if)#mpls-l2-circuit VPWS-VLAN-200 service-template VPWS-Vlan200-Service`	Bind the interface to the VC and add the service-template.
`PE2(config-if)#mpls-l2-circuit VPWS-VLAN-300 service-template VPWS-Vlan300-Service`	Bind the interface to the VC and add the service-template.
`PE2(config-if)#service-policy type qos input parent`	Assign service-policy to interface on in-direction
`PE2(config-if)#exit`	Exit interface mode
`PE2(config)#router ospf`	Configure the routing process and The Process ID should be a unique positive integer identifying the routing process
`PE2(config-router)#network 1.1.1.0/24 area 0.0.0.0`	Define the interface on which OSPF runs and associate the area ID (0) with the interface.
`PE2(config-router)#network 21.21.21.21/32 area 0.0.0.0`	Define the interface on which OSPF runs and associate the area ID (0) with the interface.
`PE2(config-router)#exit`	Exit Router OSPF mode

Validation

PE1

Validate LDP session using following show command:

```
PE1#show ldp session
Peer IP Address           IF Name    My Role    State       KeepAlive  UpTime
21.21.21.21               ce45       Passive   OPERATIONAL   30    00:05:40
```

Validate virtual circuit status using following command:

```
PE1#show mpls vc-table
VC-ID    VID  Inner-VID  Ac-Intf  Nw-Intf  Out Label  Tunnel-Label  Nexthop      Status
2000200  N/A  N/A        xe2      ce45     24320      3             21.21.21.21  Active
2000300  N/A  N/A        xe2      ce45     24321      3             21.21.21.21  Active
```

Validate QoS configuration and statistics using below commands:

```
PE1#show policy-map interface xe2

Interface xe2
Type QoS statistics status : enabled

Service-policy (qos) input: parent
----------------------------
Class-map (qos): cp1 (match all)
 match vlan 200
 police cir 200 mbps
 shape 0 kbps (inherited)

     Child Service-policy (qos) : child
     ----------------------------
     Class-map (qos): c0 (match all)
      match cos 0
      police cir 50 mbps eir 150 mbps
           matched      : 168530 packets, 252795000 bytes
           transmitted  : 85295 packets, 127942500 bytes
           dropped      : 83235 packets, 124852500 bytes

     Class-map (qos): c1 (match all)
      match cos 1
      police cir 50 mbps eir 150 mbps
           matched      : 168523 packets, 252784500 bytes
           transmitted  : 85286 packets, 127929000 bytes
           dropped      : 83237 packets, 124855500 bytes

     Class-map (qos): c2 (match all)
      match cos 2
      police cir 50 mbps eir 150 mbps
           matched      : 168558 packets, 252837000 bytes
           transmitted  : 85320 packets, 127980000 bytes
           dropped      : 83238 packets, 124857000 bytes

     Class-map (qos): c3 (match all)
      match cos 3
      police cir 50 mbps eir 150 mbps
           matched      : 168550 packets, 252825000 bytes
           transmitted  : 85310 packets, 127965000 bytes
           dropped      : 83240 packets, 124860000 bytes

     Class-map (qos): class-default (match any)

Class-map (qos): cp2 (match all)
 match vlan 300
 police cir 200 mbps
 shape 0 kbps (inherited)

     Child Service-policy (qos) : child
```

```
     ----------------------------
     Class-map (qos): c0 (match all)
      match cos 0
      police cir 50 mbps eir 150 mbps
           matched      : 168597 packets, 252895500 bytes
           transmitted  : 85336 packets, 128004000 bytes
           dropped      : 83261 packets, 124891500 bytes

     Class-map (qos): c1 (match all)
      match cos 1
      police cir 50 mbps eir 150 mbps
           matched      : 168589 packets, 252883500 bytes
           transmitted  : 85327 packets, 127990500 bytes
           dropped      : 83262 packets, 124893000 bytes

     Class-map (qos): c2 (match all)
      match cos 2
      police cir 50 mbps eir 150 mbps
           matched      : 168625 packets, 252937500 bytes
           transmitted  : 85358 packets, 128037000 bytes
           dropped      : 83267 packets, 124900500 bytes

     Class-map (qos): c3 (match all)
      match cos 3
      police cir 50 mbps eir 150 mbps

     Class-map (qos): class-default (match any)
           matched      : 168589 packets, 252883500 bytes
           transmitted  : 85333 packets, 127999500 bytes
           dropped      : 83256 packets, 124884000 bytes

Service-policy (queuing) output: default-out-policy
Interface Bandwidth 1000000 kbps
-----------------------------------
Class-map (queuing): q0
 shape 1000000 kbps (inherited)
 priority level 0
 queue-limit 1048576 bytes/8 ms (default)

Class-map (queuing): q1
 shape 1000000 kbps (inherited)
 priority level 1
 queue-limit 1048576 bytes/8 ms (default)

Class-map (queuing): q2
 shape 1000000 kbps (inherited)
 priority level 2
 queue-limit 1048576 bytes/8 ms (default)

Class-map (queuing): q3
```

```
 shape 1000000 kbps (inherited)
 priority level 3
 queue-limit 1048576 bytes/8 ms (default)

Class-map (queuing): q4
 shape 1000000 kbps (inherited)
 priority level 4
 queue-limit 1048576 bytes/8 ms (default)

Class-map (queuing): q5
 shape 1000000 kbps (inherited)
 priority level 5
 queue-limit 1048576 bytes/8 ms (default)

Class-map (queuing): q6
 shape 1000000 kbps (inherited)
 priority level 6
 queue-limit 1048576 bytes/8 ms (default)

Class-map (queuing): q7
 shape 1000000 kbps (inherited)
 priority level 7
 queue-limit 1048576 bytes/8 ms (default)


PE1#show policy-map statistics
Type qos class-map statistics:
+---------------+----------------+----------------+----------------+------------------+
|   Class-map   |   Match pkts   |  Match bytes   |  Dropped pkts  |   Dropped Bytes  |
+---------------+----------------+----------------+----------------+------------------+
xe2
 cp1
   c0           231860           347790000        114528           171792000
   c1           231854           347781000        114531           171796500
   c2           231891           347836500        114533           171799500
   c3           231884           347826000        114534           171801000
 cp2
   c0           231933           347899500        114557           171835500
   c1           231926           347889000        114558           171837000
   c2           231963           347944500        114563           171844500
   class-default 231930          347895000        114555           171832500

Type queuing class-map statistics:
+---------------+----------------+----------------+----------------+------------------+
|   Class-map   |   Total pkts   |  Total bytes   |  Dropped pkts  |   Dropped Bytes  |
+---------------+----------------+----------------+----------------+------------------+
ce45
 q0             5595949          8393932500       229131           343705500
 q1             5596049          8394085500       229228           343851000
 q2             5596147          8394232500       229327           343999500
```

```
 q3             2798111          4197172500       114702           172057500
 q4             2798195          4197300000       114762           172147500
 q6             23               1886             0                0
 q7             249              21339            0                0
```

PE2

Validate LDP session using following show command:

```
PE2#show ldp session
Peer IP Address           IF Name    My Role    State      KeepAlive  UpTime
11.11.11.11               ce45       Active    OPERATIONAL   30    00:06:59
```

Validate virtual circuit status using following command:

```
PE2#show mpls vc-table
VC-ID    VID  Inner-VID  Ac-Intf  Nw-Intf  Out Label  Tunnel-Label  Nexthop      Status
2000200  N/A  N/A        xe1      ce45     24320      3             11.11.11.11  Active
2000300  N/A  N/A        xe1      ce45     24321      3             11.11.11.11  Active
```

Validate QoS configuration and statistics using below commands:

```
PE2#show policy-map interface xe1

Interface xe1
Type QoS statistics status : enabled

Service-policy (queuing) output: default-out-policy
Interface Bandwidth 1000000 kbps
-----------------------------------
Class-map (queuing): q0
 shape 1000000 kbps (inherited)
 priority level 0
 queue-limit 1048576 bytes/8 ms (default)
      Output
         Total        : 6205481 packets, 9419929266 bytes
         Green        : 6205548 packets, 9420032490 bytes
         Yellow       : 0 packets, 0 bytes

Class-map (queuing): q1
 shape 1000000 kbps (inherited)
 priority level 1
 queue-limit 1048576 bytes/8 ms (default)
      Output
         Total        : 6205580 packets, 9420081066 bytes
         Green        : 6205651 packets, 9420188844 bytes
         Yellow       : 0 packets, 0 bytes

Class-map (queuing): q2
 shape 1000000 kbps (inherited)
 priority level 2
 queue-limit 1048576 bytes/8 ms (default)
      Output
```

© 2026 IP Infusion Inc. Proprietary

```
         Total        : 6205681 packets, 9420260190 bytes
         Green        : 6205798 packets, 9420410472 bytes
         Yellow       : 0 packets, 0 bytes

Class-map (queuing): q3
 shape 1000000 kbps (inherited)
 priority level 3
 queue-limit 1048576 bytes/8 ms (default)
      Output
         Total        : 3102900 packets, 4710206754 bytes
         Green        : 3102934 packets, 4710258366 bytes
         Yellow       : 0 packets, 0 bytes

Class-map (queuing): q4
 shape 1000000 kbps (inherited)
 priority level 4
 queue-limit 1048576 bytes/8 ms (default)
      Output
         Total        : 3102984 packets, 4710337302 bytes
         Green        : 3103019 packets, 4710388914 bytes
         Yellow       : 0 packets, 0 bytes

Class-map (queuing): q5
 shape 1000000 kbps (inherited)
 priority level 5
 queue-limit 1048576 bytes/8 ms (default)

Class-map (queuing): q6
 shape 1000000 kbps (inherited)
 priority level 6
 queue-limit 1048576 bytes/8 ms (default)

Class-map (queuing): q7
 shape 1000000 kbps (inherited)
 priority level 7
 queue-limit 1048576 bytes/8 ms (default)


PE2#show policy-map statistics
Type qos class-map statistics:
+----------------+----------------+----------------+----------------+------------------+
|   Class-map    |   Match pkts   |  Match bytes   |  Dropped pkts  |   Dropped Bytes  |
+----------------+----------------+----------------+----------------+------------------+

Type queuing class-map statistics:
+----------------+----------------+----------------+----------------+------------------+
|   Class-map    |   Total pkts   |  Total bytes   |  Dropped pkts  |   Dropped Bytes  |
+----------------+----------------+----------------+----------------+------------------+
ce45
 q6             29               2378             0                0
```

```
 q7              295             25051           0               0
xe1
 q0              6339725         9623710140      0               0
 q1              6339793         9623811846      0               0
 q2              6339857         9623908998      0               0
 q3              3169947         4811984100      0               0
 q4              3170018         4812088842      0               0
```

CHAPTER 22 Subinterface Queuing

In Qumran devices, every physical port has default eight priority queues and subinterface has default 4 priority queues when service-queue profile1 is active and 8 queues (distributed between 4 priorities) when service-queue profile2 is active. Physical port queues are created during initialization while the subinterface queues will be created/deleted when encap is set/unset respectively on subinterface. Whenever QoS feature is enabled, all priority queues of the physical ports/subinterafce will be configured with certain default egress queuing parameters.

In order to customize the treatment on the priority queues, queuing policy-map infrastructure need to be used.

Configuring Subinterface Queues

Subinterface queues are nothing different then the physical ports queues expect that number of queues assigned to a subinterface can be set via profile and by-default profile1 is set which sets 4 queues to be created for services.

User can configure the service-queue profile via cli "hardware-profile service-queue (profile1 | profile2)".

Profile1 supports 4 new queues creation for services,which is also a default profile. Profile2 supports 8 new queues creation for services.

These queues will be created or deleted when the encap is set or unset on a subinterface respectively. Like and other interface, subinterface has a default ingress egress mapping profile. i.e. dscp-to-queue and dscp-color-to-dscp respectively. Subinterface has default queuing-policy in order to support QoS treatment on the queues.

Mapping profiles (dscp-to-queue and dscp-color-to-dscp) maps packets dscp to/from 8 traffic classes. When the hardware-queues created are 4, 8 traffic classes will be mapped to these 4 hardware-queues implicitly as shown in Table 22-13:

Table 22-13: Traffic class to queue mapping

Traffic class	Queue
0	0
1, 2, 3	1
4, 5	2
6-7	3

This map can be checked via this command:

```
show queue remapping
```

Output:

```
Port queue remapping:
+------------+----------------------+
|  Queue/tc  |     hardware-queue   |
+------------+----------------------+
|      0     |          0           |
|      1     |          1           |
|      2     |          2           |
|      3     |          3           |
|      4     |          4           |
|      5     |          5           |
|      6     |          6           |
```

```
|      7     |          7            |
+------------+-----------------------+

Service queue remapping:
+------------+-----------------------+
|  Queue/tc  |     hardware-queue    |
+------------+-----------------------+
|      0     |          0            |
|      1     |          1            |
|      2     |          1            |
|      3     |          1            |
|      4     |          2            |
|      5     |          2            |
|      6     |          3            |
|      7     |          3            |
+------------+-----------------------+
```

When the number of profile 2 is active, number of new queues created will be 8 and the traffic class to hardware queues map will be one to one.

This map can be checked via this command:

```
show queue remapping
```

Output:

```
Port queue remapping:
+------------+-----------------------+
|  Queue/tc  |     hardware-queue    |
+------------+-----------------------+
|      0     |          0            |
|      1     |          1            |
|      2     |          2            |
|      3     |          3            |
|      4     |          4            |
|      5     |          5            |
|      6     |          6            |
|      7     |          7            |
+------------+-----------------------+

Service queue remapping:
+------------+-----------------------+
|  Queue/tc  |     hardware-queue    |
+------------+-----------------------+
|      0     |          0            |
|      1     |          1            |
|      2     |          2            |
|      3     |          3            |
|      4     |          4            |
|      5     |          5            |
|      6     |          6            |
|      7     |          7            |
+------------+-----------------------+
```

Configuring Default Queuing Policy-Map

When the QoS feature is enabled, all subinterfaces is supplied with a default policy-map of queuing type. The default policy-map is created with the name "default-subif-out-policy" which is reserved and modifying parameters in this policy-map is reflected on all subinterfaces that do not have customized queuing policy-maps. Customized queuing policy-maps can be created and bound to subinterface to treat subinterface differently from the default configuration.

Default queuing-policy can be accessed via following cli:

```
policy-map type queuing default subif-default-out-policy
```

Classes mapped to default queues can be accessed via the following commands.

In case of profile1:

```
class type queuing default (q0|q1|q2|q3)
```

In case of profile2:

```
class type queuing default (q0|q1|q2|q3|q4|q5|q6|q7)
```

User can configure all the queue parameters like shaping, scheduling, wred, taildrop same as on port. Show commands to verify the config and stats are same. For these configurations please check respective chapters as described in the document.

Displaying Policy-Map Configuration

With profile1 enabled, 4 priority level 0-3 will be configured by default on 4 queues as below:

- Queue 0: priority level 0
- Queue 1: priority level 1
- Queue 2: priority level 2
- Queue 3: priority level 3

```
(config)#show policy-map interface xe11.1

Interface xe11.1

Type Queuing policy-map : subif-default-out-policy

Class-map (queuing): q0
 shape 1000000 kbps (inherited)
 priority level 0
 queue-limit 1253376 bytes/10 ms (default)
Class-map (queuing): q1
 shape 1000000 kbps (inherited)
 priority level 1
 queue-limit 1253376 bytes/10 ms (default)
      Output
         Total        : 4109055 packets, 279534060 bytes
         Green        : 4120123 packets, 280222424 bytes
         Yellow       : 0 packets, 0 bytes
      Rate            : 768646.000 kbps
```

© 2026 IP Infusion Inc. Proprietary

```
Class-map (queuing): q2
 shape 1000000 kbps (inherited)
 priority level 2
 queue-limit 1253376 bytes/10 ms (default)

Class-map (queuing): q3
 shape 1000000 kbps (inherited)
 priority level 3
 queue-limit 1253376 bytes/10 ms (default)
```

With profile2 enabled, 4 priority level 0-3 will be configured by default on 8 queues as below:

- Queue 0: priority level 0
- Queue 1: priority level 1
- Queue 2: priority level 1
- Queue 3: priority level 1
- Queue 4: priority level 2
- Queue 5: priority level 2
- Queue 6: priority level 3
- Queue 7: priority level 3

```
(config)#show policy-map interface xe11.1

Interface xe11.1

Type Queuing policy-map : subif-default-out-policy

Class-map (queuing): q0
 shape 1000000 kbps (inherited)
 priority level 0
 queue-limit 1253376 bytes/10 ms (default)

Class-map (queuing): q1
 shape 1000000 kbps (inherited)
 priority level 1
 queue-limit 1253376 bytes/10 ms (default)

Class-map (queuing): q2
 shape 1000000 kbps (inherited)
 priority level 1
 queue-limit 1253376 bytes/10 ms (default)

Class-map (queuing): q3
 shape 1000000 kbps (inherited)
 priority level 1
 queue-limit 1253376 bytes/10 ms (default)
```

```
Class-map (queuing): q4
 shape 1000000 kbps (inherited)
 priority level 2
 queue-limit 1253376 bytes/10 ms (default)

Class-map (queuing): q5
 shape 1000000 kbps (inherited)
 priority level 2
 queue-limit 1253376 bytes/10 ms (default)

Class-map (queuing): q6
 shape 1000000 kbps (inherited)
 priority level 3
 queue-limit 1253376 bytes/10 ms (default)

Class-map (queuing): q7
 shape 1000000 kbps (inherited)
 priority level 3
 queue-limit 1253376 bytes/10 ms (default)
```

Creating a User-Defined Queuing Policy-Map

Qumran supports the creation of customized policy-map in which all 4 priority queues can be accessed. The following is the command to create a customized default policy-map:

```
(no|) policy-map type queuing NAME
```

Class-maps can be configured matching queues via following cli:

```
(no|) class-map type queuing NAME
```

Match queue/queues in the class-map via following cli :

```
(no|) match queue <0-7>
```

Note: The match queue range 0-7 is valid only for port queues classification when service-queue profile1 is active.

In case of profile1, service queues can be matched from 0-3 only.

For service queues/subinterface, the valid range is 0-3 with service-queue profile1 (4 queue per subinterface) and 0-7 with profile2 (8 queue per subinterface).

Once the policy-map and class-maps are configured, class-maps can be configured in the policy-map with the following command:

```
class type queuing NAME
```

Binding a User-Defined Queuing Policy-Map

Customized queuing policy-maps take affect only when the configuration is bound to an interface. Queuing policy-maps can be bound to the port with the following command:

```
service-policy type queuing output NAME
```

Where `NAME` represents the name of the queuing policy-map.

© 2026 IP Infusion Inc. Proprietary

For example:

```
class-map type queuing data
 match queue 0
!
class-map type queuing signal
 match queue 3
!
class-map type queuing voice
 match queue 1
!
policy-map type queuing configPolicy1
 class type queuing class-default-q
  exit
 class type queuing data
  exit
 class type queuing signal
  exit
 class type queuing voice
  exit
!
interface xe11.1
 service-policy type queuing output configPolicy1
```

Queue(s) which are not matched in any class in a user-defined policy-map, will be mapped to Class-default-q by default. This class-default-q is a by-default created class in a user-defined policy-map.

Displaying Policy-Map Configuration

```
(config-if)#do show policy-map in xe11.1

Interface xe11.1

Type Queuing policy-map : configPolicy1

Class-map (queuing): class-default-q
 shape 1000000 kbps (inherited)
 wfq-queue weight 1
 queue-limit 1253376 bytes/10 ms (default)
 match queue 2

Class-map (queuing): data
 shape 1000000 kbps (inherited)
 wfq-queue weight 1
 queue-limit 1253376 bytes/10 ms (default)
 match queue 0

Class-map (queuing): signal
 shape 1000000 kbps (inherited)
 wfq-queue weight 1
 queue-limit 1253376 bytes/10 ms (default)
```

```
 match queue 3

Class-map (queuing): voice
 shape 1000000 kbps (inherited)
 wfq-queue weight 1
 queue-limit 1253376 bytes/10 ms (default)
 match queue 1
      Output
         Total        : 2321147 packets, 157909736 bytes
         Green        : 2337514 packets, 158952312 bytes
         Yellow       : 0 packets, 0 bytes
      Rate            : 773130.375 kbps
```

Displaying Policy-Map Rate Statistics

```
#show policy-map statistics type queuing rate mbps
+---------------------------------+--------------------+
|            Class-map            |    Rate (in mbps)  |
+---------------------------------+--------------------+
xe11.1
 voice (q1)                        773.104
#show policy-map statistics type queuing rate kbps
+---------------------------------+--------------------+
|            Class-map            |    Rate (in kbps)  |
+---------------------------------+--------------------+
xe11.1
 voice (q1)                        772400.062
#show policy-map statistics type queuing rate gbps
+---------------------------------+--------------------+
|            Class-map            |    Rate (in gbps)  |
+---------------------------------+--------------------+
xe11.1
 voice (q1)                        0.774
```

Displaying Interface Queue Counters

```
#show interface xe11.1 counters queue-stats
E - Egress, I - Ingress, Q-Size is in bytes
+------------------+--------+----------+-----------+-----------------+---------------+
| Queue/Class-map | Q-Size |  Tx pkts | Tx bytes  |  Dropped pkts   | Dropped bytes|
+------------------+--------+----------+-----------+-----------------+---------------+
q0              (E) 1253376  0          0            0                 0
q1              (E) 1253376  1402466359 95367712820   0                 0
q2              (E) 1253376  0          0            0                 0
q3              (E) 1253376  0          0            0                 0
```

Configuration Considerations

- Max 1 level of user defined hierarchy is supported on subinterface.
- Only match queue is allowed in the class in user-define queuing policy-map.
- In user-defined policy-map, all the classes will be in wfq scheduling manner.
- Class-default-q is a self-created class map as part a policy map. It cannot be created nor be destroyed. It will be displayed only when user access it. Executing command "no class-default-q", will un-configure all the configurations of class-default-q.
- User can configure all queuing parameters like weight, priority, queue-limit, wred and shape in a class inside policy.
- Valid priority range is 0-3 and match queue will be 0-3 in case of profile1 and 0-7 in case of profile2.
- Update is possible in the policy-map except the update of match criteria. Once the class with some match criteria is used in a policy-map, it cannot be updated.
- Subinterface queuing can be achieved via vlan-shaping (match interface) as well as via default queues.
- User-defined policy with match subinterface can only be attached on parent interface if subinterface is not attached to user-defined policy-map.
- If user-defined policy with match subinterface is attached on parent interface, sub-interface's default policy-map and port shaper will be removed implicitly from subinterface.
- If the user-defined-policy is applied on parent interface matching subinterface, traffic will go to the queues created via user-defined-service-policy and the queue stats for subinterface will only be displayed via service-policy. The subinterfaces not matched in the user-defined-service policy will go to their own queues only and not to class-default as happens in case of vlan shaping.
- On encap delete from subinterface, all the qos configuration will be removed implicitly from the subinterafce.
- If the port shaper is applied on parent port and on subinterface as well, minimum shape rate will take effect.
- Queue shape percent for subinterface queues, will be calculated as in following manner:
- Percentage will be calculated from the Effective Max speed, which will be calculated as follows:

    max_speed = parent_ifp->speed.

  If shaper is applied on parent port:

    max_speed = parent_port_shaper

  If shaper is applied on subinterface:

    max_speed = subinterface_shaper

- When only subinterfaces are created with default or no queuing policy-map (when qos is disabled), max supported number of services in case of 4-queue profiles and 8-queue profile are as follows:

  QMX : With 4 queues 8K services can be supported but total number of services will be half of the limit supported in case of 8 queues i.e. 4K services approx.

  QAX : With 4 queues 4K services can be supported but total number of services will be half of the limit supported in case of 8 queues i.e. 2K services approx.

  QUX : With 4 queues 2K services can be supported but total number of services will be half of the limit supported in case of 8 queues i.e. 1K services approx.

- In the case of user-defined queuing, each class represents 1 scheduler each, so it needs to be taken into consideration while configuring max number of services that can be supported with user-defined policy.
- Max number of scheduler and connectors are as follows:

QMX: BCM88370_B0

MAX_NUM_CL_SCHD_QMX 16384

MAX_NUM_FQ_SCHD_QMX 16128

QAX: BCM88470

MAX_NUM_CL_SCHD_QAX 7936

NUM_FQ_SCHD_QAX 7936

QUX: BCM88270

MAX_NUM_CL_SCHD_QUX 4096

MAX_NUM_FQ_SCHD_QUX 3840

Max number of Voqs that can be created will be derived from below formula:

MAX_NUM_VOQ_CONN_QMX (((64*1024)/NUM_CUSTOM_QUEUE)/2)

MAX_NUM_VOQ_CONN_QAX (((32*1024)/NUM_CUSTOM_QUEUE)/2)

MAX_NUM_VOQ_CONN_QUX (((16*1024)/NUM_CUSTOM_QUEUE)/2)

Where NUM_CUSTOM_QUEUE will be based upon service-queue profile set. i.e. 4 for profile1 and 8 for profile2.

Scheduler and queues used for physical interfaces are included in these.

- Subinterface queuing is not supported for lag subinterface and L2VPN/ELINE services on Q1 devices (QMX/QAX/QUX).
- When multiple traffic class is mapped to single service queue i.e tc 1,2,3 are mapped to queue1, traffic load shared between the traffic class inside the particular queue randomly.

Quality of Service Command Reference

CHAPTER 1 Quality of Service Commands

This chapter is a reference for the ingress Quality of Service (QoS) and hierarchical QoS commands.

class-map type

Use this command to create a class-map of type `qos` or `queuing`.

Use the `no` command to remove a class-map.

Note: A class-map without any match qualification behaves similar to a default class by matching all the packets on the interface it is attached via a service policy.

Note: In a class-map, adding a match criteria will have silent exit and will not proceed with operation.

Note: In match-all class-map, only a single value with any criteria can exist. Adding a new value for the existing criteria will update the same rule. Multiple values with the same criteria can be added in a “match-any” class-map.

Note: QoS class-maps of type match-any cannot be used with Hierarchical policy-maps.

Command Syntax

```
class-map type queuing (match-any|match-all|) NAME
no class-map type queuing NAME
class-map type qos (match-any|match-all|) NAME
no class-map type qos NAME
```

Parameters

`qos`	Ingress/egress class map.
`queuing`	Egress class map.
`NAME`	Specify the class map name (maximum length 32 characters)
`match-any`	Match any parameter (boolean OR)
`match-all`	Match all parameters (boolean AND)

Default

By default, the type is `qos`.

By default, the match type is `match-all`.

Command Mode

Configuration Mode

Applicability

This command was introduced in OcNOS version 3.0.

Examples

```
#configure terminal
(config)#class-map type qos C_QOS1
(config-cmap-qos)#
```

class type qos

Use this command to add a QoS class-map to a qos policy map.

Use the `no command` to remove a QoS class-map from the policy map.

Note: Priority of class in policy-map is as follow:

- Match-all" class-maps will have priorities equal to number of match types specified in class-map “Match-any.”
- Will have lowest priority which will be the same as priority of “match-all” with one match type.
- Only policer action is supported in the class having “match access-grp” criteria in match-any class-map.

Command Syntax

```
class (type qos|) (NAME|class-default)
no class (type qos|) (NAME|class-default)
```

Parameters

NAME	Specify the class map name

Default

By default, class is type qos

Command Mode

Policy-map mode

Applicability

This command was introduced in OcNOS version 3.0.

Examples

```
(config)#
(config)#policy-map type qos PP
(config-pmap-qos)#class type qos C_PP_1
(config-pmap-c-qos)#
```

class type queuing

Use this command to add a queuing class-map to a queuing policy map.

Default queuing class cannot be deleted.

Command Syntax

```
class type queuing default (q0|q1|q2|q3|q4|q5|q6|q7)
class type queuing (NAME|class-default-q)
```

Parameters

`default`	Default queue of the port
`q0`	Queue 0 and priority 0 lowest
`q1`	Queue 1 and priority 1
`q1`	Queue 2 and priority 2
`q1`	Queue 3 and priority 3
`q1`	Queue 4 and priority 4
`q1`	Queue 5 and priority 5
`q1`	Queue 6 and priority 6
`q1`	Queue 7 and priority 7 highest
`class-default-q`	System default class matching otherwise unclassified packets

Default

No default value is specified

Command Mode

Policy Map type queuing Mode

Applicability

This command was introduced in OcNOS version 3.0.

Examples

```
(config)#
(config)#policy-map type queuing default default-out-policy
(config-pmap-que)#class type queuing default q0
(config-pmap-c-que-def)#
```

© 2026 IP Infusion Inc. Proprietary

clear qos statistics

Use this command to clear the quality of service (QoS) statistics.

Command Syntax

```
clear qos statistics (interface IFNAME|) ((type qos|type qos input|type qos
  output|type queuing|type all)|)
```

Parameters

`interface IFNAME`	(Optional) Specifies the interface name for which to clear QoS statistics (maximum length 32 characters).
`type qos`	(Optional) Clears QoS statistics
`type qos input`	(Optional) Clears QoS input statistics
`type qos output`	(Optional) Clears QoS output statistics
`type queuing`	(Optional) Clears queuing class statistics
`type all`	(Optional) Clears all QoS and queuing statistics

Default

if no parameters are configured, the command will clear type QoS and type queuing class statistics on all interfaces.

Command Mode

Privileged Exec mode

Applicability

Introduced in OcNOS version 3.0.

Examples

To clear QoS statistics on all interfaces, use the following command:

```
OcNOS#clear qos statistics
```

© 2026 IP Infusion Inc. Proprietary

clear interface counters

Use this command to clear all the interface counters.

Command Syntax

```
clear interface counters
```

Parameters

NA

Default

Command Mode

Privileged Exec mode

Applicability

This command was introduced in OcNOS version 3.0.

Examples

```
#clear interface counters
#
```

l2 queue exp (Qumran)

Use this command to map a queue color to the some exp value for L2 traffic.

Use the `no` form of this command to remove the map.

Note: Color is an optional parameter. If not provided, the same map is set for all colors.

Command Syntax

```
l2 queue <0-7> (color (green|yellow|red|all)|) exp <0-7>
no l2 queue <0-7> (color (green|yellow|red|all|)|)
```

Parameters

`<0-7>`	Identifying queue number
`color`	(green | yellow | red|all)
`<0-7>`	EXP value

Default

By default, queue values are one-to-one mapped to Exp. For example, queue 0 exp 0, queue 1 exp 1, and so on.

Command Mode

config-egress-exp-encap-map

Applicability

This command was introduced in OcNOS version 3.0 and this command is applicable for Qumran.

Example

```
(config)#qos profile exp-encap default
(config-egress-exp-encap-map)#l2 queue 2 exp 1
(config-egress-exp-encap-map)#l2 queue 4 color yellow exp 3
```

Color is an optional parameter, so if users do not provide color for all colors, the same EXP is set. If users provide color, then only that specific color egress map is changed. If, however, users provide a map for all colors, as well as maps without color, the map with color will take priority.

For example,

```
(config-egress-exp-encap-map)#no l2 queue 1 color green
(config-egress-exp-encap-map)#no l2 queue 1
```

If a user wants to remove all the mapping of queue (for all colors), then there is no need to provide color, else the user can provide a specific color to remove a specific map.

© 2026 IP Infusion Inc. Proprietary

egress l3 exp encap map (Qumran)

Use this command to map a DSCP value to the some EXP value.

For L3 traffic, use the `no` form of this command to remove the map.

Command Syntax

```
l3 dscp <0-63> exp <0-7>
no l3 dscp <0-63>
```

Parameters

`<0-63>`	DSCP value
`<0-7>`	EXP value

Default

By default, 8 DSCP values are mapped to one exp.

For example: DSCP 0-7 exp 0, DSCP 8-15 exp 1.

Command Mode

config-egress-exp-encap-map

Applicability

This command was introduced in OcNOS version 3.0 and this command is applicable for Qumran.

Example

```
(config)#qos profile exp-encap default
(config-egress-exp-encap-map)#l3 dscp 20 exp 1
(config-egress-exp-encap-map)#l3 dscp 40 exp 2
```

queue cos4

Use this command to map a queue value to the CoS value.

Use the `no` form of this command to remove the map.

Note: Egress CoS map will be taking effect only when remark CoS is enabled.

Note: Color is an optional parameter. If not provided, the same map will be set for all colors.

Command Syntax

```
queue <0-7> (color (green|yellow|red|all)|) cos <0-7>
no queue <0-7> (color (green|yellow|red|all|)|)
```

Parameters

`<0-7>`	Identifying queue number
`color`	(green|yellow|red|all)
`<0-7>`	CoS value

Default

By default, CoS to queue mapping is one to one.

Command Mode

config-egress-cos-map

Applicability

This command was introduced in OcNOS version 3.0.

Example

```
(config)#qos profile queue-color-to-cos default
(config-egress-cos-map)#queue 1 color green cos 5
(config-egress-cos-map)#queue 1 cos 6
```

Color is an optional parameter. So if users do not provide color, for all colors same CoS remarking will be set. If user provides color, then only for that specific color egress map will be changed. If user provides a map for all colors as well as without color, a map with color will take priority.

```
(config-egress-cos-map)#no queue 1 color green
(config-egress-cos-map)#no queue 1
```

If user wants to remove all the mapping of queue (for all colors) no need to provide color, else can provide a specific color to remove a specific map.

egress dscp map

Use this command to map a queue to a DSCP value.

Use the `no` form of this command to remove the map.

Note: Egress DSCP map will be taking effect only when remark DSCP is enabled.

Note: Color is an optional parameter. If not provided, the same map will be set for all colors.

Command Syntax

```
queue <0-7> (color (green|yellow|red|all)|) dscp <0-63>
no queue <0-7> (color (green|yellow|red|all|)|)
```

Parameters

`<0-7>`	Identifying queue number
`color`	(green|yellow|red|all)
`<0-63>`	DSCP value

Default

By default, 8 DSCP values are mapped to one queue.

For example: DSCP 0-7 queue 0, DSCP 8-15 queue 1.

Command Mode

config-egress-dscp-map

Applicability

This command was introduced in OcNOS version 3.0.

Example

```
(config)#qos profile dscp-to-dscp default
(config-egress-dscp-map)#dscp 20 color yellow dscp 40
(config-egress-dscp-map)#dscp 20 dscp 36
```

If user specifies color in map then map will be set for that specific color else for all colors same map will be set.

```
(config-egress-dscp-map)#no dscp 20 color yellow
(config-egress-dscp-map)#no dscp 20
```

If user wants to remove all the mapping of dscp (for all colors) no need to provide color, else can provide specific color to

remove specific map.

ingress cos map

Use this command to map a CoS value to the queue.

Use the `no` form of this command to remove the map.

Note: Color mapping is fixed: DEI 0 will be mapped to color “green” and DEI 1 will be mapped to color “yellow.”

Command Syntax

```
cos <0-7> queue <0-7>
no cos <0-7>
```

Parameters

`<0-7>`	CoS value
`<0-7>`	Identifying queue number

Default

By default, CoS to queue mapping is one to one.

Command Mode

config-ingress-cos-map

Applicability

This command was introduced in OcNOS version 3.0.

Example

```
(config)#qos profile cos-to-queue default
(config-ingress-cos-map)#cos 1 queue 2
```

© 2026 IP Infusion Inc. Proprietary

ingress dscp map

Use this command to map a DSCP value to the queue.

Use the `no` form of this command to remove the map.

Note: You can "remark" the DSCP value for the incoming traffic at ingress via map by setting the last "DSCP" field. You still need to match traffic with the incoming DSCP value in the class-map even if you are remarking them at ingress via a DSCP map. Traffic will be received with remarked DSCP value at egress if no egress DSCP map is applied. If an egress DSCP map (such as DSCP-to-DSCP) is applied at the egress port, and DSCP remarking is enabled, DSCP will be remarked again according to the mapping given for the remarked DSCP (the DSCP value given in ingress DSCP map).

Note: Last "DSCP" parameter is remark_DSCP value, which if provided will be set as the same ingress DSCP value. Color is an optional parameter. If not provided, it will be set as the default value. Please refer the Configuration Guide for the default value table.

Command Syntax

```
dscp <0-63> queue <0-7> (color (green|yellow|red)|) (dscp <0-63>|)
no dscp <0-63>
```

Parameters

`<0-63>`	DSCP value
`<0-7>`	Identifying queue number
`color`	(green|yellow|red)

Default

By default, 8 DSCP values are mapped to one queue.

For example: DSCP 0-7 queue 0, DSCP 8-15 queue 1.

Command Mode

config-ingress-dscp-map

Applicability

This command was introduced in OcNOS version 3.0.

Example

```
(config)#qos profile dscp-to-queue default
(config-ingress-dscp-map)#dscp 1 queue 2
```

ingress exp map

Use this command to map an exp value to the queue.

Use the `no` form of this command to remove the map.

Note: Color is an optional parameter. If not provided, color will be set to green.

Command Syntax

```
exp <0-7> queue <0-7> (color (green|yellow|red)|)
no exp <0-7>
```

Parameters

`<0-7>`	EXP value
`<0-7>`	Identifying queue number
`color`	(green|yellow|red)

Default

By default, EXP to queue mapping is one to one.

Command Mode

config-ingress-exp-queue-map

Applicability

This command was introduced in OcNOS version 3.0.

Example

```
(config)#qos profile exp-to-queue default
(config-ingress-exp-queue-map)#exp 1 queue 4 color green
(config-ingress-exp-queue-map)#exp 2 queue 3
```

low-delay-tolerance-profile

Use this command to configure a low delay tolerance profile on queues.

Use the `no` command to remove the applied profile.

Command Syntax

```
low-delay-tolerance-profile
no low-delay-tolerance-profile
```

Parameters

None

Default

By default, applied to queue 7 implicitly

Command Mode

Policy-class-map queuing mode

Policy map-class type queuing mode

Applicability

This command was introduced in OcNOS version 4.0

Examples

```
(config)#policy-map type queuing default default-out-policy
(config-pmap-que-def)#class type queuing default q0
(config-pmap-c-que-def)#low-delay-tolerance-profile
```

match access-group

Use this command to classify the group based on the access group.

Use the `no command` remove access group match criteria from a class map

Note: Match access-group is allowed only in “match-any” class type.

When match access-grp is configured, no more match criteria can be supported in the class-map.

When access-list is being used in class-map for match, only “permit rules” are allowed in the access-list.

Command Syntax

```
match access-group NAME
```

Parameters

`NAME`	Specify the access group name

Default

No default value is specified

Command Mode

Class-map mode

Applicability

This command was introduced in OcNOS version 3.0.

Examples

```
(config)#class-map match-any class_acl
(config-qos-match-any)#match access-group my_acl
```

© 2026 IP Infusion Inc. Proprietary

match cos

Use this command to classify the traffic based on CoS

Use the `no command` to remove the match configuration.

Note: The match commands which accept range has silent exit which makes removal of these match configuration easier. For example, classify the traffic based on CoS using the command `match cos 2,4,6` and remove the match configuration using the command `no match cos 2-6`.

Command Syntax

```
match cos <0-7>

no match cos
```

Note: Below command is applicable for class-map of type `match-any`.

```
match cos WORD

no match cos WORD
```

Note: Below command is applicable for class-map of type `match-all`.

```
match cos <0-7>

no match cos
```

Parameters

`cos WORD`	Enter cos value 0-7 or range of cos values separated by commas. Example: 2 or 2 and 4-5 or 3 and 4 and 5 or 2-4 and 5-7 etc.
`cos <0-7>`	Enter CoS value 0-7.

Default

No default value is specified

Command Mode

Class-map mode

Applicability

This command was introduced in OcNOS version 3.0.

Examples

```
(config)#class-map match-all ALL
(config-cmap-qos)#match cos 1
```

match cos inner

Use this command to classify the traffic based on inner CoS.

Use the `no` command to remove the match configuration.

Note: The match commands which accept range has silent exit which makes removal of these match configuration easier. For example, classify the traffic based on inner CoS using the command `match cos inner 2,4,6` and remove the match configuration using the command `no match cos inner 2-6`.

Command Syntax

```
match cos inner <0-7>
no match cos (inner|)
```

Note: Below `no` command is applicable for class-map of type `match-any`.

```
match cos inner WORD
no match cos inner WORD
```

Note: Below `no` command is applicable for class-map of type `match-all`.

```
match cos (<0-7>|inner <0-7>)
no match cos (inner|)
```

Parameters

`cos inner WORD`	Enter cos value 0-7 or range of cos values separated by commas. Example: 2 or 2 and 4-5 or 3 and 4 and 5 or 2-4 and 5-7 etc.
`<0-7>`	Enter CoS value 0-7.

Default

No default value is specified

Command Mode

Class-map mode

Applicability

This command was introduced in OcNOS version 3.0.

Examples

```
(config)#class-map type qos C_QOS1
(config-cmap-qos)#match cos inner 1
```

© 2026 IP Infusion Inc. Proprietary

match dscp

Use this command to classify the traffic based on DSCP.

Use the `no` command to remove the configured DSCP value.

Note:

`match dscp` cannot be configured in the class-map where match precedence is already configured.

Command Syntax

```
match dscp [WORD | af11 | af12 | af13 | af21 | af22 | af23 | af31| af32 | af33 |
  af41 | af42 | af43 | cs1 | cs2 | cs3 | cs4 | cs5 | cs6| cs7 | default | ef]
```

Note: Below `no` command is applicable for class-map of type match-any.

```
no match dscp ([WORD | af11 | af12 | af13 | af21 | af22 | af23 | af31 | af32 | af33
| af41 | af42 | af43 | cs1 | cs2 | cs3 | cs4 cs5 | cs6 | cs7 | default | ef])
```

Note: Below `no` command is applicable for class-map of type match-all.

```
no match dscp
```

Parameters

WORD	<0-63> List of DSCP values.
af11	AF11 DSCP (001010).
af12	AF12 DSCP (001100)
af13	AF13 DSCP (001110)
af21	AF21 DSCP (010010)
af22	AF22 DSCP (010100)
af23	AF23 DSCP (010110)
af31	AF31 DSCP (011010)
af32	AF32 DSCP (011100)
af33	AF33 DSCP (011110)
af41	AF41 DSCP (100010)
af42	AF42 DSCP (100100)
af43	AF43 DSCP (100110)
cs1	CS1(precedence 1) DSCP (001000)
cs2	CS2(precedence 2) DSCP (010000)
cs3	CS3(precedence 3) DSCP (011000)
cs4	CS4(precedence 4) DSCP (100000)
cs5	CS5(precedence 5) DSCP (101000)
cs6	CS6(precedence 6) DSCP (110000)
cs7	CS7(precedence 7) DSCP (111000)
default	Default DSCP (000000)
ef	EF DSCP (101110)

© 2026 IP Infusion Inc. Proprietary

`all`	Delete all matched DSCP values.

Default

No default value is specified

Command Mode

Class-map mode

Applicability

This command was introduced in OcNOS version 3.0.

Examples

```
(config)#class-map type qos C_QOS7
(config-cmap-qos)#match dscp 48
```

match ethertype

Use this command to classify the traffic based on the ethertype.

Use the `no` command to remove the match configuration.

Note: The match commands which accept range has silent exit which makes removal of these match configuration easier.

Qualifying the TPID values like 0x8100, 0x88A8, 0x9100 and 0x9200 etc as ethertype will not match if the traffic is flowing with the same TPId in the network.

Command Syntax

```
match ethertype WORD
```

Note: Below `no` command is applicable for class-map of type match-any.

```
no match ethertype(WORD|all)
```

Note: Below `no` command is applicable for class-map of type match-all.

```
no match ethertype
```

Parameters

`WORD`	Enter ethertype <0x600 to 0xffff)> or list of ethertype separated by commas. For example, 0x806,0x8035 etc.
`all`	Delete all ethertype entries.

Default

No default value is specified

Command Mode

Class-map mode

Applicability

This command was introduced in OcNOS version 3.0.

Examples

```
(config)#class-map type qos C_QOS1
(config-cmap-qos)#match ethertype 0x806
```

© 2026 IP Infusion Inc. Proprietary

match ip rtp

Use this command to configure a class map to use the Real-Time Protocol (RTP) port as a match criteria.

Use the `no command` to remove the RTP port as a match criteria.

Note: The match commands which accept range has silent exit which makes removal of these match configuration easier. For example, classify the traffic based on RTP port using the command `match ip rtp 5000,7000, 9000` and remove the match configuration using the command `no match ip rtp 5000-9000`.

Command Syntax

```
match ip rtp WORD
```

Note: Below `no` command is applicable for class-map of type match-any.

```
no match ip rtp (WORD|all)
```

Note: Below `no` command is applicable for class-map of type match-all.

```
no match ip rtp
```

Parameters

`WORD`	Specify User Datagram Protocol (UDP) or list of UDP ports that are using RTP. Valid values are from 2000 to 65535.
`all`	Delete all matched IP RTP values.

Default

No default value is specified

Command Mode

Class-map type qos

Applicability

This command was introduced in OcNOS version 3.0.

Examples

```
(config)# class-map my_test
(config-cmap-qos)# match ip rtp 2300
```

match ipv6 dscp

Use this command to classify the ipv6 traffic based on DSCP.

Use the no command to remove the configured DSCP value.

Note the following:

- The match commands which accept range have silent exit which makes removal of these match configuration easier. For example, classify the traffic based on DSCP using the command `match dscp 2,4,6` and remove the match configuration using the command `no match dscp 2-6`.
- The `match ipv6 dscp` command cannot be configured in the class-map where match ipv6 precedence is already configured.
- Any `match ipv6` commands cannot be configured in the class-map where `match ipv4` commands are already configured.

Command Syntax

```
match ipv6 dscp [WORD | af11 | af12 | af13 | af21 | af22 | af23 | af31| af32 | af33
  | af41 | af42 | af43 | cs1 | cs2 | cs3 | cs4 | cs5 | cs6| cs7 | default | ef]
```

Note: Below `no` command is applicable for class-map of type match-any.

```
no match ipv6 dscp ([WORD | af11 | af12 | af13 | af21 | af22 | af23 | af31 | af32 |
af33 | af41 | af42 | af43 | cs1 | cs2 | cs3 | cs4cs5 | cs6 | cs7 | default |
ef]|all)
```

Note: Below `no` command is applicable for class-map of type match-all.

```
no match ipv6 dscp
```

Parameters

WORD	<0-63> List of DSCP values.
af11	AF11 DSCP (001010).
af12	AF12 DSCP (001100)
af13	AF13 DSCP (001110)
af21	AF21 DSCP (010010)
af22	AF22 DSCP (010100)
af23	AF23 DSCP (010110)
af31	AF31 DSCP (011010)
af32	AF32 DSCP (011100)
af33	AF33 DSCP (011110)
af41	AF41 DSCP (100010)
af42	AF42 DSCP (100100)
af43	AF43 DSCP (100110)
cs1	CS1 (precedence 1) DSCP (001000)
cs2	CS2 (precedence 2) DSCP (010000)
cs3	CS3 precedence 3) DSCP (011000)

`cs4`	CS4 (precedence 4) DSCP (100000)
`cs5`	CS5 (precedence 5) DSCP (101000)
`cs6`	CS6 (precedence 6) DSCP (110000)
`cs7`	CS7 (precedence 7) DSCP (111000)
`default`	Default DSCP (000000)
`ef`	EF DSCP (101110)
`all`	Delete all matched DSCP values

Default

No default value is specified

Command Mode

Class-map mode

Applicability

This command was introduced in OcNOS version 3.0.

Examples

```
(config)#class-map type qos C_QOS7
(config-cmap-qos)#match ipv6 dscp 48
```

match ipv6 layer4

Use this command to classify the IPv6 traffic based on layer4 protocol src/dest port value.

Use the no command to remove the match configuration.

Please note the following:

- The match commands which accept range have silent exit which makes removal of these match configuration easier. Only one type of layer 4 matching criteria is supported per class-map.
- Any `match ipv6` commands cannot be configured in the class-map where `match ipv4` commands are already configured.

Command Syntax

```
match ipv6 layer4 (tcp|udp|any) (source-port|destination-port) WORD
no match ipv6 layer4 (((tcp|udp|any) (source-port|destination-port) WORD) |all)
```

Parameters

`tcp`	Specify TCP protocol
`udp`	Specify UDP protocol
`any`	Specify ANY protocol - TCP/UDP
`source-port`	Specify source TCP/UDP port
`destination-port`	Specify destination TCP/UDP port
`WORD`	Enter TCP/UDP port value <1-65535> or range of values separated by commas such as 1 or 1,4-5 or 50,51,52
`all`	Delete all layer4 port entries

Default

No default value is specified

Command Mode

Class-map mode

Applicability

This command was introduced in OcNOS version 3.0.

Examples

```
(config)#class-map type qos C_QOS1
(config-cmap-qos)#match ipv6 layer4 tcp source-port 1
```

match ipv6 precedence

Use this command to IPv6 traffic classification based on precedence.

Use the no command to remove the match configuration.

Please note the following:

- The match commands which accept range has silent exit which makes removal of these match configuration easier. For example, classify the traffic based on precedence using the command match ipv6 precedence 2,4,6 and remove the match configuration using the command no match ipv6 precedence 2-6.
- The `match ipv6 precedence` command cannot be configured in the class-map where match ipv6 dscp is already configured.
- Any `match ipv6` commands cannot be configured in the class-map where `match ipv4` commands are already configured.

Command Syntax

```
match ipv6 precedence [WORD | critical | flash | flash-override | immediate |
  internet | network | priority | routine]
```

Note: Below `no` command is applicable for class-map of type match-any.

```
no match ipv6 precedence ([WORD | critical | flash | flash-override| immediate |
internet | network | priority | routine]|all)
```

Note: Below `no` command is applicable for class-map of type match-all.

```
no match ipv6 precedence
```

Parameters

`word`	IP precedence value
`critical`	Critical precedence
`flash`	Flash precedence
`flash-override`	Flash override precedence
`immediate`	Immediate precedence
`internet`	Internetwork control precedence
`network`	Network control precedence
`priority`	Priority precedence
`routine`	Routine precedence
`all`	Delete all matched IP precedence values

Default

No default value is specified

Command Mode

Class-map mode

Applicability

This command was introduced in OcNOS version 3.0.

Examples

```
(config)# class-map my_test
(config-cmap-qos)#match ipv6 precedence critical
```

match layer4

Use this command to classify the traffic based on layer4 protocol src/dest port value.

Use the `no` command to remove the match configuration.

Note: The match commands which accept range has silent exit which makes removal of these match configuration easier. Only one type of layer4 matching criteria is supported per class-map.

Command Syntax

```
match layer4 (tcp|udp|any) (source-port|destination-port) <1-65535>
no match layer4 ((tcp|udp|any) (source-port|destination-port) <1-65535>)
```

Parameters

`tcp`	Specify TCP protocol
`udp`	Specify UDP protocol
`any`	Specify ANY protocol – TCP/UDP
`source-port`	Specify source TCP/UDP port
`destination-port`	Specify destination TCP/UDP port
`<1-65535>`	Enter TCP/UDP port value <1-65535> or range of values separated by commas. e.g. 1 or 1,4-5 or 50,51,52

Default

No default value is specified

Command Mode

Class map Type QoS mode

Applicability

This command was introduced in OcNOS version 3.0.

Examples

```
(config)#class-map type qos C_QOS1
(config-cmap-qos)#match layer4 tcp source-port 1
```

match mpls

Use this command to classify the traffic based on the top mpls exp value.

Use the `no` command to remove the match configuration.

Note: The match commands which accept range has silent exit which makes removal of these match configuration easier.

Command Syntax

```
match mpls experimental topmost <0-7>
```

```
no match mpls experimental topmost
```

Note: Below `no` command is applicable for class-map of type match-any.

```
no match mpls experimental topmost (<0-7> |all)
```

Parameters

`<0-7>`	Enter EXP value <0-7> or range of exp values seperated by commas. e.g. 2 or 2,4-5 or 3,4,5 or 2-4,5-7 etc.
`all`	Delete all experimental values.

Default

No default value is specified

Command Mode

Class map Type QoS mode

Applicability

This command was introduced in OcNOS version 3.0.

Examples

```
(config)#class-map type qos C_QOS1
(config-cmap-qos)#match mpls experimental topmost 3
(config-cmap-qos)#exit

(config)#class-map match-any class_match_any
(config-qos-match-any)#no match mpls experimental topmost all
(config-qos-match-any)#exit

(config)#class-map match-all class_match_all
(config-cmap-qos)#no match mpls experimental topmost
(config-cmap-qos)#exit
```

match precedence

Use this command to traffic classification based on precedence.

Use the `no` command to remove the match configuration.

Note: The match commands which accept range has silent exit which makes removal of these match configuration easier. For example, classify the traffic based on precedence using the command `match precedence 2,4,6` and remove the match configuration using the command `no match precedence 2-6`.

Match precedence cannot be configured in the “match-all” class-map where `match dscp` is already configured.

Command Syntax

```
match precedence [WORD | critical | flash | flash-override | immediate | internet |
  network | priority | routine]
```

```
no match precedence
```

Note: Below `no` command is applicable for class-map of type match-any.

```
no match precedence ([WORD | critical | flash | flash-override| immediate |
internet | network | priority | routine]|all)
```

Parameters

`word`	IP precedence value (0-7)
`critical`	Critical precedence
`flash`	Flash precedence
`flash-override`	Flash override precedence
`immediate`	Immediate precedence
`internet`	Internetwork control precedence
`network`	Network control precedence
`priority`	Priority precedence
`routine`	Routine precedence
`all`	Delete all matched IP precedence values.

Default

No default value is specified

Command Mode

Class map Type QoS mode

Applicability

This command was introduced in OcNOS version 3.0.

Examples

```
(config)#class-map my_test
(config-cmap-qos)#match precedence critical
(config-cmap-qos)#exit
```

© 2026 IP Infusion Inc. Proprietary

```
(config)#class-map match-any class_any
(config-qos-match-any)#no match precedence all
(config-qos-match-any)#exit
```

match traffic-type

Use this command to classify the traffic based on the traffic type.

Use the `no` command to remove the match configuration.

Command Syntax

```
match traffic-type (l2-unknown|l2-mc|l2-bc|default)
match traffic-type
```

Note: Below `no` command is applicable for class-map of type match-any.

```
no match traffic-type (l2-unknown|l2-mc|l2-bc|default)
```

Parameters

`l2-unknown`	Unknown L2 traffic
`l2-mc`	L2 multicast
`l2-bc`	L2 broadcast
`default`	L2 traffic-type all

Default

No default value is specified

Command Mode

Class-map Type QoS mode

Applicability

This command was introduced in OcNOS version 4.1.

Examples

```
(config)#class-map type qos C_QOS1
(config-cmap-qos)#match traffic-type l2-unknown
(config-cmap-qos)#exit

(config)#class-map match-any C_MATCH_ANY
(config-qos-match-any)#no match traffic-type default
(config-qos-match-any)#exit
```

© 2026 IP Infusion Inc. Proprietary

match vlan

Use this command to classify the traffic based on a VLAN.

Use the `no` command to remove the match configuration.

Note: The match commands which accept range has silent exit which makes removal of these match configuration easier. For example, classify the traffic based on VLAN using the command `match vlan 2,4,6` and remove the match configuration using the command `no match vlan 2-6`.

Command Syntax

```
match vlan WORD
no match vlan
```

Note: Below `no` command is applicable for class-map of type match-any.

```
no match vlan (WORD|all)
```

Parameters

`WORD`	Enter VLAN ID <1-4094> or range of VLAN ID's separated by commas. For example, 2 or 2,4-5 or 50,51,52 or 100-120,122-130 etc.
`all`	Delete all VLAN ID entries.

Default

No default value is specified

Command Mode

Class-map Type QoS mode

Applicability

This command was introduced in OcNOS version 3.0.

Examples

```
(config)#class-map type qos C_QOS1
(config-cmap-qos)#match vlan 100
(config-cmap-qos)#exit

(config)#class-map match-any C_ANY
(config-qos-match-any)#no match vlan all
(config-qos-match-any)#exit
```

© 2026 IP Infusion Inc. Proprietary

match vlan inner

Use this command to classify the traffic based on the inner VLAN.

Use the no command to remove the match configuration.

Note: The match commands which accept range has silent exit which makes removal of these match configuration easier. For example, classify the traffic based on the inner VLAN using the command `match vlan inner 2,4,6` and remove the match configuration using the command `no match vlan inner 2-6`.

Command Syntax

```
match vlan inner WORD
no match vlan inner
```

Note: Below no command is applicable for class-map of type match-any.

```
no match vlan inner (WORD|all)
```

Parameters

WORD	Enter VLAN ID <1-4094> or list of VLAN ID's separated by commas. For example, 2,4 etc.
all	Delete all VLAN ID entries.

Default

No default value is specified

Command Mode

Class-map Type QoS mode

Applicability

This command was introduced in OcNOS version 3.0.

Examples

```
(config)#class-map type qos C_QOS1
(config-cmap-qos)#match vlan inner 1
(config-cmap-qos)#exit

(config)#class-map match-any C_ANY
(config-qos-match-any)#no match vlan inner all
(config-qos-match-any)#exit
```

police

Use this command to configure policing of the data rates for a particular class of traffic.

Use the `no` command to remove a policing configuration.

Note: Committed Information Rate (CIR) and Excess Information Rate (EIR) can only be whole numbers.

The default burst is 5 ms traffic of the configured rate. Burst configured in milliseconds and microseconds are converted in kbytes by calculating it with input rates.

Police rate (CIR/EIR) value in percentage is only applicable on physical interfaces.

Command Syntax

For Qumran MX:

```
police (colour-blind | colour-aware|) (cir) (<1-720000000> (kbps|mbps|gbps) |
  percent <1-100>)((eir (<1-720000000> (kbps|mbps|gbps) | percent <1-100>)|)((bc)
  <1-4161> (kbytes|mbytes|ms|us)|)((be) <1-4161> (kbytes|mbytes|ms|us)|))
```

```
no police
```

For Qumran AX:

```
police (colour-blind | colour-aware|) (cir) (<1-500000000> (kbps|mbps|gbps) |
  percent <1-100>)((eir (<1-500000000> (kbps|mbps|gbps) | percent <1-100>)|)((bc)
  <1-4161> (kbytes|mbytes|ms|us)|)((be) <1-4161> (kbytes|mbytes|ms|us)|))
```

```
no police
```

Parameters

`colour-blind`	Do not police on color.
`colour-aware`	Do police on color.
`cir`	Committed information rate.
`<1-720000000>`	Excess information rate values 22kbps-720gbps.
`<1-500000000>`	Excess information rate values 22kbps-720gbps.
`percent <1-100>`	Specify percentage of link rate.
`eir`	Excess information rate.
`kbps`	Specify the units of kilobits per second.
`mbps`	Specify the units of megabits per second.
`gbps`	Specify the units of gigabits per second.
`bc <1-4161>`	Burst rate committed.
`be <1-4161>`	Burst rate extended.
`kbytes`	Specify the units of bc/be in kilobytes per second.
`mbytes`	Specify the units of bc/be in megabytes per second.
`ms`	Specify the units of bc/be in milliseconds.
`us`	Specify the units of bc/be in microseconds.

Default

By default, policer type is colour-blind

Command Mode

Policy map class Type QoS mode and Policy map class Type copp mode

Applicability

This command was introduced in OcNOS version 3.0. Added parameter `copp in` OcNOS version 6.6.0.

Examples

```
(config)#policy-map type qos 2345
(config-pmap-qos)#class type qos 2345
(config-pmap-c-qos)# police cir 2 mbps eir 4 mbps bc 2 mbytes be 4 mbytes

(config)#config terminal
(config)#policy-map type copp copp1

#configure terminal
Enter configuration commands, one per line.  End with CNTL/Z.
(config-pmap-copp)#class type copp 2
(config-pmap-c-copp)#police colour-blind cir 2 mbps
```

policy-map

Use this command to create a policy map and enter policy-map mode.

Use the `no` form of the command to remove a policy map.

Note: You cannot delete a policy map if it is attached to an interface.

Command Syntax

```
policy-map {NAME | (type (queuing|queuing default) NAME)}
policy-map {NAME | (type qos NAME)
policy-map {NAME | (type copp NAME)}
no policy-map {NAME | (type (queuing|queuing default) NAME)}
no policy-map {NAME | (type qos NAME)}
no policy-map {NAME | (type qos NAME)}
```

Parameters

`copp`	Specify CoPP policy-map
`NAME`	Specify a policy-map name (maximum 32 characters)
`type`	Specify the policy-map type
`qos`	QoS policy map (ingress/egress policy-map)
`queuing`	Queuing policy map (egress policy-map)
`default`	Default queues of the port
`qos NAME`	Specify a policy-map name

Default

No default value is specified

Command Mode

Configuration mode

Applicability

This command was introduced in OcNOS version 3.0. Added parameter `copp and copp NAME in` OcNOS version 6.6.0

Examples

```
(config)#policy-map type qos PQOS

(config)#config terminal
(config)#policy-map type copp copp1
```

© 2026 IP Infusion Inc. Proprietary

priority level <0-7>

Use this command to configure a single output queuing class as the priority queue.

Use `no` command to unset the priority level from the queue.

Strict priority mode supports 8 level, i.e., 0 to 7. The queue is constructed such that a higher priority level has a higher priority.

If more than one queue is in the same level, then there will be fair queuing between those queues.

On qos-enable, all the queues will be in strict-priority. After changing the mode or un-setting the priority, queues will be set for WFQ mode with the default weight 1. To set them again in strict priority, the user needs to configure them manually.

Command Syntax

```
priority level <0-7>
no priority level
```

Parameters

`level <0-7>`	Priority level values. 0 to 7 for ports and 0 to 3 for services.

Default

No default value is specified

Command Mode

Policy-map class Type Queuing mode

Applicability

This command was introduced in OcNOS version 3.0.

Examples

```
#configure terminal
(config)#policy-map type queuing default DEFAULT_OUT_POLICY
(config-pmap-que-def)#class type queuing default q0
(config-pmap-c-que-def)#priority level 2
(config-pmap-c-que-def)#exit
(config-pmap-que-def)#exit
(config)#
```

priority (QoS)

Use this command to set the user priority for the class attached to this policy-map

Use the `no` parameter with this command to unset the priority value

Note: The higher the priority number in a policy map, the greater the priority assigned to the class. User-configured priorities override default priorities. This rule does not apply to classes with match access-group criteria, as priorities in such cases are determined by the sequence number in the access list. This behavior is consistent across all platforms.

Command Syntax

```
priority <1-1000>
no priority
```

Parameters

`<1-1000>`	Priority value

Default

No default value is specified

Command Mode

Policy map class Type QoS mode

Applicability

This command was introduced in OcNOS version 3.0.

Examples

```
#configure terminal
(config)#policy-map PMAP1
(config-pmap-qos)#class CMAP1
(config-pmap-c-qos)#priority 20
(config-pmap-c-qos)#exit
(config-pmap-qos)#exit
(config)#
```

qos (enable | disable)

Use this command to globally enable or disable Quality-of-Service (QoS).

Note: Enabling or disabling QoS is a disruptive operation, stopping all traffic on ports which causes traffic loss.

Command Syntax

```
qos (enable | disable)
```

Parameters

None

Default

By default, QoS is enabled.

Command Mode

Configure mode

Applicability

This command was introduced in OcNOS version 3.0.

Examples

```
#configure terminal
(config)#show running-config qos
qos enable
!
!
!

(config)#qos disable
(config)#commit

(config)#show running-config qos
% QoS is not enabled globally

(config)#qos enable
(config)#commit
(config)#show running-config qos
qos enable
!
!
!
```

qos map-profile (Qumran)

Use this command to attach (map) a profile to an interface.

Use the `no` form of this command to remove a profile.

Note: By-default, “default” profiles are attached on their supported interfaces (cos-to-queue and queue-to-cos on L2 interfaces and DSCP-to-queue and DSCP-to-DSCP on L3 interfaces).

You can create and attach your own profile to supported interfaces. After removing a user-defined profiles from an interface, the “default” profile is applied.

Scalability:

- Total user configurable queue-to-cos map: 13
- Total user configurable queue-to-dscp map: 14
- Total user configurable dscp-to-queue map: 13

Note: As EXP and DSCP maps share same bank, the number of user configurable dscp maps will vary as per number of exp maps created.

Command Syntax

```
qos map-profile (cos-to-queue | dscp-to-queue | queue-color-to-cos | dscp-to-dscp |
  exp-encap | precedence-to-queue | precedence-to-precedence | dscp-to-encap |
  dscp-to-encap) NAME

no qos map-profile (cos-to-queue | dscp-to-queue | queue-color-to-cos | dscp-todscp
  | exp-encap | precedence-to-queue | precedence-to-precedence | dscp-to-encap |
  dscp-to-encap) NAME
```

Parameters

`cos-to-queue`	CoS to Traffic-Class map profile
`dscp-encap`	profile for dscp encapsulation map
`dscp-to-dscp`	DSCP to DSCP map profile
`dscp-to-queue`	DSCP to Traffic-Class map profile
`precedence-to-precedence`	Precedence to Precedence map profile
`precedence-to-queue`	Precedence to Traffic-Class map profile
`queue-color-to-cos`	Traffic-Class color to CoS map profile
`queue-to-exp`	Traffic-Class to exp map profile
`NAME`	Profile map name (maximum 32 characters)

Default

By default, the default `cos-to-queue-profile` is applied to an L2 interface, and the default `DSCP-to-queue profile` is attached to an L3 interface.

Command Mode

Interface modes

Applicability

This command was introduced in OcNOS version 3.0 and this command is applicable for Qumran.

© 2026 IP Infusion Inc. Proprietary

Example

```
(config-if)#qos map-profile cos-to-queue cos-map
(config-if)#qos map-profile dscp-to-queue dscp-map
(config-if)#qos map-profile queue-color-to-cos egress-cos-map
(config-if)#qos map-profile dscp-to-dscp dscp-dscp-map
(config-if)#qos map-profile exp-encap exp-encap-map
```

qos map-profile (Qumran2)

Use this command to attach (map) a profile to an interface.

Use the `no` form of this command to remove a profile.

Note: By-default, “default” profiles are attached on their supported interfaces (cos-to-queue and queue-to-cos on L2 interfaces and DSCP-to-queue and DSCP-to-DSCP on L3 interfaces).

You can create and attach your own profile to supported interfaces. After removing a user-defined profiles from an interface, the “default” profile is applied.

Command Syntax

```
qos map-profile (cos-to-queue | dscp-to-queue | queue-color-to-cos | dscp-to-dscp |
  queue-to-exp | precedence-to-queue | precedence-to-precedence | dscp-to-encap)
  NAME
```

```
no qos map-profile (cos-to-queue | dscp-to-queue | queue-color-to-cos | dscp-todscp
  | queue-to-exp | precedence-to-queue | precedence-to-precedence | dscp-to-encap)
  NAME
```

Parameters

`os-to-queue`	CoS to Traffic-Class map profile
`dscp-encap`	profile for dscp encapsulation map
`dscp-to-dscp`	DSCP to DSCP map profile
`dscp-to-queue`	DSCP to Traffic-Class map profile
`precedence-to-precedence`	Precedence to Precedence map profile
`precedence-to-queue`	Precedence to Traffic-Class map profile
`queue-color-to-cos`	Traffic-Class color to CoS map profile
`queue-to-exp`	Traffic-Class to exp map profile
`NAME`	Profile map name (maximum 32 characters)

Default

By default, the default `cos-to-queue-profile` is applied to an L2 interface, and the default `DSCP-to-queue profile` is attached to an L3 interface.

Command Mode

Interface modes

Applicability

This command was introduced in OcNOS version 5.1 and this command is applicable for Qumran2.

Example

```
(config-if)#qos map-profile cos-to-queue cos-map
(config-if)#qos map-profile dscp-to-queue dscp-map
(config-if)#qos map-profile queue-color-to-cos egress-cos-map
(config-if)#qos map-profile dscp-to-dscp dscp-dscp-map
(config-if)#qos map-profile queue-to-exp queue-to-exp-map
```

qos profile

Use this command to create new profiles or to update “default” profiles.

Use the `no` form of this command to remove the profiles.

Note: “exp-encap” profile will be ineffective on tunnels with only 2 nodes with PHP configuration.

Command Syntax

```
qos profile (cos-to-queue | dscp-to-queue | queue-color-to-cos | dscp-to-dscp |
  dscp-encap | exp-to-queue | precedence-to-precedence  |  precedence-to-queue  |
  queue-color-to-dscp  | queue-to-exp |  queue-to-precedence ) (NAME|default)
```

```
no qos profile (cos-to-queue | dscp-to-queue | queue-color-to-cos | dscp-to-dscp |
  dscp-encap | exp-to-queue | precedence-to-precedence  |  precedence-to-queue  |
  queue-color-to-dscp  | queue-to-exp |  queue-to-precedence ) (cos-to-queue |
  dscp-to-queue | queue-color-to-cos | dscp-to-dscp | dscp-encap | exp-to-queue)
  NAME
```

Parameters

`cos-to-queue`	CoS to Traffic-Class profile
`dscp-encap`	Egress profile to map dscp(L3) | Traffic-Class+colour(L2) to dscpEncap
`dscp-to-dscp`	DSCP to DSCP profile
`dscp-to-queue`	DSCP to Traffic-Class profile
`exp-to-queue`	EXP to Traffic-Class ingress profile
`precedence-to-precedence`	Precedence to Precedence profile
`precedence-to-queue`	Precedence to Traffic-Class profile
`queue-color-to-cos`	Traffic-Class color to CoS profile
`queue-color-to-dscp`	Traffic-Class color to DSCP profile
`queue-to-exp`	Traffic-Class (remark-dscp/8 in case of L3 and queue in case of L2) to exp egress profile
`queue-to-precedence`	Traffic-Class to Precedence profile
`NAME`	Profile map name (maximum 32 characters)

Default

By default, “default” profile is created for all the profile types. These profiles can only be updated by the user neither be created nor be destroyed.

Command Mode

Configure modes

Applicability

This command was introduced in OcNOS version 3.0.

Example

To create a new profile:

```
(config)#qos profile cos-to-queue cos-map
(config)#qos profile dscp-to-queue dscp-map
(config)#qos profile queue-color-to-cos egress-cos-map
(config)#qos profile dscp-to-dscp dscp-dscp-map
(config)#qos profile exp-encap exp-encap-map
```

qos profile exp-encap (Qumran)

Use this command to create new profiles or to update "default" exp encapsulation profiles. These profiles are applicable on the nodes where MPLS label is inserted.

Use the `no` form of this command to remove the exp encapsulation profiles.

Note: `exp-encap` profile will be ineffective on tunnels with only 2 nodes with PHP configuration.

Command Syntax

```
qos profile exp-encap (NAME|default)
no qos profile exp-encap NAME
```

Parameters

`exp-encap`	Profile for the exp encapsulation map
`NAME`	Profile map name (maximum 32 characters)

Default

By default, “default” profile is created for all the profile types. These profiles can only be updated by the user neither be created nor be destroyed.

Command Mode

Configure modes

Applicability

This command was introduced in OcNOS version 3.0 and this command is applicable for Qumran.

Example

To create a new profile:

```
(config)#qos profile exp-encap exp-encap-map
```

qos profile precedence-to-precedence

Use this command to set the precedence to precedence profile.

Use the no form of this command delete the profile.

Use show qos-profile to show the profile detail.

Use qos map-profile (Qumran) to map the profile on interface.

Command Syntax

```
qos profile precedence-to-precedence (NAME|)precedence <0-7> precedence <0-7>
```

Parameters

`NAME`	The profile name to be created
`precedence <0-7>`	In precedence value
`precedence <0-7>`	Out precedence value

Command Mode

Configure modes

Applicability

This command was introduced in OcNOS version 5.0.

Example

```
#configure terminal
(config)#qos profile precedence-to-precedence prec-prec-map
(config-egress-prec-map)#precedence 1 precedence 0
(config)#interface xe3
(config-if)#qos map-profile precedence-to-precedence prec-prec-map
```

qos profile precedence-to-queue

Use this command to set the precedence to queue profile.

Use the no form of this command delete the profile.

Use show qos-profile to show the profile detail.

Use qos map-profile (Qumran) to map the profile on interface.

Command Syntax

```
qos profile precedence-to-queue (NAME|)
precedence <0-7> queue <0-7>
no qos profile precedence-to-queue (NAME|)
no precedence <0-7> queue <0-7>
```

Parameters

`NAME`	The profile name to be created
`precedence <0-7>`	Precedence value
`queue <0-7>`	Traffic-Class values

Command Mode

Configure modes

Applicability

This command was introduced in OcNOS version 5.0.

Example

```
#configure terminal
(config)#qos profile precedence-to-queue prec-map
(config-ingress-prec-map)#precedence 1 queue 1
(config)#interface xe1
(config-if)#qos map-profile precedence-to-queue prec-map
```

qos profile queue-to-exp (Qumran2)

Use this command to create new profiles or to update "default" queue to exp profiles.

Use the `no` form of this command to remove the profiles.

Note: `queue-to-exp` profile will be ineffective on tunnels with only 2 nodes with PHP configuration.

Note: Non-default profile is applicable only on the nodes where MPLS label is encapsulated. On swap and php nodes default profile is applicable when lsp-model is uniform.

Command Syntax

```
qos profile queue-to-exp (NAME|default)
no qos profile queue-to-exp NAME
```

Parameters

`queue-to-exp`	Profile for Traffic-Class to exp map
`NAME`	Profile map name (maximum 32 characters)

Default

By default, “default” profile is created for all the profile types. These profiles can only be updated by the user neither be created nor be destroyed.

Command Mode

Configure modes

Applicability

This command was introduced in OcNOS version 5.1 and this command is applicable for Qumran2.

Example

To create a new profile:

```
To create a new profile:
(config)#qos profile queue-to-exp que-exp-map
```

qos red-drop-disable

Use this command to disable red packet drop in the system. Red packets are dropped in the system by default to achieve ingress rate limiting via policer and storm control. However, this command is used when there is no use-case for rate limiting traffic and red packets need to be allowed in the system

Use the `no` form of this command to enable red packet drop.

Command Syntax

```
qos red-drop-disbale
no qos red-drop-disable
```

Parameters

None

Defaults

By default, red packet drop is enabled.

Command Mode

Configure

Applicability

This command was introduced in OcNOS version 3.0.

Examples

```
#configure terminal
(config)# qos red-drop-disable
(config)#no qos red-drop-disable
```

qos remark

Use this command to enable remarking of the Class of service (CoS) and Differentiated Services Control Protocol (DSCP) set by the egress map.

Use the `no` command to disable remarking of the CoS and DSCP.

Command Syntax

In Config mode :

```
qos remark (cos|dscp)
no qos remark (cos|dscp)
```

In interface mode:

```
qos remark (cos|dscp)(enable|disable)
no qos remark (cos|dscp)
```

Parameters

`type`	Remarking type, e.g. CoS or DSCP
`(enable|disable)`	Remarking action

Default

By default, remarking is disabled.

Command Mode

Configure Mode

Interface Mode

Applicability

This command was introduced in OcNOS version 3.0.

Examples

```
(config)#qos remark cos
(config)#qos remark dscp
(config-if)# qos remark cos disable
(config-if)# qos remark cos enable
(config-if)# qos remark dscp disable
```

Interface remarking will take priority over global remarking configurations.

qos statistics

Use this command to enable Quality of Service (QoS) statistics.

Use the `no` command to disable QoS statistics

Note: Class-map statistics is cleared whenever the match or action property of the class is modified dynamically.

Command Syntax

```
qos statistics
no qos statistics
```

Parameters

None

Default

By default, QoS statistics is disabled

Command Mode

Configure Mode

Applicability

This command was introduced in OcNOS version 3.0.

Examples

```
(config)#qos statistics
```

© 2026 IP Infusion Inc. Proprietary

qos untagged-priority

Use this command to set internal priority for untagged traffic on L2 ports.

Use the `no` form of the command to remove the configuration.

Note: This command is not supported on Qumran 2 series platforms.

Command Syntax

```
qos untagged-priority <0-7>
no qos untagged-priority
```

Parameters

`qos` Quality of Service

`untagged-priority` Internal priority for untagged traffic

`<0-7>` Value

Default

No default value is specified

Command Mode

Interface mode

Applicability

This command was introduced in OcNOS version 3.0.

Examples

```
(config)#interface xe1
(config-if)#qos untagged-priority 3
```

queue exp (Qumran2)

Use this command to map a queue color to the some exp value for the traffic entering or within mpls network.

Use the `no` form of this command to remove the map.

Note: Color is an optional parameter. If not provided, the same map is set for all colors.

Command Syntax

```
queue <0-7> (color (green|yellow|red|all)|) exp <0-7>
no queue <0-7> (color (green|yellow|red|all|)|)
```

Parameters

`<0-7>`	Traffic-Class values
`color`	Green, yellow, red, or all
`<0-7>`	MPLS experimental values

Default

By default, queue values are one-to-one mapped to Exp. For example, queue 0 exp 0, queue 1 exp 1, and so on.

Command Mode

config-egress-queue-exp-map

Applicability

This command was introduced in OcNOS version 5.1 and this command is applicable for Qumran2.

Examples

```
(config)#qos profile exp-to-queue default
(config-egress-queue-exp-map)#queue 2 exp 1
(config-egress-queue-exp-map)#queue 4 color yellow exp 3
```

Color is an optional parameter, so if users do not provide color for all colors, the same EXP is set. If users provide color, then only that specific color egress map is changed. If, however, users provide a map for all colors, as well as maps without color, the map with color will take priority.

For example,

```
(config-egress-queue-exp-map)#no queue 1 color green
(config-egress-queue-exp-map)#no queue 1
```

If a user wants to remove all the mapping of queue (for all colors), then there is no need to provide color, else the user can provide a specific color to remove a specific map.

queue-limit

Use this command to configure tail drop by setting queue limits on egress queues.

Use the `no command` to remove a queue limit.

Command Syntax

```
queue-limit (<1-629145600>) (packets|bytes|kbytes|mbytes|ms)
no queue-limit
```

Parameters

`<1-629145600>`	Specify queue-limit threshold range values in packets, bytes, mbytes,ms us or Kilobytes.
`packets`	Specify the units of queue-limit in packets (min 9 - max 614400)
`bytes`	Specify the units of queue-limit in bytes (min 9416 - max 629145600)
`kbytes`	Specify the units of queue-limit in kilobytes (min 9 - max 614400)
`mbytes`	Specify the units of queue-limit in mega-bytes (min 1 - max 600)
`ms`	Specify the units of queue-limit in milliseconds (min 1 - max 50)
`us`	Specify the units of queue-limit in micro-seconds (min 1 - max 50000)

Default

Default queue size is 62914560 bytes

Command Mode

Policy-class-map-queue mode

Applicability

This command was introduced in OcNOS version 3.0.

Examples

```
(config)# policy-map type queuing default P1
(config-pmap-que-def)#class type queuing default q1
(config-pmap-c-que-def)# queue-limit 1 mbytes
```

random-detect

Use this command to configure weighted random early detection (WRED).

Use the `no` command to remove a WRED configuration.

Command Syntax

```
random-detect (green|yellow|red|all|) (min-threshold|)<1-629145600>
  (bytes|kbytes|mbytes|ms|packets)(max-threshold|) <1-629145600>
  (bytes|kbytes|mbytes|ms|packets) (drop-probability <1-100>|)

random-detect (weight <1-31>|))

no random-detect (green|yellow|red|all|)

no random-detect weight
```

Parameters

`min-threshold`	
	Specify the minimum threshold. In the range of <1-629145600>
`max-threshold`	
	Specify the maximum threshold. In the range of <1-629145600>
`packets`	Specify the units of queue-limit in packets – range (min 1, max 614400)
`bytes`	Specify the units of threshold in bytes – range (min 1024, max 629145600)
`kbytes`	Specify the units of threshold in kilobytes – range (min 1, max 614400)
`mbytes`	Specify the units of threshold in mega-bytes – range (min 1, max 600)
`ms`	Specify the units of threshold in milliseconds – range (min 1, max 50)
	Threshold value in ms will be calculated from interface-speed and will be converted into bytes
`drop-probability`	
	Drop-probability is the fraction of packets dropped when the average queue depth is at the maximum threshold.It can be configured per color Specify the drop-probability in the range <1-100>
`weight`	Configures the weight factor used in calculating the average queue length. Specify the weight in the range <1-31>. Value specified will be set for all colors

Default

No default value is specified

Command Mode

Policy-class map queue mode

Applicability

This command was introduced in OcNOS version 3.0.

Examples

```
(config)# policy-map type queuing default P1
```

© 2026 IP Infusion Inc. Proprietary

```
(config-pmap-que-def)#class type queuing default q1
(config-pmap-c-que-def)# random-detect min-threshold 1000 mbytes max-threshold
2000 mbytes
```

shape

Use this command to configure shaping on an egress queue to impose a maximum rate on it.

Use the `no command` to remove a shaping configuration.

Note: The configurable queuing shape rate ranges from a minimum of 469 Kbps to a maximum of 483 Gbps.

- The granularity for the low-rate range (Q1) remains 469 Kbps, and for the higher range, it is 1.56%.
- Additionally, the granularity for Q2 is 2.6 Mbps to 4Tb/s, with a corresponding credit size of 4K.

Command Syntax

```
shape (<1-483000000>(kbps|mbps|gbps) | percent <1-100>)
no shape
```

Parameters

`<1-483000000>`	Shaping is based on an average rate. Average rate for shaping in the range of `<1-483000000>.Min shape rate configured is 469 kbps and max shape rate configured is 483 gbps.`
`kbps`	Specify the units of kbps per second
`mbps`	Specify the units of mbps per second
`gbps`	Specify the units of gbps per second
`percent`	Specify the percentage from 1 to 100

Default

No default value is specified

Command Mode

Policy-class-map queue mode

Applicability

This command was introduced in OcNOS version 3.0.

Examples

```
(config)# policy-map type queuing default default-out-policy
(config-pmap-que-def)# class type queuing default q0
(config-pmap-c-que-def)# shape percent 25
(config-pmap-c-que)#
```

shape rate

Use this command to configure shaping on an egress port to impose a maximum rate on it.

Use the `no` form of the command to remove a shaping configuration.

Note: Minimum configurable port shape rate is 52 Kbps with Granularity of 52 Kbps and maximum shape rate is 1000 Gbps, but applicable maximum rate may be speed of interface.

Command Syntax

```
shape rate <1-1000000000> (kbps|mbps|gbps)
no shape rate
```

Parameters

`<1-1000000000>`	Specify rate for shaping in the range of `<1-1000000000>. Min shape rate configured on port is 52kbps and max shape rate configured on port is 1000gbps`
`kbps`	Specify the units of kbps per second.
`mbps`	Specify the units of mbps per second.
`gbps`	Specify the units of gbps per second.

Default

No default value is specified

Command Mode

Interface mode

Applicability

This command was introduced in OcNOS version 3.0.

Examples

```
(config)# interface xe11
(config-if)# shape rate 100 mbps
```

service-policy type qos

Use this command to attach a service-policy of type qos to the interface.

Use the `no` command to remove a service-policy from an interface.

Command Syntax

```
service-policy type qos input NAME
no service-policy type qos input NAME
```

Parameters

`type`	Specify whether the policy map is of type qos.
`NAME`	Specify the policy map to attach to this interface.

Default

No default value is specified

Command Mode

Interface mode

Applicability

This command was introduced in OcNOS version 3.0.

Examples

```
(config)#interface xe3
(config-if)#service-policy type qos input PQOS
```

service-policy type queuing

Use this command to attach a service-policy of type queuing to the interface.

Use the `no` command to remove a service-policy from an interface.

Command Syntax

```
service-policy type queuing output NAME
no service-policy type queuing output NAME
```

Parameters

`type`	Specify whether the policy map is of type queuing.
`NAME`	Specify the policy map to attach to this interface.

Default

By default, default-out-policy is attached on all interface

Command Mode

Interface mode

Applicability

This command was introduced in OcNOS version 3.0.

Examples

```
(config)#interface xe3
(config-if)#service-policy type queuing output PQOS
```

service-queue

Use this command to configure service-queues for sub-interfaces or use port interface queues.

Use the `no` parameter of this command to remove a queue limit.

Command Syntax

```
service-queue physical | flow-id
no service-queue
```

Parameters

`flow-id`	Configures queue per interface.
`physical`	Configures physical port queues.

Command Mode

Interface mode

Applicability

This command was introduced in OcNOS version 6.4.1 and this command is applicable for Qumran1 and Qumran2 series platforms.

Examples

Note: When user create a sub-interface (eg., xe4.1), the sub-interface will use port queues for traffic forwarding and will not have its own queues.

```
OcNOS(config)#interface xe4
OcNOS(config-if)#service-queue physical
```

set cos

Use this command for matching traffic classes set action as change CoS in the egress packet to the prescribed value.

Use the `no command` to remove the assigned value from the class.

Note: This command only applies to normal switch ports of type trunk/hybrid. This command does not apply to AC, CEP, or PNP ports. See the qos map-profile (Qumran) command for options for these types of ports.

Command Syntax

```
set cos <0-7>
no set cos
```

Parameters

`<0-7>`	Specify CoS value to assign for this class of traffic

Default

No default value is specified

Command Mode

Policy-map-class-qos mode

Applicability

This command was introduced in OcNOS version 3.0.

Examples

```
(config)#policy-map my_policy1
(config-pmap-qos)#class traffic_class2
(config-pmap-c-qos)#no set cos
(config-pmap-c-qos)#
```

© 2026 IP Infusion Inc. Proprietary

set dscp

Use this command for matching traffic classes set action as change DSCP in the egress packet to the prescribed value.

Use the `no` command to remove the assigned value from the class

Command Syntax

```
set dscp (<0-63>|af11| af12| af13| af21| af22| af23| af31| af32| af33| af41| af42|
  af43| cs1| cs2| cs3| cs4| cs5| cs6| cs7| default| ef )
```

```
no set dscp
```

Parameters

`<0-63>`	DSCP value
`af11`	DSCP (001011) decimal value 11
`af12`	DSCP (001100) decimal value 12
`af13`	DSCP (001101) decimal value 13
`af21`	DSCP (010101) decimal value 21
`af22`	DSCP (010110) decimal value 22
`af23`	DSCP (010111) decimal value 23
`af31`	DSCP (011111) decimal value 31
`af32`	DSCP (010000) decimal value 32
`af33`	DSCP (010001) decimal value 33
`af41`	DSCP (101001) decimal value 41
`af42`	DSCP (101010) decimal value 42
`af43`	DSCP (101011) decimal value 38
`cs1`	(Precedence 1) DSCP (001000) decimal value 8
`cs2`	(Precedence 2) DSCP (010000) decimal value 16
`cs3`	(Precedence 3) DSCP (011000) ecimal value 24
`cs4`	(Precedence 4) DSCP (100000) decimal value 32
`cs5`	(Precedence 5) DSCP (101000) decimal value 40
`cs6`	(Precedence 6) DSCP (110000) decimal value 48
`cs7`	(Precedence 7) DSCP (111000) decimal value 56
`default`	DSCP (000000) decimal value 0
`ef`	DSCP (101110) decimal value 46

Default

No default value is specified

Command Mode

Policy-map-class qos mode

Applicability

This command was introduced in OcNOS version 3.0.

Examples

```
#configure terminal
(config)#policy-map pmap1
(config-pmap-qos)#class cmap1
(config-pmap-c-qos)#set dscp af12
```

set precedence

Use this command for matching traffic classes set action as change precedence in the egress packet to the prescribed value.

Use the `no command` to leave the precedence value unchanged for the class

Command Syntax

```
set (precedence (<0-7>| critical| flash | flash-override|immediate|internet|
  network| priority| routine))

no set precedence
```

Parameters

`<0-7>`	Specify IP precedence value to assign for this class of traffic
`critical`	Critical precedence
`flash`	Flash precedence
`flash-override`	Flash override precedence
`immediate`	Immediate precedence
`internet`	Internetwork control precedence
`network`	Network control precedence
`priority`	Priority precedence
`routine`	Routine precedence

Default

No default value is specified

Command Mode

Policy-map-class qos mode

Applicability

This command was introduced in OcNOS version 3.0.

Examples

```
(config)# policy-map policy1
(config-pmap-qos)# class class2
(config-pmap-c-qos)#set precedence 3
(config-pmap-c-qos)#
```

© 2026 IP Infusion Inc. Proprietary

set queue

Use this command for matching traffic classes set action as change CoS in the egress packet to the prescribed value.

Use the `no command` to remove the assigned value from the class.

Command Syntax

```
set queue <0-7>
no set queue
```

Parameters

`<0-7>`	Specify queue value to assign for this class of traffic

Default

No default value is specified

Command Mode

Policy-map-class qos mode

Applicability

This command was introduced in OcNOS version 3.0.

Examples

```
(config)# policy-map my_policy1
(config-pmap-qos)# class traffic_class2
(config-pmap-c-qos)# set queue 3
(config-pmap-c-qos)#
```

show class-map

Use this command to display qos/queuing class maps.

Command Syntax

```
show class-map (type (qos|queuing)|) (NAME|)
```

Parameters

`type qos`	Specify the QoS type class map
`type queuing`	Specify the queuing type class map
`NAME`	Specify the named class map

Command Mode

Privileged executive mode

Applicability

This command was introduced in OcNOS version 3.0.

Examples

```
#show class-map c1
  Type qos class-maps
  ===================
      class-map c1
         match cos 3
```

show interface counters

Use this command to see the interface counters.

Note: VPLS BUM traffic will not be read in the interface queue statistics.

Command Syntax

```
show interface IFNAME counters queue-stats
```

Parameters

`IFNAME`	Interface name.

Command Mode

Exec

Applicability

This command was introduced in OcNOS version 3.0.

Examples

```
#show interface xe1 counters queue-stats
E - Egress, I - Ingress, Q-Size is in bytes
+---+------+--------+---------+---------+-----------+
| Q | Q-Sz | Tx pkt | Tx byte | Drp pkt | Drop byte |
+---+------+--------+---------+---------+-----------+
q0   629160 100      12000     0         0
q1   629160 0        0         0         0
q2   629160 0        0         0         0
q3   629160 0        0         0         0
q4   629160 0        0         0         0
q5   629160 0        0         0         0
q6   629160 0        0         0         0
q7   629160 0        0         0         0
```

© 2026 IP Infusion Inc. Proprietary

show policy-map

Use this command to display:

1. Type qos/queuing policy-map
2. Non zero statistics for type qos or queuing classes on interfaces

Command Syntax

```
show policy-map (NAME | (statistics (class NAME | type (qos | (queuing (default rate
  (gbps | kbps | mbps) | rate (gbps | kbps | mbps)|)))) | type (qos | queuing)|)
```

Parameters

`NAME`	Name of the policy
`qos`	Type qos policy-map
`statistics`	Displays QoS statistics
`class NAME`	Name of the class
`type qos`	Type of the QoS
`type queuing`	Type of the queuing
`queuing default`	Default queue of the port
`queuing rate`	Default queue of the rate
`gbps`	Rate in gbps
`kbps`	Rate in kbps
`mbps`	Rate in mbps

Command Mode

Privileged executive mode and Configure mode

Applicability

This command was introduced in OcNOS version 3.0.

Examples

```
P1#show policy-map statistics
Type qos class-map statistics:
+-------------------------------+----------------+----------------+----------------+------------------+
|            Class-map          |    Match pkts  |   Match bytes  |   Dropped pkts |   Dropped Bytes  |
+-------------------------------+----------------+----------------+----------------+------------------+
ce29/1
C1                              100              6796             0                0

Type queuing class-map statistics:
+-------------------------------+----------------+----------------+----------------+------------------+
|            Class-map          |    Total pkts  |   Total bytes  |   Dropped pkts |   Dropped Bytes  |
+-------------------------------+----------------+----------------+----------------+------------------+
ce1/1
q7                              1689             108888           0                0
ce2/1
q6                              8                912              0                0
ce5/1
q6                              8                912              0                0
ce29/1
q7                              104              7052             0                0
```

© 2026 IP Infusion Inc. Proprietary

```
P1#
P1#show policy-map type qos st
P1#show policy-map statistics type qos ?
|  Output modifiers

    Output redirection
    <cr>

P1#show policy-map statistics type qos
+---------------------------------+-----------------+-----------------+-----------------+------------------+
|            Class-map            |    Match pkts   |   Match bytes   |   Dropped pkts  |   Dropped Bytes  |
+---------------------------------+-----------------+-----------------+-----------------+------------------+
ce29/1
C1                                 100               6796              0                 0
P1#,
```

show policy-map interface

Use this command to:

- Display non-zero QoS statistics and configurations of type QoS and queuing policy-maps on an interface.
- Display the interfaces and policy-maps attached to them.

Note:

- Queuing policy-map drop statistics include all dropped packet counts in the queue (even policer drops).
- To check statistics, the user needs to enable the QoS statistics profile for Qumran devices. QoS can use either the ingress-ACL statistics profile or the ingress-QoS statistics profile. When ACL groups are configured on the same interface as QoS and both ACL and QoS require explicit counters, the user must configure the ingress-QoS statistics profile along with the ingress-ACL statistics profile. Note that this configuration has other limitations on statistics profiles. For more details, refer to the hardware-profile statistics command.

Command Syntax

```
show policy-map interface (NAME (type (qos (input | output |)| queuing)|
  (statistics (class NAME (type (qos | queuing (rate|)) | type (qos | queuing (rate
  (gbps|kbps|mbps)|))))) | class NAME)|brief)
```

Parameters

`interface IFNAME`	Specifies the interface name for which to display policy maps (maximum length 32 characters).
`type qos`	(Optional) Displays QoS policy maps for the specified interface.
`type qos input`	(Optional) Displays QoS input policy maps for the specified interface.
`type qos output`	(Optional) Displays QoS output policy maps for the specified interface.
`statistics class`	(Optional) Displays statistics of the particular class.
`statistics class NAME`	(Optional) Name of the particular class.
`statistics type`	(Optional) Displays statistics of the particular type.
`statistics type qos`	(Optional) Displays statistics of the particular QoS type.
`statistics type queing rate (gbps|kbps|mbps)`	(Optional) Displays statistics of the particular queuing rate type in gbps or kbps or mbps.
`interface class NAME`	(Optional) Displays statistics of the particular class.
`type queuing`	(Optional) Displays queuing policy-maps for the specified interface.
`brief`	Displays all QoS and queuing policy maps for the specified interface.

Command Mode

Privileged executive and Configure mode

Applicability

Introduced in OcNOS version 3.0.

Examples

To display the attached policy-maps on the interfaces, use the following command:

```
OcNOS#show policy-map interface xe1

Interface xe1
Type QoS statistics status : enabled

Service-policy (qos) input: A
----------------------------
Class-map (qos): A (match all)
 match vlan 2
 set cos 3
      matched      : 88091758 packets, 132137620500 bytes
      transmitted  : 88091758 packets, 132137620500 bytes

Class-map (qos): B (match all)
 match vlan 3


Service-policy (queuing) output: default-out-policy
-----------------------------------
Class-map (queuing): q0
  priority level 0

Class-map (queuing): q1
  priority level 1

Class-map (queuing): q2
  priority level 2

Class-map (queuing): q3
  priority level 3

Class-map (queuing): q4
  priority level 4

Class-map (queuing): q5
  priority level 5

Class-map (queuing): q6
  priority level 6

Class-map (queuing): q7
  priority level 7


OcNOS#show policy-map interface xe2
```

© 2026 IP Infusion Inc. Proprietary

```
Interface xe2
Type QoS statistics status : enabled

Service-policy (queuing) output: asd
-----------------------------------
Class-map (queuing): q0
  priority level 0

Class-map (queuing): q1
  priority level 1

Class-map (queuing): q2
  priority level 2

Class-map (queuing): q3
  priority level 3
      Output
         Total        : 88331951 packets, 132497934000 bytes
         Green        : 88332412 packets, 132498622500 bytes
         Yellow       : 0 packets, 0 bytes
      Dropped
         Total        : 0 packets, 0 bytes
         Green        : 0 packets, 0 bytes
         Yellow       : 0 packets, 0 bytes
         Red          : 0 packets, 0 bytes

Class-map (queuing): q4
  priority level 4

Class-map (queuing): q5
  priority level 5

Class-map (queuing): q6
  priority level 6

Class-map (queuing): q7
  priority level 7
```

To display the attached QoS policy-maps on the interfaces, use the following command:

```
OcNOS#show policy-map interface xe1 type qos input

Interface xe1
Type QoS statistics status : enabled

Service-policy (qos) input: A
----------------------------
Class-map (qos): A (match all)
 match vlan 2
 set cos 3
      matched      : 89779233 packets, 134668833000 bytes
      transmitted  : 89779233 packets, 134668833000 bytes

Class-map (qos): B (match all)
 match vlan 3

OcNOS#show policy-map interface xe2 type queuing
```

```
Interface xe2
Type QoS statistics status : enabled

Service-policy (queuing) output: asd
-----------------------------------
Class-map (queuing): q0
  priority level 0

Class-map (queuing): q1
  priority level 1

Class-map (queuing): q2
  priority level 2

Class-map (queuing): q3
  priority level 3
      Output
         Total         : 119883527 packets, 179825298000 bytes
         Green         : 119883992 packets, 179825995500 bytes
         Yellow        : 0 packets, 0 bytes
      Dropped
         Total         : 0 packets, 0 bytes
         Green         : 0 packets, 0 bytes
         Yellow        : 0 packets, 0 bytes
         Red           : 0 packets, 0 bytes

Class-map (queuing): q4
  priority level 4

Class-map (queuing): q5
  priority level 5

Class-map (queuing): q6
  priority level 6

Class-map (queuing): q7
  priority level 7
```

To display all QoS and queuing policy maps on the interfaces, use the following command:

```
#show policy-map interface brief
+------------+------------+------------------+
|            |    QOS     |                  |
| Interface  |-----+------|QUEUE             |
|            |INPUT|OUTPUT|                  |
+------------+-----+------+------------------+
 ce49                       default-out-policy
 ce50                       default-out-policy
```

show storm-control

Use the command to verify the BUM rate limit configured.

Note:

- Discard counters for BUM rate limiting are not supported on Qumran1 series platforms.
- Discards for BUM traffic support are only available on Qumran2 series platforms.

Command Syntax

```
show storm-control (INTERFACE-NAME|)
show storm-control (INTERFACE-NAME|)discards
```

Parameters

INTERFACE-NAME	Specifies the interface name for which to display the storm control details

Command Mode

Privileged executive mode

Applicability

Introduced before OcNOS version 1.3.

Example

```
#show storm-control
*The hardware applicable value is displayed
Port    BcastLevel(burst)       McastLevel(burst)   DlfLevel(burst)
Discards
ce5     100.000 kbps(0 Kbps)   100.0000%(0 Kbps)    1000 mbps(10 Kbps)    0
ce9/1   100.000 kbps(0 Kbps)   100.0000%(0 Kbps)    1000 mbps(10 Kbps)    0

#show storm-control discards
Port    BcastLevel(burst)       McastLevel(burst)   DlfLevel(burst)
Discards
ce5     0                       0                   0                     0
ce9/1   0                       0                   0                     0
ce0     0                       0                   0                     0
```

show qos-profile

Use this command to show the all configured QoS profiles configurations like type, name, maps configured except for the default maps, attached info (to how many interfaces it is attached) etc,

Command Syntax

```
show qos-profile (type (cos-to-queue | dscp-encap | dscp-to-dscp | dscp-to-queue |
  exp-to-queue |  queue-to-exp | precedence-to-precedence | precedence-to-queue |
  queue-color-to-cos)(NAME|)
```

Parameters

cos-to-queue	profile for cos to Traffic-Class map
dscp-encap	profile for dscp encapsulation map
dscp-to-dscp	profile for in Dscp to Dscp map
dscp-to-queue	profile for dscp to Traffic-Class map
exp-to-queue	profile for exp to Traffic-Class map
precedence-to-precedence	profile for precedence to precedence map
precedence-to-queue	profile for precedence to Traffic-Class map
queue-color-to-cos	profile for Traffic-Class color to cos map
NAME	Profile map name (maximum 32 characters)

Command Mode

Privileged exec, config, interface, class-map, policy-map and policy-map-class

Applicability

This command was introduced in OcNOS version 3.0.

Examples

```
#show policy-map interface xe1

Interface xe1
Type QoS statistics status : enabled

Service-policy (qos) input: A
----------------------------
Class-map (qos): A (match all)
 match vlan 2
 set cos 3
      matched      : 88091758 packets, 132137620500 bytes
      transmitted  : 88091758 packets, 132137620500 bytes

Class-map (qos): B (match all)
 match vlan 3


Service-policy (queuing) output: default-out-policy
-----------------------------------
Class-map (queuing): q0
```

```
  priority level 0

Class-map (queuing): q1
  priority level 1
Class-map (queuing): q2
  priority level 2

Class-map (queuing): q3
  priority level 3

Class-map (queuing): q4
  priority level 4

Class-map (queuing): q5
  priority level 5

Class-map (queuing): q6
  priority level 6

Class-map (queuing): q7
  priority level 7


#show policy-map interface xe2

Interface xe2
Type QoS statistics status : enabled

Service-policy (queuing) output: asd
-----------------------------------
Class-map (queuing): q0
  priority level 0

Class-map (queuing): q1
  priority level 1

Class-map (queuing): q2
  priority level 2

Class-map (queuing): q3
  priority level 3
      Output
         Total        : 88331951 packets, 132497934000 bytes
         Green        : 88332412 packets, 132498622500 bytes
         Yellow       : 0 packets, 0 bytes
      Dropped
         Total        : 0 packets, 0 bytes
         Green        : 0 packets, 0 bytes
         Yellow       : 0 packets, 0 bytes
         Red          : 0 packets, 0 bytes

Class-map (queuing): q4
  priority level 4

Class-map (queuing): q5
  priority level 5
```

```
Class-map (queuing): q6
  priority level 6

Class-map (queuing): q7
  priority level 7

#show policy-map interface xe1 type qos input

Interface xe1
Type QoS statistics status : enabled

Service-policy (qos) input: A
----------------------------
Class-map (qos): A (match all)
 match vlan 2
 set cos 3
      matched      : 89779233 packets, 134668833000 bytes
      transmitted  : 89779233 packets, 134668833000 bytes

Class-map (qos): B (match all)
 match vlan 3

#show policy-map interface xe2 type queuing

Interface xe2
Type QoS statistics status : enabled

Service-policy (queuing) output: asd
-----------------------------------
Class-map (queuing): q0
  priority level 0

Class-map (queuing): q1
  priority level 1

Class-map (queuing): q2
  priority level 2

Class-map (queuing): q3
  priority level 3
      Output
         Total        : 119883527 packets, 179825298000 bytes
         Green        : 119883992 packets, 179825995500 bytes
         Yellow       : 0 packets, 0 bytes
      Dropped
         Total        : 0 packets, 0 bytes
         Green        : 0 packets, 0 bytes
         Yellow       : 0 packets, 0 bytes
         Red          : 0 packets, 0 bytes

Class-map (queuing): q4
  priority level 4

Class-map (queuing): q5
  priority level 5

Class-map (queuing): q6
```

```
  priority level 6

Class-map (queuing): q7
  priority level 7

#show policy-map interface br
+-------------+------------+-------------------+
|             |    QOS     |                   |
| Interface   |-----+------|QUEUE              |
|             |INPUT|OUTPUT|                   |
+-------------+-----+------+-------------------+
 ce49                       default-out-policy
 ce50                       default-out-policy
```

© 2026 IP Infusion Inc. Proprietary

show qos-profile interface

Use this command to show the all configured QoS profiles configurations attached on an interface.

Command Syntax

```
show qos-profile interface NAME
```

Parameters

NAME	Profile map name (maximum 32 characters)

Command Mode

Privileged exec, config, interface, class-map, policy-map and policy-map-class

Applicability

This command was introduced in OcNOS version 3.0.

Examples

L3 interface

```
OcNOS#show qos-profile interface ce0
profile name: default
 profile type: dscp-to-queue (Ingress)
 mapping:
 Ingress map for dscp to queue-color
 -------+----------------- | -------+----------------- | -------+-------------
---- | -------+-----------------
  INPUT |     OUTPUT       |  INPUT |     OUTPUT       |  INPUT |     OUTPUT
|  INPUT |     OUTPUT
 -------+----------------- | -------+----------------- | -------+-------------
---- | -------+-----------------
  DSCP   |   TC   | Color     |  DSCP   |   TC   | Color     |  DSCP   |   TC   |
Color    |  DSCP   |   TC   | Color
 -------+-------+--------- | -------+-------+--------- | -------+-------+-----
---- | -------+-------+---------
   0      0       green    |  16      2       green    |  32      4
green    |  48      6       green
   1      0       green    |  17      2       green    |  33      4
green    |  49      6       green
   2      0       green    |  18      2       green    |  34      4
green    |  50      6       green
   3      0       green    |  19      2       green    |  35      4
green    |  51      6       green
   4      0       green    |  20      2       yellow   |  36      4
yellow   |  52      6       green
   5      0       green    |  21      2       green    |  37      4
green    |  53      6       green
   6      0       green    |  22      2       red      |  38      4
red      |  54      6       green
   7      0       green    |  23      2       green    |  39      4
green    |  55      6       green
   8      1       green    |  24      3       green    |  40      5
green    |  56      7       green
   9      1       green    |  25      3       green    |  41      5
green    |  57      7       green
```

```
  10      1       green    |  26      3       green    |  42      5
green    |  58      7      green
  11      1       green    |  27      3       green    |  43      5
green    |  59      7      green
  12      1       yellow   |  28      3       yellow   |  44      5
green    |  60      7      green
  13      1       green    |  29      3       green    |  45      5
green    |  61      7      green
  14      1       red      |  30      3       red      |  46      5
green    |  62      7      green
  15      1       green    |  31      3       green    |  47      5
green    |  63      7      green

profile name: default
 profile type: queue-color-to-dscp (Egress)
 Status: Inactive
mapping:
 ----------------+-------- | ----------------+-------- | ----------------+----
----
      INPUT      | OUTPUT  |      INPUT      | OUTPUT  |      INPUT      |
OUTPUT
 ----------------+-------- | ----------------+-------- | ----------------+----
----
    TC  | Color  | DSCP    |    TC  | Color  | DSCP    |    TC  | Color  |
DSCP
 -------+--------+-------- | -------+--------+-------- | -------+--------+----
----
  0       green     0      |  0       yellow    0      |  0       red       0
  1       green    10      |  1       yellow   12      |  1       red      14
  2       green    18      |  2       yellow   20      |  2       red      22
  3       green    26      |  3       yellow   28      |  3       red      30
  4       green    34      |  4       yellow   36      |  4       red      38
  5       green    40      |  5       yellow   40      |  5       red      40
  6       green    48      |  6       yellow   48      |  6       red      48
  7       green    56      |  7       yellow   56      |  7       red      56

OcNOS#
```

L2 interface

```
OcNOS#show qos-profile interface ce2
profile name: default
 profile type: cos-to-queue (Ingress)
 mapping:
 ---------------+----------------- | ---------------+-----------------
      INPUT     |      OUTPUT      |      INPUT     |      OUTPUT
 ---------------+----------------- | ---------------+-----------------
   COS  |  DEI  |   TC  |  Color   |   COS  |  DEI  |   TC  |  Color
 -------+-------+-------+--------- | -------+-------+-------+---------
  0       0       0        green   |  0       1       0        yellow
  1       0       1        green   |  1       1       1        yellow
  2       0       2        green   |  2       1       2        yellow
  3       0       3        green   |  3       1       3        yellow
  4       0       4        green   |  4       1       4        yellow
  5       0       5        green   |  5       1       5        yellow
  6       0       6        green   |  6       1       6        yellow
  7       0       7        green   |  7       1       7        yellow

profile name: default
```

```
 profile type: queue-color-to-cos (Egress)
 Status: Inactive
mapping:
 ----------------+-------- | ----------------+-------- | ----------------+----
----
      INPUT      | OUTPUT  |      INPUT      | OUTPUT  |      INPUT      |
OUTPUT
 ----------------+-------- | ----------------+-------- | ----------------+----
----
    TC  | Color  | COS     |    TC  | Color  | COS     |    TC  | Color  | COS
 -------+--------+-------- | -------+--------+-------- | -------+--------+----
----
  0       green    0       |  0       yellow   0       |  0       red      0
  1       green    1       |  1       yellow   1       |  1       red      1
  2       green    2       |  2       yellow   2       |  2       red      2
  3       green    3       |  3       yellow   3       |  3       red      3
  4       green    4       |  4       yellow   4       |  4       red      4
  5       green    5       |  5       yellow   5       |  5       red      5
  6       green    6       |  6       yellow   6       |  6       red      6
  7       green    7       |  7       yellow   7       |  7       red      7
```

show queuing interface

Use this command to see the configurations of queues that are attached to an interface.

Command Syntax

```
show queuing interface NAME
```

Parameters

NAME	Interface name.

Command Mode

Privileged exec & config mode

Applicability

This command was introduced in OcNOS version 3.0.

Examples

```
# show queuing interface xe1/1
Egress Queuing for Ethernet xe1/1 [System]
------------------------------------------------------------------------------
----------------------
L0              L1              L2              Group  PrioLevel   Shape
Bandwidth
------------------------------------------------------------------------------
----------------------
q0                                                       -      High        -
-
q1                                                       -      High        -
-
q2                                                       -      High        -
-
q3                                                       -      High        -
-
q4                                                       -      High        -
-
q5                                                       -      High        -
-
q6                                                       -      High        -
-
q7                                                       -      High        -
-
```

show running-config qos

Use this command to show the user configured QoS configurations.

Command Syntax

```
show running-config qos (all|)
```

Parameters

`all`	Show all QoS related configuration information including all defaults.

Command Mode

Exec, config, interface, class-map, policy-map and policy-map-class

Applicability

This command was introduced in OcNOS version 3.0.

Examples

```
#show running-config qos
qos enable
!
!
#show running-config qos  ?
  all  diplay all qos info including defaults
  |    Output modifiers
  >    Output redirection
  <cr>

#show running-config qos  all
qos enable
!
qos profile cos-to-queue default
 cos 0 dei 0 queue 0 color green
 cos 0 dei 1 queue 0 color yellow
 cos 0 queue 0
 cos 1 dei 0 queue 1 color green
 cos 1 dei 1 queue 1 color yellow
 cos 1 queue 1
 cos 2 dei 0 queue 2 color green
 cos 2 dei 1 queue 2 color yellow
 cos 2 queue 2
 cos 3 dei 0 queue 3 color green
 cos 3 dei 1 queue 3 color yellow
 cos 3 queue 3
 cos 4 dei 0 queue 4 color green
 cos 4 dei 1 queue 4 color yellow
 cos 4 queue 4
 cos 5 dei 0 queue 5 color green
 cos 5 dei 1 queue 5 color yellow
 cos 5 queue 5
 cos 6 dei 0 queue 6 color green
 cos 6 dei 1 queue 6 color yellow
 cos 6 queue 6
```

```
 cos 7 dei 0 queue 7 color green
 cos 7 dei 1 queue 7 color yellow
 cos 7 queue 7
!
qos profile queue-color-to-cos default
 queue 0 color green cos 0
 queue 0 color yellow cos 0
 queue 0 color red cos 0
 queue 0 cos 0
 queue 1 color green cos 1
 queue 1 color yellow cos 1
 queue 1 color red cos 1
 queue 1 cos 1
 queue 2 color green cos 2
 queue 2 color yellow cos 2
 queue 2 color red cos 2
 queue 2 cos 2
 queue 3 color green cos 3
 queue 3 color yellow cos 3
 queue 3 color red cos 3
 queue 3 cos 3
 queue 4 color green cos 4
 queue 4 color yellow cos 4
 queue 4 color red cos 4
 queue 4 cos 4
 queue 5 color green cos 5
 queue 5 color yellow cos 5
 queue 5 color red cos 5
 queue 5 cos 5
 queue 6 color green cos 6
 queue 6 color yellow cos 6
 queue 6 color red cos 6
 queue 6 cos 6
 queue 7 color green cos 7
 queue 7 color yellow cos 7
 queue 7 color red cos 7
 queue 7 cos 7
!
qos profile dscp-to-queue default
 dscp 0 queue 0 color green dscp 0
 dscp 1 queue 0 color green dscp 1
 dscp 2 queue 0 color green dscp 2
 dscp 3 queue 0 color green dscp 3
 dscp 4 queue 0 color green dscp 4
 dscp 5 queue 0 color green dscp 5
 dscp 6 queue 0 color green dscp 6
 dscp 7 queue 0 color green dscp 7
 dscp 8 queue 1 color green dscp 8
 dscp 9 queue 1 color green dscp 9
 dscp 10 queue 1 color green dscp 10
 dscp 11 queue 1 color green dscp 11
 dscp 12 queue 1 color yellow dscp 12
 dscp 13 queue 1 color green dscp 13
 dscp 14 queue 1 color yellow dscp 14
 dscp 15 queue 1 color green dscp 15
 dscp 16 queue 2 color green dscp 16
 dscp 17 queue 2 color green dscp 17
```

```
 dscp 18 queue 2 color green dscp 18
 dscp 19 queue 2 color green dscp 19
 dscp 20 queue 2 color yellow dscp 20
 dscp 21 queue 2 color green dscp 21
 dscp 22 queue 2 color yellow dscp 22
 dscp 23 queue 2 color green dscp 23
 dscp 24 queue 3 color green dscp 24
 dscp 25 queue 3 color green dscp 25
 dscp 26 queue 3 color green dscp 26
 dscp 27 queue 3 color green dscp 27
 dscp 28 queue 3 color yellow dscp 28
 dscp 29 queue 3 color green dscp 29
 dscp 30 queue 3 color yellow dscp 30
 dscp 31 queue 3 color green dscp 31
 dscp 32 queue 4 color green dscp 32
 dscp 33 queue 4 color green dscp 33
 dscp 34 queue 4 color green dscp 34
 dscp 35 queue 4 color green dscp 35
 dscp 36 queue 4 color yellow dscp 36
 dscp 37 queue 4 color green dscp 37
 dscp 38 queue 4 color yellow dscp 38
 dscp 39 queue 4 color green dscp 39
 dscp 40 queue 5 color green dscp 40
 dscp 41 queue 5 color green dscp 41
 dscp 42 queue 5 color green dscp 42
 dscp 43 queue 5 color green dscp 43
 dscp 44 queue 5 color green dscp 44
 dscp 45 queue 5 color green dscp 45
 dscp 46 queue 5 color green dscp 46
 dscp 47 queue 5 color green dscp 47
 dscp 48 queue 6 color green dscp 48
 dscp 49 queue 6 color green dscp 49
 dscp 50 queue 6 color green dscp 50
 dscp 51 queue 6 color green dscp 51
 dscp 52 queue 6 color green dscp 52
 dscp 53 queue 6 color green dscp 53
 dscp 54 queue 6 color green dscp 54
 dscp 55 queue 6 color green dscp 55
 dscp 56 queue 7 color green dscp 56
 dscp 57 queue 7 color green dscp 57
 dscp 58 queue 7 color green dscp 58
 dscp 59 queue 7 color green dscp 59
 dscp 60 queue 7 color green dscp 60
 dscp 61 queue 7 color green dscp 61
 dscp 62 queue 7 color green dscp 62
 dscp 63 queue 7 color green dscp 63
!
qos profile dscp-to-dscp default
 dscp 0 color green dscp 0
 dscp 0 color yellow dscp 0
 dscp 0 color red dscp 0
 dscp 0 dscp 0
 dscp 1 color green dscp 1
 dscp 1 color yellow dscp 1
 dscp 1 color red dscp 1
 dscp 1 dscp 1
 dscp 2 color green dscp 2
```

© 2026 IP Infusion Inc. Proprietary

```
dscp 2 color yellow dscp 2
dscp 2 color red dscp 2
dscp 2 dscp 2
dscp 3 color green dscp 3
dscp 3 color yellow dscp 3
dscp 3 color red dscp 3
dscp 3 dscp 3
dscp 4 color green dscp 4
dscp 4 color yellow dscp 4
dscp 4 color red dscp 4
dscp 4 dscp 4
dscp 5 color green dscp 5
dscp 5 color yellow dscp 5
dscp 5 color red dscp 5
dscp 5 dscp 5
dscp 6 color green dscp 6
dscp 6 color yellow dscp 6
dscp 6 color red dscp 6
dscp 6 dscp 6
dscp 7 color green dscp 7
dscp 7 color yellow dscp 7
dscp 7 color red dscp 7
dscp 7 dscp 7
dscp 8 color green dscp 8
dscp 8 color yellow dscp 8
dscp 8 color red dscp 8
dscp 8 dscp 8
dscp 9 color green dscp 9
dscp 9 color yellow dscp 9
dscp 9 color red dscp 9
dscp 9 dscp 9
dscp 10 color green dscp 10
dscp 10 color yellow dscp 10
dscp 10 color red dscp 10
dscp 10 dscp 10
dscp 11 color green dscp 11
dscp 11 color yellow dscp 11
dscp 11 color red dscp 11
dscp 11 dscp 11
dscp 12 color green dscp 12
dscp 12 color yellow dscp 12
dscp 12 color red dscp 12
dscp 12 dscp 12
dscp 13 color green dscp 13
dscp 13 color yellow dscp 13
dscp 13 color red dscp 13
dscp 13 dscp 13
dscp 14 color green dscp 14
dscp 14 color yellow dscp 14
dscp 14 color red dscp 14
dscp 14 dscp 14
dscp 15 color green dscp 15
dscp 15 color yellow dscp 15
dscp 15 color red dscp 15
dscp 15 dscp 15
dscp 16 color green dscp 16
dscp 16 color yellow dscp 16
```

```
dscp 16 color red dscp 16
dscp 16 dscp 16
dscp 17 color green dscp 17
dscp 17 color yellow dscp 17
dscp 17 color red dscp 17
dscp 17 dscp 17
dscp 18 color green dscp 18
dscp 18 color yellow dscp 18
dscp 18 color red dscp 18
dscp 18 dscp 18
dscp 19 color green dscp 19
dscp 19 color yellow dscp 19
dscp 19 color red dscp 19
dscp 19 dscp 19
dscp 20 color green dscp 20
dscp 20 color yellow dscp 20
dscp 20 color red dscp 20
dscp 20 dscp 20
dscp 21 color green dscp 21
dscp 21 color yellow dscp 21
dscp 21 color red dscp 21
dscp 21 dscp 21
dscp 22 color green dscp 22
dscp 22 color yellow dscp 22
dscp 22 color red dscp 22
dscp 22 dscp 22
dscp 23 color green dscp 23
dscp 23 color yellow dscp 23
dscp 23 color red dscp 23
dscp 23 dscp 23
dscp 24 color green dscp 24
dscp 24 color yellow dscp 24
dscp 24 color red dscp 24
dscp 24 dscp 24
dscp 25 color green dscp 25
dscp 25 color yellow dscp 25
dscp 25 color red dscp 25
dscp 25 dscp 25
dscp 26 color green dscp 26
dscp 26 color yellow dscp 26
dscp 26 color red dscp 26
dscp 26 dscp 26
dscp 27 color green dscp 27
dscp 27 color yellow dscp 27
dscp 27 color red dscp 27
dscp 27 dscp 27
dscp 28 color green dscp 28
dscp 28 color yellow dscp 28
dscp 28 color red dscp 28
dscp 28 dscp 28
dscp 29 color green dscp 29
dscp 29 color yellow dscp 29
dscp 29 color red dscp 29
dscp 29 dscp 29
dscp 30 color green dscp 30
dscp 30 color yellow dscp 30
dscp 30 color red dscp 30
```

```
dscp 30 dscp 30
dscp 31 color green dscp 31
dscp 31 color yellow dscp 31
dscp 31 color red dscp 31
dscp 31 dscp 31
dscp 32 color green dscp 32
dscp 32 color yellow dscp 32
dscp 32 color red dscp 32
dscp 32 dscp 32
dscp 33 color green dscp 33
dscp 33 color yellow dscp 33
dscp 33 color red dscp 33
dscp 33 dscp 33
dscp 34 color green dscp 34
dscp 34 color yellow dscp 34
dscp 34 color red dscp 34
dscp 34 dscp 34
dscp 35 color green dscp 35
dscp 35 color yellow dscp 35
dscp 35 color red dscp 35
dscp 35 dscp 35
dscp 36 color green dscp 36
dscp 36 color yellow dscp 36
dscp 36 color red dscp 36
dscp 36 dscp 36
dscp 37 color green dscp 37
dscp 37 color yellow dscp 37
dscp 37 color red dscp 37
dscp 37 dscp 37
dscp 38 color green dscp 38
dscp 38 color yellow dscp 38
dscp 38 color red dscp 38
dscp 38 dscp 38
dscp 39 color green dscp 39
dscp 39 color yellow dscp 39
dscp 39 color red dscp 39
dscp 39 dscp 39
dscp 40 color green dscp 40
dscp 40 color yellow dscp 40
dscp 40 color red dscp 40
dscp 40 dscp 40
dscp 41 color green dscp 41
dscp 41 color yellow dscp 41
dscp 41 color red dscp 41
dscp 41 dscp 41
dscp 42 color green dscp 42
dscp 42 color yellow dscp 42
dscp 42 color red dscp 42
dscp 42 dscp 42
dscp 43 color green dscp 43
dscp 43 color yellow dscp 43
dscp 43 color red dscp 43
dscp 43 dscp 43
dscp 44 color green dscp 44
dscp 44 color yellow dscp 44
dscp 44 color red dscp 44
dscp 44 dscp 44
```

© 2026 IP Infusion Inc. Proprietary

```
dscp 45 color green dscp 45
dscp 45 color yellow dscp 45
dscp 45 color red dscp 45
dscp 45 dscp 45
dscp 46 color green dscp 46
dscp 46 color yellow dscp 46
dscp 46 color red dscp 46
dscp 46 dscp 46
dscp 47 color green dscp 47
dscp 47 color yellow dscp 47
dscp 47 color red dscp 47
dscp 47 dscp 47
dscp 48 color green dscp 48
dscp 48 color yellow dscp 48
dscp 48 color red dscp 48
dscp 48 dscp 48
dscp 49 color green dscp 49
dscp 49 color yellow dscp 49
dscp 49 color red dscp 49
dscp 49 dscp 49
dscp 50 color green dscp 50
dscp 50 color yellow dscp 50
dscp 50 color red dscp 50
dscp 50 dscp 50
dscp 51 color green dscp 51
dscp 51 color yellow dscp 51
dscp 51 color red dscp 51
dscp 51 dscp 51
dscp 52 color green dscp 52
dscp 52 color yellow dscp 52
dscp 52 color red dscp 52
dscp 52 dscp 52
dscp 53 color green dscp 53
dscp 53 color yellow dscp 53
dscp 53 color red dscp 53
dscp 53 dscp 53
dscp 54 color green dscp 54
dscp 54 color yellow dscp 54
dscp 54 color red dscp 54
dscp 54 dscp 54
dscp 55 color green dscp 55
dscp 55 color yellow dscp 55
dscp 55 color red dscp 55
dscp 55 dscp 55
dscp 56 color green dscp 56
dscp 56 color yellow dscp 56
dscp 56 color red dscp 56
dscp 56 dscp 56
dscp 57 color green dscp 57
dscp 57 color yellow dscp 57
dscp 57 color red dscp 57
dscp 57 dscp 57
dscp 58 color green dscp 58
dscp 58 color yellow dscp 58
dscp 58 color red dscp 58
dscp 58 dscp 58
dscp 59 color green dscp 59
```

```
 dscp 59 color yellow dscp 59
 dscp 59 color red dscp 59
 dscp 59 dscp 59
 dscp 60 color green dscp 60
 dscp 60 color yellow dscp 60
 dscp 60 color red dscp 60
 dscp 60 dscp 60
 dscp 61 color green dscp 61
dscp 61 color yellow dscp 61
 dscp 61 color red dscp 61
 dscp 61 dscp 61
 dscp 62 color green dscp 62
 dscp 62 color yellow dscp 62
 dscp 62 color red dscp 62
 dscp 62 dscp 62
 dscp 63 color green dscp 63
 dscp 63 color yellow dscp 63
 dscp 63 color red dscp 63
 dscp 63 dscp 63
!
qos profile exp-to-queue default
 exp 0 queue 0 color green
 exp 1 queue 1 color green
 exp 2 queue 2 color green
 exp 3 queue 3 color green
 exp 4 queue 4 color green
 exp 5 queue 5 color green
 exp 6 queue 6 color green
 exp 7 queue 7 color green
!
qos profile dscp-to-exp default
 dscp 0 color green exp 0
 dscp 0 color yellow exp 0
 dscp 0 color red exp 0
 dscp 0 exp 0
 dscp 1 color green exp 0
 dscp 1 color yellow exp 0
 dscp 1 color red exp 0
 dscp 1 exp 0
 dscp 2 color green exp 0
 dscp 2 color yellow exp 0
 dscp 2 color red exp 0
 dscp 2 exp 0
 dscp 3 color green exp 0
 dscp 3 color yellow exp 0
 dscp 3 color red exp 0
 dscp 3 exp 0
 dscp 4 color green exp 0
 dscp 4 color yellow exp 0
 dscp 4 color red exp 0
 dscp 4 exp 0
 dscp 5 color green exp 0
 dscp 5 color yellow exp 0
 dscp 5 color red exp 0
 dscp 5 exp 0
 dscp 6 color green exp 0
 dscp 6 color yellow exp 0
```

```
dscp 6 color red exp 0
dscp 6 exp 0
dscp 7 color green exp 0
dscp 7 color yellow exp 0
dscp 7 color red exp 0
dscp 7 exp 0
dscp 8 color green exp 1
dscp 8 color yellow exp 1
dscp 8 color red exp 1
dscp 8 exp 1
dscp 9 color green exp 1
dscp 9 color yellow exp 1
dscp 9 color red exp 1
dscp 9 exp 1
dscp 10 color green exp 1
dscp 10 color yellow exp 1
dscp 10 color red exp 1
dscp 10 exp 1
dscp 11 color green exp 1
dscp 11 color yellow exp 1
dscp 11 color red exp 1
dscp 11 exp 1
dscp 12 color green exp 1
dscp 12 color yellow exp 1
dscp 12 color red exp 1
dscp 12 exp 1
dscp 13 color green exp 1
dscp 13 color yellow exp 1
dscp 13 color red exp 1
dscp 13 exp 1
dscp 14 color green exp 1
dscp 14 color yellow exp 1
dscp 14 color red exp 1
dscp 14 exp 1
dscp 15 color green exp 1
dscp 15 color yellow exp 1
dscp 15 color red exp 1
dscp 15 exp 1
dscp 16 color green exp 2
dscp 16 color yellow exp 2
dscp 16 color red exp 2
dscp 16 exp 2
dscp 17 color green exp 2
dscp 17 color yellow exp 2
dscp 17 color red exp 2
dscp 17 exp 2
dscp 18 color green exp 2
dscp 18 color yellow exp 2
dscp 18 color red exp 2
dscp 18 exp 2
dscp 19 color green exp 2
dscp 19 color yellow exp 2
dscp 19 color red exp 2
dscp 19 exp 2
dscp 20 color green exp 2
dscp 20 color yellow exp 2
dscp 20 color red exp 2
```

```
dscp 20 exp 2
dscp 21 color green exp 2
dscp 21 color yellow exp 2
dscp 21 color red exp 2
dscp 21 exp 2
dscp 22 color green exp 2
dscp 22 color yellow exp 2
dscp 22 color red exp 2
dscp 22 exp 2
dscp 23 color green exp 2
dscp 23 color yellow exp 2
dscp 23 color red exp 2
dscp 23 exp 2
dscp 24 color green exp 3
dscp 24 color yellow exp 3
dscp 24 color red exp 3
dscp 24 exp 3
dscp 25 color green exp 3
dscp 25 color yellow exp 3
dscp 25 color red exp 3
dscp 25 exp 3
dscp 26 color green exp 3
dscp 26 color yellow exp 3
dscp 26 color red exp 3
dscp 26 exp 3
dscp 27 color green exp 3
dscp 27 color yellow exp 3
dscp 27 color red exp 3
dscp 27 exp 3
dscp 28 color green exp 3
dscp 28 color yellow exp 3
dscp 28 color red exp 3
dscp 28 exp 3
dscp 29 color green exp 3
dscp 29 color yellow exp 3
dscp 29 color red exp 3
dscp 29 exp 3
dscp 30 color green exp 3
dscp 30 color yellow exp 3
dscp 30 color red exp 3
dscp 30 exp 3
dscp 31 color green exp 3
dscp 31 color yellow exp 3
dscp 31 color red exp 3
dscp 31 exp 3
dscp 32 color green exp 4
dscp 32 color yellow exp 4
dscp 32 color red exp 4
dscp 32 exp 4
dscp 33 color green exp 4
dscp 33 color yellow exp 4
dscp 33 color red exp 4
dscp 33 exp 4
dscp 34 color green exp 4
dscp 34 color yellow exp 4
dscp 34 color red exp 4
dscp 34 exp 4
```

© 2026 IP Infusion Inc. Proprietary

```
dscp 35 color green exp 4
dscp 35 color yellow exp 4
dscp 35 color red exp 4
dscp 35 exp 4
dscp 36 color green exp 4
dscp 36 color yellow exp 4
dscp 36 color red exp 4
dscp 36 exp 4
dscp 37 color green exp 4
dscp 37 color yellow exp 4
dscp 37 color red exp 4
dscp 37 exp 4
dscp 38 color green exp 4
dscp 38 color yellow exp 4
dscp 38 color red exp 4
dscp 38 exp 4
dscp 39 color green exp 4
dscp 39 color yellow exp 4
dscp 39 color red exp 4
dscp 39 exp 4
dscp 40 color green exp 5
dscp 40 color yellow exp 5
dscp 40 color red exp 5
dscp 40 exp 5
dscp 41 color green exp 5
dscp 41 color yellow exp 5
dscp 41 color red exp 5
dscp 41 exp 5
dscp 42 color green exp 5
dscp 42 color yellow exp 5
dscp 42 color red exp 5
dscp 42 exp 5
dscp 43 color green exp 5
dscp 43 color yellow exp 5
dscp 43 color red exp 5
dscp 43 exp 5
dscp 44 color green exp 5
dscp 44 color yellow exp 5
dscp 44 color red exp 5
dscp 44 exp 5
dscp 45 color green exp 5
dscp 45 color yellow exp 5
dscp 45 color red exp 5
dscp 45 exp 5
dscp 46 color green exp 5
dscp 46 color yellow exp 5
dscp 46 color red exp 5
dscp 46 exp 5
dscp 47 color green exp 5
dscp 47 color yellow exp 5
dscp 47 color red exp 5
dscp 47 exp 5
dscp 48 color green exp 6
dscp 48 color yellow exp 6
dscp 48 color red exp 6
dscp 48 exp 6
dscp 49 color green exp 6
```

```
dscp 49 color yellow exp 6
dscp 49 color red exp 6
dscp 49 exp 6
dscp 50 color green exp 6
dscp 50 color yellow exp 6
dscp 50 color red exp 6
dscp 50 exp 6
dscp 51 color green exp 6
dscp 51 color yellow exp 6
dscp 51 color red exp 6
dscp 51 exp 6
dscp 52 color green exp 6
dscp 52 color yellow exp 6
dscp 52 color red exp 6
dscp 52 exp 6
dscp 53 color green exp 6
dscp 53 color yellow exp 6
dscp 53 color red exp 6
dscp 53 exp 6
dscp 54 color green exp 6
dscp 54 color yellow exp 6
dscp 54 color red exp 6
dscp 54 exp 6
dscp 55 color green exp 6
dscp 55 color yellow exp 6
dscp 55 color red exp 6
dscp 55 exp 6
dscp 56 color green exp 7
dscp 56 color yellow exp 7
dscp 56 color red exp 7
dscp 56 exp 7
dscp 57 color green exp 7
dscp 57 color yellow exp 7
dscp 57 color red exp 7
dscp 57 exp 7
dscp 58 color green exp 7
dscp 58 color yellow exp 7
dscp 58 color red exp 7
dscp 58 exp 7
dscp 59 color green exp 7
dscp 59 color yellow exp 7
dscp 59 color red exp 7
dscp 59 exp 7
dscp 60 color green exp 7
dscp 60 color yellow exp 7
dscp 60 color red exp 7
dscp 60 exp 7
dscp 61 color green exp 7
dscp 61 color yellow exp 7
dscp 61 color red exp 7
dscp 61 exp 7
dscp 62 color green exp 7
dscp 62 color yellow exp 7
dscp 62 color red exp 7
dscp 62 exp 7
dscp 63 color green exp 7
dscp 63 color yellow exp 7
```

© 2026 IP Infusion Inc. Proprietary

```
 dscp 63 color red exp 7
 dscp 63 exp 7
!
policy-map type queuing default default-out-policy
 class type queuing default q0
  priority level 0
  exit
 class type queuing default q1
  priority level 1
  exit
 class type queuing default q2
  priority level 2
  exit
 class type queuing default q3
  priority level 3
  exit
 class type queuing default q4
  priority level 4
  exit
 class type queuing default q5
  priority level 5
  exit
class type queuing default q6
  priority level 6
  exit
 class type queuing default q7
  priority level 7
  exit
!
interface ce49
 service-policy type queuing output default-out-policy
!
interface ce50
 service-policy type queuing output default-out-policy
!
interface ce51
 service-policy type queuing output default-out-policy
!
interface ce52
 service-policy type queuing output default-out-policy
!
interface ce53
 service-policy type queuing output default-out-policy
!
interface ce54
 service-policy type queuing output default-out-policy
!
interface xe1
 service-policy type queuing output default-out-policy
!
interface xe2
 service-policy type queuing output default-out-policy
!
interface xe3
 service-policy type queuing output default-out-policy
!
interface xe4
```

```
 service-policy type queuing output default-out-policy
!
interface xe5
 service-policy type queuing output default-out-policy
!
interface xe6
 service-policy type queuing output default-out-policy
!
interface xe7
 service-policy type queuing output default-out-policy
!
interface xe8
 service-policy type queuing output default-out-policy
!
interface xe9
 service-policy type queuing output default-out-policy
!
interface xe10
 service-policy type queuing output default-out-policy
!
interface xe11
 service-policy type queuing output default-out-policy
!
interface xe12
 service-policy type queuing output default-out-policy
!
interface xe13
 service-policy type queuing output default-out-policy
!
interface xe14
 service-policy type queuing output default-out-policy
!
interface xe15
 service-policy type queuing output default-out-policy
!
interface xe16
 service-policy type queuing output default-out-policy
!
interface xe17
 service-policy type queuing output default-out-policy
!
interface xe18
 service-policy type queuing output default-out-policy
!
interface xe19
 service-policy type queuing output default-out-policy
!
interface xe20
 service-policy type queuing output default-out-policy
!
interface xe21
 service-policy type queuing output default-out-policy
!
interface xe22
 service-policy type queuing output default-out-policy
!
interface xe23
```

© 2026 IP Infusion Inc. Proprietary

```
service-policy type queuing output default-out-policy
!
interface xe24
 service-policy type queuing output default-out-policy
!
interface xe25
 service-policy type queuing output default-out-policy
!
interface xe26
 service-policy type queuing output default-out-policy
!
interface xe27
 service-policy type queuing output default-out-policy
!
interface xe28
 service-policy type queuing output default-out-policy
!
interface xe29
 service-policy type queuing output default-out-policy
!
interface xe30
 service-policy type queuing output default-out-policy
!
interface xe31
 service-policy type queuing output default-out-policy
!
interface xe32
 service-policy type queuing output default-out-policy
!
interface xe33
 service-policy type queuing output default-out-policy
!
interface xe34
 service-policy type queuing output default-out-policy
!
interface xe35
 service-policy type queuing output default-out-policy
!
interface xe36
 service-policy type queuing output default-out-policy
!
interface xe37
 service-policy type queuing output default-out-policy
!
interface xe38
 service-policy type queuing output default-out-policy
!
interface xe39
 service-policy type queuing output default-out-policy
!
interface xe40
 service-policy type queuing output default-out-policy
!
interface xe41
 service-policy type queuing output default-out-policy
!
interface xe42
```

```
 service-policy type queuing output default-out-policy
!
interface xe43
 service-policy type queuing output default-out-policy
!
interface xe44
 service-policy type queuing output default-out-policy
!
interface xe45
 service-policy type queuing output default-out-policy
!
interface xe46
service-policy type queuing output default-out-policy
!
interface xe47
 service-policy type queuing output default-out-policy
!
interface xe48
 service-policy type queuing output default-out-policy
!
```

storm-control

Use this command to set the rising threshold level for broadcast, multicast, or destination lookup failure traffic. The storm control action occurs when traffic utilization reaches this level.

Storm control is used to block the forwarding of unnecessary flooded traffic. A packet storm occurs when a large number of broadcast packets are received on a port. Forwarding these packets can cause the network to slow down or time out.

Use the `no` form of this command to disable storm control.

Note:

- Granularity of storm-control is around 18 kbps for lower values and it increases when configured with increase in rate. Deviation is expected +-7/8 % of rate configured for lower value rate changes.
- If the burst-size limit is set too low, too many packets will be subjected to rate limiting, and the desired rate limit will not be achieved. The burst size needs to be large enough to accommodate the maximum transmission unit (MTU) size of the packets.
- The `storm-control dlf` command applies rate limiting to all Broadcast, Unknown Unicast, and Multicast (BUM) traffic, not just the unknown unicast packets that typically trigger DLF.

Command Syntax

```
storm-control (errdisable|((broadcast|multicast|dlf) (level LEVEL | <0-4294967294>
  (kbps|mbps|gbps) burst-size <1-33292>)))

no storm-control (errdisable|broadcast|multicast|dlf)
```

Parameters

`errdisable`	Error disable interface on storm control
`broadcast`	Broadcast rate limiting.
`multicast`	Multicast rate limiting.
`dlf`	Destination lookup failure (DLF) limiting.
`level`	Sets the percentage of the threshold.
`LEVEL`	The percentage of the threshold; percentage of the maximum speed (pps) of the interface <0.0000-99.0000>.
`<0-4294967294>`	Sets absolute threshold value <0-4294967294>
`kbps`	Specifies the units of Kilobits per second.
`mbps`	Specifies the units of Megabits per second.
`gbps`	Specifies the units of Gigabits per second.
`burst-size`	Set burst size
`<32-128000>`	Value of burst in kbps

Default

Disabled

Command Mode

Interface mode

Applicability

This command was introduced in OcNOS version 3.0.

Examples

```
#configure terminal
(config)#interface xe1
(config-if)#storm-control broadcast level 30

(config)#interface xe2
(config-if)#storm-control multicast level 30

(config)#interface xe3
(config-if)#storm-control multicast 300 mbps

(config)#interface xe4
(config-if)#no storm-control multicast
```

trust dscp

Use this command to classify the traffic based on DSCP map on L2 port.

Use the `no` form of the command to remove the configuration.

Note: By default, the trust of L2 ports is CoS. If the user wants to map the traffic according to the DSCP value, `trust dscp` can be set on ports to achieve the requirement.

Tagged packet color is based on the DEI bit. Untagged packet color is based on DSCP value.

Out-DSCP option in dscp-to-queue profile is not applicable on L2 interfaces when trust DSCP is set

Only “default” dscp-to-queue profile is valid for trust DSCP. User-defined `dscp-to-queue` cannot be attached on L2 interfaces.

Command Syntax

```
trust dscp
no trust dscp
```

Parameters

`trust`	Configure port trust state
`dscp`	Classifies ingress packets with the packet DSCP values

Default

No default value is specified

Command Mode

Interface mode

Applicability

This command was introduced in OcNOS version 3.0.

Examples

```
(config)#interface xe1
(config-if)#trust dscp
```

© 2026 IP Infusion Inc. Proprietary

wfq-queue weight

Use this command to set WFQ-queue weight for a queue.

Use the `no` form of the command to un-set configured WFQ configuration.

Command Syntax

```
wfq-queue weight <1-127>
no wfq-queue weights
```

Parameters

`<1-127>`	WFQ queue weight to be configured.

Default

No default value is specified

Command Mode

Policy-class-map queuing Mode

Applicability

This command was introduced in OcNOS version 3.0.

Examples

```
(config)#policy-map type queuing default default-out-policy
(config-pmap-que-def)# class type queuing default q0
(config-pmap-c-que-def)#wfq-queue weight 2
```

© 2026 IP Infusion Inc. Proprietary

vc-qos map-profile

Use this command is used to binding PCP to TC mapping profile to attachment circuits.

Use the `no` form of this command to remove the profile.

Note: Profile configured on the VPLS service is effective for all attachment circuits associated with VPLS service. However, if a profile is bound to attachment circuit, that profile takes higher priority for that attachment circuit.

Command Syntax

```
vc-qos map-profile (cos-to-queue | queue-color-to-cos) NAME
no vc-qos map-profile (cos-to-queue | queue-color-to-cos)  NAME
```

Parameters

`cos-to-queue`	Profile for CoS to queue map
`queue-color-to-cos`	Profile for queue-color to CoS map
`NAME`	Profile map name (maximum 32 characters)

Default

By-default, “default” global cos-to-queue map is applied on an attachment-circuit if no user-defined cos-to-queue profile is applied on interface or VPLS-service.

Traffic received on VPLS service will be affected by QoS treatment by configurations in the following order:

1. PCP to TC/TC to PCP profile configuration bound to attachment circuit.
2. PCP to TC/TC to PCP profile configuration bound to VPLS service.
3. PCP to TC/TC to PCP profile configuration bound to ingress port.
4. Global PCP to TC/TC to PCP profile configuration.

Command Mode

interface-VPLS modes

Applicability

This command was introduced in OcNOS version 3.0.

Example

```
(config)#interface xe1
(config-if)#switchport
(config-if)#mpls-vpls vpls1 service-template st1
(config-if-vpls)#vc-qos map-profile queue-color-to-cos qc-profile-2
```

© 2026 IP Infusion Inc. Proprietary

vpls-qos map-profile

Use this command is used to binding PCP to TC mapping profile to VPLS service.

Use the `no` form of this command to remove the profile.

Command Syntax

```
vpls-qos map-profile (cos-to-queue | queue-color-to-cos) NAME
no vpls-qos map-profile (cos-to-queue | queue-color-to-cos) NAME
```

Parameters

`cos-to-queue`	Profile for CoS to queue map
`queue-color-to-cos`	Profile for queue-color to CoS map
`NAME`	Profile map name (maximum 32 characters)

Default

By-default, “default” global `cos-to-queue` map is applied on a VPLS service if no user-defined `cos-to-queue` profile is applied on interface.

Traffic received on a VPLS service will be affected by QoS treatment by configurations in the following order:

1. PCP to TC profile configuration bound to VPLS service.
2. PCP to TC profile configuration bound to ingress port.
3. Global PCP to TC profile configuration.

Command Mode

MPLS-VPLS mode

Applicability

This command was introduced in OcNOS version 3.0.

Example

```
(config)#mpls vpls vpls1 1
(config-vpls)#vpls-qos map-profile cos-to-queue CQ-PROFILE-1
(config-vpls)#exit
```

© 2026 IP Infusion Inc. Proprietary

Index

© 2026 IP Infusion Inc. Proprietary

S

T

V

W