Cisco Nexus 3548 Switch NX-OS Quality of Service Configuration Guide, Release 5.0(3)A1(1)

First Published: November 06, 2012

Last Modified: November 06, 2012

Americas Headquarters
Cisco Systems, Inc.
170 West Tasman Drive
San Jose, CA 95134-1706
USA
http://www.cisco.com
Tel: 408 526-4000
800 553-NETS (6387)
Fax: 408 527-0883

Text Part Number: OL-27857-01

CONTENTS

Preface

This preface contains the following sections:

Audience

This publication is for experienced network administrators who configure and maintain Cisco Nexus devices .

Document Conventions

Command descriptions use the following conventions:

Convention	Description
bold	Bold text indicates the commands and keywords that you enter literally as shown.
Italic	Italic text indicates arguments for which the user supplies the values.
[x]	Square brackets enclose an optional element(keyword or argument).
[x | y]	Square brackets enclosing keywords or arguments separated by a vertical bar indicate an optional choice.
{x | y}	Braces enclosing keywords or arguments separated by a vertical bar indicate a required choice.

Convention	Description
[x {y | z}]	Nested set of square brackets or braces indicate optional or required choices within optional or required elements. Braces and a vertical bar within square brackets indicate a required choice within an optional element.
`variable`	Indicates a variable for which you supply values, in context where italics cannot be used.
string	A nonquoted set of characters. Do not use quotation marks around the string or the string will include the quotation marks.

Examples use the following conventions:

Convention	Description
`screen font`	Terminal sessions and information the switch displays are in screen font.
`boldface screen font`	Information you must enter is in boldface screen font.
italic screen font	Arguments for which you supply values are in italic screen font.
< >	Nonprinting characters, such as passwords, are in angle brackets.
[ ]	Default responses to system prompts are in square brackets.
!, #	An exclamation point (!) or a pound sign (#) at the beginning of a line of code indicates a comment line.

This document uses the following conventions:

Note Means *reader take note*. Notes contain helpful suggestions or references to material not covered in the manual.

Caution Means *reader be careful*. In this situation, you might do something that could result in equipment damage or loss of data.

Related Documentation for Nexus 3548 Switch NX-OS Software

The Cisco Nexus 3548 Switch NX-OS documentation set is available at the following URL:

http://www.cisco.com/en/US/products/ps11541/tsd_products_support_series_home.html

Release Notes

The release notes are available at the following URL:

http://cisco.com/en/US/products/ps11541/prod_release_notes_list.html

Installation and Upgrade Guides

The installation and upgrade guides are available at the following URL:

http://cisco.com/en/US/products/ps11541/prod_installation_guides_list.html

License Information

For information about feature licenses in NX-OS, see the *Cisco NX-OS Licensing Guide*, available at the following URL: http://www.cisco.com/en/US/docs/switches/datacenter/sw/nx-os/licensing/guide/b_Cisco_NX-OS_Licensing_Guide.html.

For the NX-OS end user agreement and copyright information, see *License and Copyright Information for Cisco NX-OS Software*, available at the following URL: http://www.cisco.com/en/US/docs/switches/datacenter/sw/4_0/nx-os/license_agreement/nx-ossw_lisns.html.

Configuration Guides

The configuration guides are available at the following URL:

http://cisco.com/en/US/products/ps11541/products_installation_and_configuration_guides_list.html

Command References

The command references are available at the following URL:

http://cisco.com/en/US/products/ps11541/prod_command_reference_list.html

Error and System Messages

The error and system message reference guides are available at the following URL:

http://cisco.com/en/US/products/ps11541/products_system_message_guides_list.html

Documentation Feedback

To provide technical feedback on this document, or to report an error or omission, please send your comments to nexus3k-docfeedback@cisco.com . We appreciate your feedback.

CHAPTER 1

Overview

This chapter contains the following sections:

Quality of Service Overview

This document describes the configurable Cisco NX-OS Quality of Service (QoS) features. You use the QoS features to provide the most desirable flow of traffic through a network. QoS allows you to classify the network traffic, prioritize the traffic flow, and provide congestion avoidance. The control of traffic is based on the fields in the packets that flow through the system. You use the Modular QoS CLI (MQC) to create the traffic classes and policies of the QoS features.

QoS features are applied using QoS policies and queuing policies, as follows:

- QoS policies include classification and marking features.
- Queuing policies use the queuing and scheduling features.
- Network QoS policies include configuring maximum transmission unit (MTU).

CHAPTER 2

Configuring QoS

This chapter contains the following sections:

Information About Quality of Service

The configurable Cisco NX-OS quality of service (QoS) features allow you to classify the network traffic, prioritize the traffic flow, and provide congestion avoidance.

The default QoS configuration on the device provides best-effort service for Ethernet traffic. QoS can be configured to provide additional classes of service for Ethernet traffic. Cisco NX-OS QoS features are configured using Cisco Modular QoS CLI (MQC).

Note In the event of congestion or collisions, Ethernet will drop packets. The higher level protocols detect the missing data and retransmit the dropped packets.

Modular QoS CLI

The Cisco Modular QoS CLI (MQC) provides a standard set of commands for configuring QoS.

You can use MQC to define additional traffic classes and to configure QoS policies for the whole system and for individual interfaces. Configuring a QoS policy with MQC consists of the following steps:

1. Define traffic classes.
2. Associate policies and actions with each traffic class.

3 Attach policies to logical or physical interfaces as well as at the global system level.

MQC provides two command types to define traffic classes and policies:

class-map

Defines a class map that represents a class of traffic based on packet-matching criteria. Class maps are referenced in policy maps.

The class map classifies incoming packets based on matching criteria, such as the IEEE 802.1p class of service (CoS) value. Unicast and multicast packets are classified.

policy-map

Defines a policy map that represents a set of policies to be applied on a class-by-class basis to class maps.

The policy map defines a set of actions to take on the associated traffic class, such as limiting the bandwidth or dropping packets.

You define the following class-map and policy-map object types when you create them:

network-qos

Defines MQC objects that you can use for system level related actions.

qos

Defines MQC objects that you can use for classification.

queuing

Defines MQC objects that you can use for queuing and scheduling.

Note The qos type is the default for the **class-map** and **policy-map** commands, but not for the **service-policy** which requires that you specify an explicit type.

You can attach policies to interfaces or EtherChannels as well as at the global system level by using the **service-policy** command.

You can view all or individual values for MQC objects by using the **show class-map** and **show policy-map** commands.

An MQC target is an entity (such as an Ethernet interface) that represents a flow of packets. A service policy associates a policy map with an MQC target and specifies whether to apply the policy on incoming or outgoing packets. This mapping enables the configuration of QoS policies such as marking, bandwidth allocation, buffer allocation, and so on.

System Classes

The system qos is a type of MQC target. You use a service policy to associate a policy map with the system qos target. A system qos policy applies to all interfaces on the switch unless a specific interface has an overriding service-policy configuration. The system qos policies are used to define system classes, the classes of traffic across the entire switch, and their attributes.

If service policies are configured at the interface level, the interface-level policy always takes precedence over system class configuration or defaults.

On the Cisco Nexus device, a system class is uniquely identified by a qos-group value. A total of eight system classes are supported. The device supports one default class which is always present on the switch. Up to seven additional system classes can be created by the administrator.

Default System Classes

The device provides the following system classes:

- Drop system class

  By default, the software classifies all unicast and multicast Ethernet traffic into the default drop system class. This class is identified by qos-group 0.

  This class is created automatically when the system starts up (the class is named **class-default** in the CLI). You cannot delete this class and you cannot change the match criteria associated with the default class.

Information About Policy Types

The device supports a number of policy types. You create class maps in the policy types.

There are three policy types

- Network-qos
- Queuing
- QoS

The following QoS parameters can be specified for each type of class:

- Type network-qos—A network-qos policy is used to instantiate system classes and associate parameters with those classes that are of system-wide scope.
  - Classification—The traffic that matches this class are as follows:
    - QoS Group—A class map of type network-qos identifies a system class and is matched by its associated qos-group.
  - Policy—The actions that are performed on the matching traffic are as follows:

    Note A network-qos policy can only be attached to the system QoS target.

  - MTU—The MTU that needs to be enforced for the traffic that is mapped to a system class.

    Note The Cisco Nexus device supports one MTU for all classes for all ports.

- Set CoS value—This configuration is used to mark 802.1p values for all traffic mapped to this system class.
- Congestion Control DCTCP and ECN—Data Center TCP (DCTCP) is an enhancement to the TCP congestion control algorithm for data center networks. It leverages Explicit Congestion Notification (ECN) feature, to mark all the packets when the queue length exceeds a configured DCTCP threshold value. The routers and end hosts use this marking as a signal that the network is congested to slow down sending packets. To enable an DCTCP/ECN, use the "congestion-control dctcp ecn" command in the network-qos policy map mode.

Note Enabling DCTCP and ECN on a class on a network-qos policy implies that DCTCP and ECN is enabled for all ports in the system.

- Type queuing—A type queuing policy is used to define the scheduling characteristics of the queues associated with system classes.

  The Cisco Nexus device supports type queuing in the egress direction.

Note Some configuration parameters when applied to an EtherChannel are not reflected on the configuration of the member ports.

  - Classification—The traffic that matches this class are as follows:
    - QoS Group—A class map of type queuing identifies a system class and is matched by its associated QoS group.
  - Policy—The actions that are performed on the matching traffic are as follows:

Note These policies can be attached to the system qos target or to any interface. The output queuing policy is used to configure output queues on the device associated with system classes.

    - Bandwidth—Sets the guaranteed scheduling deficit weighted round robin (DWRR) percentage for the system class.
    - Priority—Sets a system class for strict-priority scheduling. Only one system class can be configured for priority in a given queuing policy.

- Type qos—A type QoS policy is used to classify traffic that is based on various Layer 2, Layer 3, and Layer 4 fields in the frame and to map it to system classes.

Note Some configuration parameters when applied to an EtherChannel are not reflected on the configuration of the member ports.

- Classification—The traffic that matches this class are as follows:
  - Access Control Lists—Classifies traffic based on the criteria in existing ACLs.
  - Class of Service—Matches traffic based on the CoS field in the frame header.
  - DSCP—Classifies traffic based on the Differentiated Services Code Point (DSCP) value in the DiffServ field of the IP header.
  - IP Real Time Protocol—Classifies traffic on the port numbers used by real-time applications.
  - Precedence—Classifies traffic based on the precedence value in the type of service (ToS) field of the IP header.
- Policy—The actions that are performed on the matching traffic are as follows:

Note This policy can be attached to the system or to any interface. It applies to input traffic only.

  - QoS Group—Sets the QoS group that corresponds to the system class this traffic flow is mapped to.
    - Cisco Nexus device supports the following:
      - Five QoS groups
      - Five queues for unicast
      - Five queues for multicast

MTU

TheCisco Nexus device supports one MTU for all classes for all ports.

When configuring MTU, follow these guidelines:

- For the Cisco Nexus device, the MTU is controlled by the value configured on the class default. The same MTU must be configured on all classes.
- Enter the **system jumbomtu** command to define the upper bound of any MTU in the system. The system jumbo MTU has a default value of 9216 bytes. The minimum MTU is 1500 bytes and the maximum MTU is 9216 bytes.
- The system class MTU sets the MTU for all packets in the class. The system class MTU cannot be configured larger than the global jumbo MTU.
- The default system class has a default MTU of 1500 bytes. You can configure this value.
- You can specify the MTU value for either a single Layer 3 interface or a range of Layer 3 interfaces. When you change the Layer 3 interface MTU value to the jumbo MTU value (1500 bytes or greater), you must also change the network QoS MTU value to 1500 bytes or greater. The device generates a syslog message to inform you of this requirement.

Trust Boundaries

The trust boundary is enforced by the incoming interface as follows:

- By default, all Ethernet interfaces are trusted interfaces.The 802.1p CoS and DSCP are preserved unless the marking is configured. There is no default CoS to queue and DSCP to queue mapping. You can define and apply a policy to create these mappings. By default, without a user defined policy, all traffic is assigned to the default queue.
- Any packet that is not tagged with an 802.1p CoS value is classified into the default drop system class. If the untagged packet is sent over a trunk, it is tagged with the default untagged CoS value, which is zero.
- You can override the default untagged CoS value for an Ethernet interface or port channel.
- You can override the default untagged CoS value for an Ethernet interface or a port channel interface using the **untagged cos** *cos-value* command.
- You can override the default untagged Cos value for an Ethernet or a Layer 3 interface or a port channel interface using the **untagged cos** *cos-value* command.

After the system applies the untagged CoS value, QoS functions the same as for a packet that entered the system tagged with the CoS value.

Ingress Classification Policies

You use classification to partition traffic into classes. You classify the traffic based on the port characteristics (CoS field) or the packet header fields that include IP precedence, Differentiated Services Code Point (DSCP), and Layer 2 to Layer 4 parameters. The values used to classify traffic are called match criteria. When you define a traffic class, you can specify multiple match criteria or you can determine the traffic class by matching any or all criteria.

Traffic that fails to match any class is assigned to a default class of traffic called class-default.

Egress Queuing Policies

You can associate an egress policy map with an Ethernet interface to guarantee the bandwidth for the specified traffic class or to configure the egress queues.

The bandwidth allocation limit applies to all traffic on the interface.

Each Ethernet interface supports up to five queues, one for each system class. The queues have the following default configuration:

- In addition to these queues, control traffic that is destined for the CPU uses strict priority queues. These queues are not accessible for user configuration.
- Standard Ethernet traffic in the default drop system class is assigned a queue. This queue uses WRR scheduling with 100 percent of the bandwidth.

If you add a system class, a queue is assigned to the class. You must reconfigure the bandwidth allocation on all affected interfaces. Bandwidth is not dedicated automatically to user-defined system classes.

You can configure a strict priority queue. This queue is serviced before all other queues except the control traffic queue (which carries control rather than data traffic).

QoS for Traffic Directed to the CPU

The device automatically applies QoS policies to traffic that is directed to the CPU to ensure that the CPU is not flooded with packets. Control traffic, such as bridge protocol data units (BPDU) frames, is given higher priority to ensure delivery.

QoS Configuration Guidelines and Limitations

To maintain optimal switch performance, follow these guidelines when configuring system classes and policies:

- Switch resources (such as buffers, virtual output queues, and egress queues) are partitioned based on the default and user-defined system classes. Cisco NX-OS automatically adjusts the resource allocation to accommodate the configured system classes.
- When configuring Port Channels, the service policy configured on an Port Channel applies to all member interfaces.
- By default, queues 6 and 7 are reserved for control plane traffic and queue 5 for SPAN traffic. So you can configure four classes along with the default class.

Configuring System Classes

Configuring Class Maps

You can create or modify a class map with the **class-map** command. The class map is a named object that represents a class of traffic. In the class map, you specify a set of match criteria for classifying the packets. You can then reference class maps in policy maps.

Note The class map type default is type qos and its match criteria default is match-all.

Procedure

	Command or Action	Purpose
Step 1	switch# **configure terminal**	Enters global configuration mode.
Step 2	switch(config)# **class-map** [**type** {**network-qos** | **qos** | **queuing**}] *class-map name*	Creates or accesses a named object that represents the specified class of traffic. Class-map names can contain alphabetic, hyphen, or underscore characters, are case sensitive, and can be up to 40 characters. The three class-map configuration modes are as follows:

	Command or Action	Purpose
		• **network-qos**—Network-wide (global) mode. CLI prompt: switch(config-cmap-nq)# • **qos**—Classification mode; this is the default mode. CLI prompt: switch(config-cmap-qos)# • **queuing**—Queuing mode. CLI prompt: switch(config-cmap-que)#
Step 3	switch(config)# **class-map** [**type qos**] [**match-all** | **match-any**] *class-map name*	(Optional) Specifies that packets must match any or all criteria that is defined for a class map. • **match-all**—Classifies traffic if packets match all criteria that is defined for a specified class map (for example, if both the defined CoS and the ACL criteria match). • **match-any**—Classifies traffic if packets match any criteria that is defined for a specified class map (for example, if either the CoS or the ACL criteria matches). Class-map names can contain alphabetic, hyphen, or underscore characters, are case sensitive, and can be up to 40 characters.
Step 4	switch(config)# **no class-map** [**type** {**network-qos** | **qos** | **queuing**}] *class-name*	(Optional) Deletes the specified class map. **Note** You cannot delete the system-defined class map: class-default. Class-map names can contain alphabetic, hyphen, or underscore characters, are case sensitive, and can be up to 40 characters.

Configuring ACL Classification

You can classify traffic by matching packets based on an existing access control list (ACL). Traffic is classified by the criteria defined in the ACL. The **permit** and **deny** ACL keywords are ignored in the matching; even if a match criteria in the access-list has a **deny** action, it is still used for matching for this class.

Procedure

	Command or Action	Purpose
Step 1	switch# **configure terminal**	Enters configuration mode.
Step 2	switch(config)# **class-map type qos** *class-name*	Creates a named object that represents a class of traffic. Class-map names can contain alphabetic, hyphen, or underscore characters, are case sensitive, and can be up to 40 characters.

	Command or Action	Purpose
Step 3	switch(config-cmap-qos)# **match access-group name** *acl-name*	Configures a traffic class by matching packets based on the *acl-name*. The **permit** and **deny** ACL keywords are ignored in the matching. **Note** You can only define a single ACL in a class map. You cannot add any other match criteria to a class with a **match access-group** defined.
Step 4	switch(config-cmap-qos)# **no match access-group name** *acl-name*	(Optional) Removes the match from the traffic class.

This example shows how to classify traffic by matching packets based on existing ACLs:

```
switch# configure terminal
switch(config)# class-map type qos class_acl
switch(config-cmap-qos)# match access-group name acl-01
```

Use the **show class-map** command to display the ACL class-map configuration:

```
switch# show class-map class_acl
```

Configuring CoS Classification

You can classify traffic based on the class of service (CoS) in the IEEE 802.1Q header. This 3-bit field is defined in IEEE 802.1p to support QoS traffic classes. CoS is encoded in the high order 3 bits of the VLAN ID Tag field and is referred to as *user_priority*.

Procedure

	Command or Action	Purpose
Step 1	switch# **configure terminal**	Enters configuration mode.
Step 2	switch(config)# **class-map type qos** *class-name*	Creates a named object that represents a class of traffic. Class-map names can contain alphabetic, hyphen, or underscore characters, are case sensitive, and can be up to 40 characters.
Step 3	switch(config-cmap-qos)# **match cos** *cos-value*	Specifies the CoS value to match for classifying packets into this class. You can configure a CoS value in the range of 0 to 7.
Step 4	switch(config-cmap-qos)# **no match cos** *cos-value*	(Optional) Removes the match from the traffic class.

This example shows how to classify traffic by matching packets based on a defined CoS value:

```
switch# configure terminal
switch(config)# class-map type qos match-any class_cos
```

```
switch(config-cmap-qos)# match cos 4, 5-6
```

Use the **show class-map** command to display the CoS value class-map configuration:

```
switch# show class-map class_cos
```

Configuring DSCP Classification

You can classify traffic based on the Differentiated Services Code Point (DSCP) value in the DiffServ field of the IP header.

Table 1: Standard DSCP Values

Value	List of DSCP Values
af11	AF11 dscp (001010)—decimal value 10
af12	AF12 dscp (001100)—decimal value 12
af13	AF13 dscp (001110)—decimal value 14
af21	AF21 dscp (010010)—decimal value 18
af22	AF22 dscp (010100)—decimal value 20
af23	AF23 dscp (010110)—decimal value 22
af31	AF31 dscp (011010)—decimal value 26
af32	AF32 dscp (011100)—decimal value 28
af33	AF33 dscp (011110)—decimal value 30
af41	AF41 dscp (100010)—decimal value 34
af42	AF42 dscp (100100)—decimal value 36
af43	AF43 dscp (100110)—decimal value 38
cs1	CS1 (precedence 1) dscp (001000)—decimal value 8
cs2	CS2 (precedence 2) dscp (010000)—decimal value 16
cs3	CS3 (precedence 3) dscp (011000)—decimal value 24
cs4	CS4 (precedence 4) dscp (100000)—decimal value 32

Value	List of DSCP Values
cs5	CS5 (precedence 5) dscp (101000)—decimal value 40
cs6	CS6 (precedence 6) dscp (110000)—decimal value 48
cs7	CS7 (precedence 7) dscp (111000)—decimal value 56
default	Default dscp (000000)—decimal value 0
ef	EF dscp (101110)—decimal value 46

Procedure

	Command or Action	Purpose
Step 1	switch# **configure terminal**	Enters global configuration mode.
Step 2	switch(config)# **class-map type qos** *class-name*	Creates a named object that represents a class of traffic. Class-map names can contain alphabetic, hyphen, or underscore characters, are case sensitive, and can be up to 40 characters.
Step 3	switch(config-cmap-qos)# **match dscp** *dscp-list*	Configures the traffic class by matching packets based on the values in the *dscp-list* variable. For a list of DSCP values, see the Standard DSCP Values table.
Step 4	switch(config-cmap-qos)# **no match dscp** *dscp-list*	(Optional) Removes the match from the traffic class. For a list of DSCP values, see the Standard DSCP Values table.

This example shows how to classify traffic by matching packets based on the DSCP value in the DiffServ field of the IP header:

```
switch# configure terminal
switch(config)# class-map type qos match-any class_dscp
switch(config-cmap-qos)# match  dscp af21, af32
```

Use the **show class-map** command to display the DSCP class-map configuration:

```
switch# show class-map class_dscp
```

Configuring IP RTP Classification

The IP Real-time Transport Protocol (RTP) is a transport protocol for real-time applications that transmits data such as audio or video and is defined by RFC 3550. Although RTP does not use a common TCP or UDP port, you typically configure RTP to use ports 16384 to 32767. UDP communications use an even port and the next higher odd port is used for RTP Control Protocol (RTCP) communications.

You can classify based on UDP port ranges, which are likely to target applications using RTP.

Procedure

	Command or Action	Purpose
Step 1	switch# **configure terminal**	Enters configuration mode.
Step 2	switch(config)# **class-map type qos** *class-name*	Creates a named object that represents a class of traffic. Class-map names can contain alphabetic, hyphen, or underscore characters, are case sensitive, and can be up to 40 characters.
Step 3	switch(config-cmap-qos)# **match ip rtp** *port-number*	Configures the traffic class by matching packets based on a range of lower and upper UDP port numbers, which is likely to target applications using RTP. Values can range from 2000 to 65535.
Step 4	switch(config-cmap-qos)# **no match ip rtp** *port-number*	(Optional) Removes the match from the traffic class.

This example shows how to classify traffic by matching packets based on UDP port ranges that are typically used by RTP applications:

```
switch# configure terminal
switch(config)# class-map type qos match-any class_rtp
switch(config-cmap-qos)# match  ip rtp 2000-2100, 4000-4100
```

Use the **show class-map** command to display the RTP class-map configuration:

```
switch# show class-map class_rtp
```

Configuring Precedence Classification

You can classify traffic based on the precedence value in the type of service (ToS) byte field of the IP header. The following table shows the precedence values:

Table 2: Precedence Values

Value	List of Precedence Values
<0-7>	IP precedence value
critical	Critical precedence (5)
flash	Flash precedence (3)
flash-override	Flash override precedence (4)
immediate	Immediate precedence (2)
internet	Internetwork control precedence (6)

Value	List of Precedence Values
network	Network control precedence (7)
priority	Priority precedence (1)
routine	Routine precedence (0)

Procedure

	Command or Action	Purpose
Step 1	switch# **configure terminal**	Enters global configuration mode.
Step 2	switch(config)# **class-map type qos match-any** *class-name*	Creates a named object that represents a class of traffic. Class-map names can contain alphabetic, hyphen, or underscore characters, are case sensitive, and can be up to 40 characters.
Step 3	switch(config-cmap-qos)#**match precedence** *precedence-values*	Configures the traffic class by matching packets based on precedence values. For a list of precedence values, see the Precedence Values table.
Step 4	switch((config-cmap-qos)# **no match precedence** *precedence-values*	(Optional) Removes the match from the traffic class. For a list of precedence values, see the Precedence Values table.

This example shows how to classify traffic by matching packets based on the precedence value in the ToS byte field of the IP header:

```
switch# configure terminal
switch(config)# class-map type qos match-any class_precedence
switch(config-cmap-qos)# match precedence 1-2, critical
```

Use the **show class-map** command to display the IP precedence value class-map configuration:

```
switch# show class-map class_precedence
```

Creating Policy Maps

The **policy-map** command is used to create a named object that represents a set of policies that are to be applied to a set of traffic classes.

The device provides one default system class: a drop class for best-effort service (class-default). You can define up to four additional system classes for Ethernet traffic.

The following predefined policy maps are used as default service policies:

- network-qos: default-nq-policy
- Input qos: default-in-policy
- Output queuing: default-out-policy

You need to create a policy map to specify the policies for any user-defined class. In the policy map, you can configure the QoS parameters for each class. You can use the same policy map to modify the configuration of the default classes.

The device distributes all the policy-map configuration values to the attached network adapters.

Before You Begin

Before creating the policy map, define a class map for each new system class.

Procedure

	Command or Action	Purpose
Step 1	switch# **configure terminal**	Enters configuration mode.
Step 2	switch(config)# **policy-map** [**type** {**network-qos** | **qos** | **queuing**}] *policy-name*	Creates a named object representing a set of policies that are to be applied to a set of traffic classes. Policy-map names can contain alphabetic, hyphen, or underscore characters, are case sensitive, and can be up to 40 characters. The three policy-map configuration modes are as follows: • network-qos—Network-wide (global) mode. CLI prompt: switch(config-pmap-nq)# • qos—Classification mode; this is the default mode. CLI prompt: switch(config-pmap-qos)# • queuing—Queuing mode. CLI prompt: switch(config-pmap-que)#
Step 3	switch(config)# **no policy-map** [**type** {**network-qos** | **qos** | **queuing**}] *policy-name*	(Optional) Deletes the specified policy map.
Step 4	switch(config-pmap)# **class** [**type** {**network-qos** | **qos** | **queuing**}] *class-name*	Associates a class map with the policy map, and enters configuration mode for the specified system class. The three class-map configuration modes are as follows: • network-qos—Network-wide (global) mode. CLI prompt: switch(config-pmap-c-nq)# • qos—Classification mode; this is the default mode. CLI prompt: switch(config-pmap-c-qos)# • queuing—Queuing mode. CLI prompt: switch(config-pmap-c-que)# **Note** The associated class map must be the same type as the policy-map type.
Step 5	switch(config-pmap)# **no class** [**type** {**network-qos** | **qos** | **queuing**}] *class-name*	(Optional) Deletes the class map association.

Configuring Type QoS Policies

Type qos policies are used for classifying the traffic of a specific system class identified by a unique qos-group value. A type qos policy can be attached to the system or to individual interfaces for ingress traffic only.

You can set a maximum of five QoS groups for ingress traffic.

Procedure

	Command or Action	Purpose
Step 1	switch# **configure terminal**	Enters global configuration mode.
Step 2	switch(config)# **policy-map type qos** *policy-name*	Creates a named object that represents a set of policies that are to be applied to a set of traffic classes. Policy-map names can contain alphabetic, hyphen, or underscore characters, are case sensitive, and can be up to 40 characters.
Step 3	switch(config-pmap-qos)# [**class** | **class-default**] **type qos** *class-name*	Associates a class map with the policy map, and enters configuration mode for the specified system class. **Note** The associated class map must be the same type as the policy map type.
Step 4	switch(config-pmap-c-qos)# **set qos-group** *qos-group-value*	Configures one or more **qos-group** values to match on for classification of traffic into this class map. The list below identifies the ranges of the *qos-group-value* . There is no default value. **Note** The switch can only support a maximum of five QoS groups within this range.
Step 5	switch(config-pmap-c-qos)# **no set qos-group** *qos-group-value*	(Optional) Removes the **qos-group** values from this class.

This example shows how to define a type qos policy map:

```
switch# configure terminal
switch(config)# policy-map type qos policy-s1
switch(config-pmap-qos)# class type qos class-s1
switch(config-pmap-c-qos)# set qos-group 2
```

Configuring Type Network QoS Policies

Type network qos policies can only be configured on the system qos attachment point. They are applied to the entire switch for a particular class.

Procedure

	Command or Action	Purpose
Step 1	switch# **configure terminal**	Enters configuration mode.

	Command or Action	Purpose
Step 2	switch(config)# **policy-map type network-qos** *policy-name*	Creates a named object that represents a set of policies that are to be applied to a set of traffic classes. Policy-map names can contain alphabetic, hyphen, or underscore characters, are case sensitive, and can be up to 40 characters.
Step 3	switch(config-pmap-nq)# **class type network-qos** *class-name*	Associates a class map with the policy map, and enters configuration mode for the specified system class. **Note** The associated class map must be the same type as the policy map type.
Step 4	switch(config-pmap-c-nq)# **mtu** *mtu-value*	Specifies the MTU value in bytes. **Note** The *mtu-value* that you configure must be less than the value set by the **system jumbomtu** command.
Step 5	switch(config-pmap-c-nq)# **no mtu**	(Optional) Resets the MTU value in this class.
Step 6	switch(config-pmap-c-nq)# **set cos** *cos-value*	Specifies a 802.1Q CoS value which is used to mark packets on this interface. The value range is from 0 to 7.
Step 7	switch(config-pmap-c-nq)# **no set cos** *cos-value*	(Optional) Disables the marking operation in this class.

This example shows how to define a type network-qos policy map:

```
switch# configure terminal
switch(config)# policy-map type network-qos policy-que1
switch(config-pmap-nq)# class type network-qos class-que1
switch(config-pmap-c-nq)# mtu 5000
switch(config-pmap-c-nq)# set cos 4
```

Configuring Type Queuing Policies

Type queuing policies are used for scheduling and buffering the traffic of a specific system class. A type queuing policy is identified by its QoS group and can be attached to the system or to individual interfaces for input or output traffic.

Procedure

	Command or Action	Purpose
Step 1	switch# **configure terminal**	Enters global configuration mode.
Step 2	switch(config)# **policy-map type queuing** *policy-name*	Creates a named object that represents a set of policies that are to be applied to a set of traffic classes. Policy-map names can contain alphabetic, hyphen, or underscore characters, are case sensitive, and can be up to 40 characters.

	Command or Action	Purpose
Step 3	switch(config-pmap-que)# **class type queuing** *class-name*	Associates a class map with the policy map, and enters configuration mode for the specified system class.
Step 4	switch(config-pmap-c-que)# **bandwidth percent** *percentage*	Specifies the guaranteed percentage of interface bandwidth allocated to this class. By default, no bandwidth is specified for a class. **Note** Before you can successfully allocate bandwidth to the class, you must first reduce the default bandwidth configuration on class-default and class-fcoe.
Step 5	switch(config-pmap-c-que)# **no bandwidth percent** *percentage*	(Optional) Removes the bandwidth specification from this class.
Step 6	switch(config-pmap-c-que)# **priority**	Specifies that traffic in this class is mapped to a strict priority queue. **Note** Only one class in each policy map can have strict priority set on it.
Step 7	switch(config-pmap-c-que)# **no priority**	(Optional) Removes the strict priority queuing from the traffic in this class.

This example shows how to define a type queuing policy map:

```
switch# configure terminal
switch(config)# policy-map type queuing policy-queue1
switch(config-pmap-que)# class type queuing class-queue1
switch(config-pmap-c-que)# bandwidth 20
```

Information About Marking

Marking is a method that you use to modify the QoS fields of the incoming and outgoing packets.

You can use marking commands in traffic classes that are referenced in a policy map. The marking features that you can configure are listed below:

- DSCP
- IP precedence
- CoS

Configuring CoS Marking

The value of the CoS field is recorded in the high-order three bits of the VLAN ID Tag field in the IEEE 802.1Q header.

Procedure

	Command or Action	Purpose
Step 1	switch# **configure terminal**	Enters global configuration mode.
Step 2	switch(config) # **policy-map** [**type network-qos**] *policy-map name*	Creates or accesses the policy map named *policy-map-name* and enters policy-map mode. The policy-map name can contain alphabetic, hyphen, or underscore characters, is case sensitive, and can be up to 40 characters.
Step 3	switch(config-pmap-nq) # **class** [**type network-qos**] {*class-map name* |**class-default**}	Creates a reference to *class-map-name* and enters policy-map class configuration mode. Use the **class-default** keyword to select all traffic that is not currently matched by classes in the policy map.
Step 4	switch(config-pmap-c-nq) # **set cos** *cos-value*	Specifies the CoS value to cos-value. The *cos-value* can range from 0 to 7. **Note** This command is only supported for egress policies.

Configuring DSCP Marking

You can set the DSCP value in the six most significant bits of the DiffServ field of the IP header to a specified value. You can enter numeric values from 0 to 60, in addition to the standard DSCP values shown in the table below:

Note You can set DSCP or IP Precedence but you can not set both values because they modify the same field in the IP packet.

Table 3: Standard DSCP Values

Value	List of DSCP Values
af11	AF11 dscp (001010)—decimal value 10
af12	AF12 dscp (001100)—decimal value 12
af13	AF13 dscp (001110)—decimal value 14
af21	AF21 dscp (010010)—decimal value 18
af22	AF22 dscp (010100)—decimal value 20

Value	List of DSCP Values
af23	AF23 dscp (010110)—decimal value 22
af31	AF31 dscp (011010)—decimal value 26
af32	AF40 dscp (011100)—decimal value 28
af33	AF33 dscp (011110)—decimal value 30
af41	AF41 dscp (100010)—decimal value 34
af42	AF42 dscp (100100)—decimal value 36
af43	AF43 dscp (100110)—decimal value 38
cs1	CS1 (precedence 1) dscp (001000)—decimal value 8
cs2	CS2 (precedence 2) dscp (010000)—decimal value 16
cs3	CS3 (precedence 3) dscp (011000)—decimal value 24
cs4	CS4 (precedence 4) dscp (100000)—decimal value 32
cs5	CS5 (precedence 5) dscp (101000)—decimal value 40
cs6	CS6 (precedence 6) dscp (110000)—decimal value 48
cs7	CS7 (precedence 7) dscp (111000)—decimal value 56
default	Default dscp (000000)—decimal value 0
ef	EF dscp (101110)—decimal value 46

Procedure

	Command or Action	Purpose
Step 1	**config t**	Enters configuration mode.
Step 2	**policy-map type qos** *qos-policy-map-name*	Creates or accesses the policy map named policy-map-name, and then enters policy-map mode. The policy-map name can

	Command or Action	Purpose
		contain alphabetic, hyphen, or underscore characters, is case sensitive, and can be up to 40 characters.
Step 3	**class** [**type qos**] {*class-map-name* | **class-default**}	Creates a reference to class-map-name, and enters policy-map class configuration mode. Use the **class-default** keyword to select all traffic that is not currently matched by classes in the policy map.
Step 4	**set dscp** *dscp-value*	Sets the DSCP value to dscp-value. See the Standards DSCP Values table.

This example shows how to display the policy-map configuration as shown below:

```
switch# show policy-map policy1
```

Configuring IP Precedence Marking

You can set the value of the IP precedence field in bits 0 to 2 of the IPv4 type of service (ToS) field. The following table shows the precedence values:

Note You can set IP Precedence or DSCP but you can not set both values because they modify the same field in the IP packet.

Table 4: Precedence Values

Value	List of Precedence Values
<0-7>	IP precedence value
critical	Critical precedence (5)
flash	Flash precedence (3)
flash-override	Flash override precedence (4)
immediate	Immediate precedence (2)
internet	Internetwork control precedence (6)
network	Network control precedence (7)
priority	Priority precedence (1)
routine	Routine precedence (0)

Procedure

	Command or Action	Purpose
Step 1	**config t**	Enters configuration mode.
Step 2	**policy-map** [**type qos**] *qos-policy-map-name*	Creates or accesses the policy map named *policy-map-name*, and then enters policy-map mode. The policy-map name can contain alphabetic, hyphen, or underscore characters, is case sensitive, and can be up to 40 characters.
Step 3	**class** [**type qos**] {*class-map-name* | **class-default**}	Creates a reference to class-map-name, and enters policy-map class configuration mode. Use the **class-default** keyword to select all traffic that is not currently matched by classes in the policy map.
Step 4	**set precedence** *precedence-value*	Sets the IP precedence value to precedence-value. You can enter one of the values shown in the Precedence Values table.

This example shows how to set the precedence marking to 5:

```
switch(config)# policy-map type qos my_policy
switch(config-pmap-qos)# class type qos my_class
switch(config-pmap-c-qos)# set precedence 5
switch(config-pmap-c-qos)#
```

Attaching the System Service Policy

The **service-policy** command specifies the system class policy map as the service policy for the system.

Procedure

	Command or Action	Purpose
Step 1	switch# **configure terminal**	Enters global configuration mode.
Step 2	switch(config)# **system qos**	Enters system class configuration mode.
Step 3	switch(config-sys-qos)# **service-policy type** {**network-qos** [**input** | **output**] | **qos input** |**queuing output**} *policy-name*	Specifies the policy map to use as the service policy for the system. There are three policy-map configuration modes: • network-qos—Network-wide (system qos) mode. • qos—Classification mode (system qos input or interface input only). • queuing—Queuing mode (output at system qos and interface).

	Command or Action	Purpose
		Note There is no default policy-map configuration mode; you must specify the **type**. The **input** keyword specifies that this policy map should be applied to traffic received on an interface. The **output** keyword specifies that this policy map should be applied to traffic transmitted from an interface. You can only apply **input** to a qos policy; you can only apply **output** to a queuing policy.

Restoring the Default System Service Policies

If you have created and attached new policies to the system QoS configuration, enter the **no** form of the command to reapply the default policies.

Procedure

	Command or Action	Purpose
Step 1	switch# **configure terminal**	Enters global configuration mode.
Step 2	switch(config)# **system qos**	Enters system class configuration mode.
Step 3	switch(config-sys-qos)# **no service-policy type qos input** *policy-map name*	Resets the classification mode policy map. This policy-map configuration is for system QoS input or interface input only:
Step 4	switch(config-sys-qos)# **no service-policy type network-qos** *policy-map name*	Resets the network-wide policy map.
Step 5	switch(config-sys-qos)# **no service-policy type queuing output** *policy-map name*	Resets the output queuing mode policy map.

```
switch# configure terminal
switch(config)# system qos
switch(config-sys-qos)# no service-policy type qos input my-in-policy
switch(config-sys-qos)# no service-policy type network-qos my-nq-policy
switch(config-sys-qos)# no service-policy type queuing output my-out-policy
```

Enabling the Jumbo MTU

You can enable the jumbo MTU for the whole switch by setting the MTU to its maximum size (9216 bytes) in the policy map for the default Ethernet system class (class-default).

Note The Cisco Nexus device supports 1 MTU for all classes for all ports.

This example shows how to configure the default Ethernet system class to support the jumbo MTU:

```
switch(config)# policy-map type network-qos jumbo
switch(config-pmap-nq)# class type network-qos class-default
switch(config-pmap-c-nq)# mtu 9216
switch(config-pmap-c-nq)# exit
switch(config-pmap-nq)# exit
switch(config)# system qos
switch(config-sys-qos)# service-policy type network-qos jumbo
```

Note The **system jumbomtu** command defines the maximum MTU size for the switch. However, jumbo MTU is only supported for system classes that have MTU configured.

Verifying the Jumbo MTU

On theCisco Nexus device, traffic is classified into one of eight QoS groups, and the MTU is configured at the QoS group level. Because the Cisco Nexus device supports different MTU values for different QoS groups, it is not possible to represent the MTU as one value for each interface. By default, all Ethernet traffic is in QoS group 0. Therefore, to verify the jumbo MTU for Ethernet traffic, use the **show queuing interface ethernet** *slot/chassis_number* command, and verify that the HW MTU field for QoS group 0 is 9216. The **show interface** command always shows an MTU value of 1500, which is the expected value.

This example shows how to display jumbo MTU information for Ethernet 1/19:

```
switch# sh queuing int e1/19
Ethernet1/19 queuing information:
  TX Queuing
    qos-group  sched-type  oper-bandwidth
        0       WRR             50
        1       WRR             50

  RX Queuing
    qos-group 0
    q-size: 243200, HW MTU: 9280 (9216 configured)
    drop-type: drop, xon: 0, xoff: 1520
    Statistics:
        Pkts received over the port             : 2119963420
        Ucast pkts sent to the cross-bar        : 2115648336
        Mcast pkts sent to the cross-bar        : 4315084
        Ucast pkts received from the cross-bar  : 2592447431
        Pkts sent to the port                   : 2672878113
        Pkts discarded on ingress               : 0
        Per-priority-pause status               : Rx (Inactive), Tx (Inactive)

    qos-group 1
    q-size: 76800, HW MTU: 2240 (2158 configured)
    drop-type: no-drop, xon: 128, xoff: 240
    Statistics:
        Pkts received over the port             : 0
        Ucast pkts sent to the cross-bar        : 0
        Mcast pkts sent to the cross-bar        : 0
        Ucast pkts received from the cross-bar  : 0
        Pkts sent to the port                   : 0
        Pkts discarded on ingress               : 0
        Per-priority-pause status               : Rx (Inactive), Tx (Inactive)

  Total Multicast crossbar statistics:
    Mcast pkts received from the cross-bar      : 80430744
```

Configuring QoS on Interfaces

Configuring Untagged CoS

Any incoming packet not tagged with an 802.1p CoS value is assigned the default untagged CoS value of zero (which maps to the default Ethernet drop system class). You can override the default untagged CoS value for an Ethernet or EtherChannel interface.

Procedure

	Command or Action	Purpose
Step 1	switch# **configure terminal**	Enters configuration mode.
Step 2	switch(config)# **interface** {**ethernet** [*chassis*/]*slot*/*port* | **port-channel** *channel-number*}	Enters configuration mode for the specified interface or port channel.
Step 3	switch(config-if)# **no switchport**	(Optional) Selects a Layer 3 interface.
Step 4	switch(config-if)# **untagged cos** *cos-value*	Configures the untagged CoS value. Values can be from 1 to 7.

This example shows how to set the CoS value to 4 for untagged frames received on an interface:

```
switch# configure terminal
switch(config)# interface ethernet 1/2
switch(config-if)# untagged cos 4
```

This example shows how to set the CoS value to 3 for untagged frames received on a Layer 3 interface:

```
switch# configure terminal
switch(config)# interface ethernet 1/5
switch(config-if)# no switchport
switch(config-if)# untagged cos 3
switch(config-if)#
```

Configuring Buffers and Queues

Configuring a Multicast Slow Receiver Port

When you have a combination of 10-gigabyte and 1-gigabyte ports, you can use this command on the 1-gigabyte port to reduce the effects of the 1-gigabyte port blocking the 10-gigabyte port. Use this command on the 1-gigabyte port only when there is Head-of-Line Blocking (HOLB) on the 10-gigabyte port due to a slow receiver on the 1-gigabyte port.

Procedure

	Command or Action	Purpose
Step 1	switch# **configure terminal**	Enters global configuration mode.
Step 2	switch(config)# **hardware profile multicast slow-receiver port** {**all** | **port** *port-number*}	Configures all 1-gigabyte ports or the specified 1-gigabyte port as a slow-receiver port so that it does not block the 10-gigabyte ports.
Step 3	switch(config)# **copy running-config startup-config**	(Optional) Saves the change persistently through reboots and restarts by copying the running configuration to the startup configuration.

The following example shows how to configure port 46 as the multicast slow-receiver port:

```
switch# configure terminal
switch(config)# hardware profile multicast slow-receiver port 46
switch(config)# copy running-config startup-config
```

Configuring the Percentage of Buffer Used for a Specific QoS Group or Virtual Lane

You can configure the percentage of shared buffer used for a specific QoS group or virtual lane (VL)

Procedure

	Command or Action	Purpose
Step 1	switch# **configure terminal**	Enters global configuration mode.
Step 2	switch# **hardware profile buffer qosgroup** *number* **threshold** *percentage*	Configures the buffer for the specified QoS group. The *number* argument specifies the QoS group number. The range is from 0 to 4. The *percentage* argument specified the percentage of maximum usages. The range is from 1 to 100.
Step 3	switch(config)# **copy running-config startup-config**	(Optional) Saves the change persistently through reboots and restarts by copying the running configuration to the startup configuration.

The following example shows how to configure the shared buffer for QoS group 1 to a maximum of 40 percent usage:

```
switch# configure terminal
```

```
switch(config)# hardware profile buffer qosgroup 1 threshold 40
switch(config)# copy running-config startup-config
```

Configuring the Percentage of Buffer Used for SPAN Traffic

You can configure the percentage of shared buffer used for SPAN traffic.

Procedure

	Command or Action	Purpose
Step 1	switch# **configure terminal**	Enters global configuration mode.
Step 2	switch# **hardware buffer span-threshold** *percentage*	Configures the percentage of maximum usage of the hardware buffer for SPAN traffic. The *percentage* range is from 0 to 100.
Step 3	switch(config)# **copy running-config startup-config**	(Optional) Saves the change persistently through reboots and restarts by copying the running configuration to the startup configuration.

The following example shows how to configure 30 percent of the hardware buffer for SPAN traffic:

```
switch# configure terminal
switch(config)# hardware buffer span-threshold 30
switch(config)# copy running-config startup-config
```

Verifying the Qos Configuration

To verify the QoS configurations, perform one of these tasks:

Command	Purpose
switch# **show class-map**	Displays the class maps defined on the device.
switch# **show policy-map** [*name*]	Displays the policy maps defined on the device. Optionally, you can display the named policy only.
switch# **show policy-map interface** [*interface number*]	Displays the policy map settings for an interface or all interfaces.
switch# **show policy-map system**	Displays the policy map settings attached to the system qos.
switch# **show policy-map type** {**network-qos** | **qos** | **queuing**} [*name*]	Displays the policy map settings for a specific policy type. Optionally, you can display the named policy only.

Command	Purpose
switch# **show interface untagged-cos** [**module** *number*]	Displays the untagged CoS values for all interfaces.
switch# **show wrr-queue cos-map** [*var*]	Displays the mapped CoS values to egress queues.
switch# **running-config ipqos**	Displays information about the running configuration for QoS.
switch# **startup-config ipqos**	Displays information about the startup configuration for QoS.
switch# **show queuing interface ethernet** *slot-no/port-no*	Displays the queuing information on interfaces.

This example shows how to configure a network QoS policy:

```
switch(config)# class-map type network-qos cnq1
switch(config-cmap-nq)# match qos-group 1
switch(config-cmap-nq)# exit
switch(config)# class-map type network-qos cnq6
switch(config-cmap-nq)# match qos-group 6
switch(config-cmap-nq)#
switch(config-cmap-nq)# exit
switch(config)# policy-map type network-qos pnqos
switch(config-pmap-nq)# class type network-qos cnq1
switch(config-pmap-nq-c)# set cos 4
switch(config-pmap-nq-c)# exit
switch(config-pmap-nq)# class type network-qos cnq6
switch(config-pmap-nq-c)# set cos 5
switch(config-pmap-nq-c)# congestion-control random-detect ecn
switch(config-pmap-nq-c)# exit
switch(config-pmap-nq)# class type network-qos class-default
switch(config-pmap-nq-c)# mtu 9216
switch(config-pmap-nq-c)# exit
switch(config-pmap-nq)# exit
switch(config)# system qos
switch(config-sys-qos)# service-policy type network-qos pnqos
switch(config-sys-qos)#
```

This example shows how to configure a queuing policy:

```
switch(config)# class-map type queuing cqu1
switch(config-cmap-que)# match qos-group 1
switch(config-cmap-que)# exit
switch(config)# class-map type queuing cqu6
switch(config-cmap-que)# match qos-group 6
switch(config-cmap-que)# exit
switch(config)# policy-map type queuing pqu
switch(config-pmap-que)# class type queuing class-default
switch(config-pmap-c-que)# bandwidth percent 70
switch(config-pmap-c-que)# exit
switch(config-pmap-que)# class type queuing cqu1
switch(config-pmap-c-que)# bandwidth percent 10
switch(config-pmap-c-que)# exit
switch(config-pmap-que)# class type queuing cqu6
switch(config-pmap-c-que)# bandwidth percent 20
switch(config-pmap-c-que)# exit
switch(config-pmap-que)# exit
switch(config)# system qos
switch(config-sys-qos)# service-policy type queuing output pqu
switch(config-sys-qos)#
```

This example shows how to configure a QoS policy:

```
switch(config)# class-map type qos cqos1
switch(config-cmap-qos)# match cos 1
switch(config-cmap-qos)# exit
switch(config)# class-map type qos cqos6
switch(config-cmap-qos)# match cos 6
switch(config-cmap-qos)# exit
switch(config)# policy-map type qos pqos
switch(config-pmap-qos)# class type qos cqos1
switch(config-pmap-c-qos)# set qos-group 1
switch(config-pmap-c-qos)# exit
switch(config-pmap-qos)# class type qos cqos6
switch(config-pmap-c-qos)# set qos-group 6
switch(config-pmap-c-qos)# exit
switch(config-pmap-qos)# exit
switch(config)# system qos
switch(config-sys-qos)# service-policy type qos input pqos
switch(config-sys-qos)#
```

This example shows how to verify the untagged-cos configuration on interfaces:

```
switch(config-if)# show interface untagged-cos
================================

Interface      Untagged-CoS
================================

Ethernet1/1  4
Ethernet1/2
Ethernet1/3  5
Ethernet1/4
Ethernet1/5
Ethernet1/6
Ethernet1/7
Ethernet1/8
Ethernet1/9
Ethernet1/10
Ethernet1/11
Ethernet1/12
Ethernet1/13
Ethernet1/14
Ethernet1/15
Ethernet1/16
Ethernet1/17
```

This example shows how to display the QoS running configuration:

```
switch(config)# show running-config ipqos

!Command: show running-config ipqos
!Time: Mon Mar 15 08:24:12 2010

version 5.0(3)U1(1)
class-map type qos match-all cqos1
  match cos 1
class-map type qos match-all cqos6
  match cos 6
class-map type queuing cqu1
  match qos-group 1
class-map type queuing cqu6
  match qos-group 6
policy-map type qos pqos
  class cqos1
    set qos-group 1
  class cqos6
    set qos-group 6
policy-map type queuing pqu
  class type queuing cqu1
    bandwidth percent 10
  class type queuing cqu6
    bandwidth percent 20
  class type queuing class-default
    bandwidth percent 70
```

```
class-map type network-qos cnq1
  match qos-group 1
class-map type network-qos cnq6
  match qos-group 6
policy-map type network-qos pnqos
  class type network-qos cnq1
    set cos 4
  class type network-qos cnq6
    set cos 5
    congestion-control random-detect ecn
  class type network-qos class-default
    mtu 9216
system qos
  service-policy type qos input pqos
  service-policy type network-qos pnqos
  service-policy type queuing output pqu


interface Ethernet1/1
  untagged cos 4

interface Ethernet1/3
  untagged cos 5

switch(config)#
```

This example shows how to display the class map configuration:

```
switch(config)# show class-map


  Type qos class-maps
  ===================

    class-map type qos match-all cqos1
      match cos 1

    class-map type qos match-all cqos6
      match cos 6

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


  Type queuing class-maps
  =======================

    class-map type queuing cqu1
      match qos-group 1

    class-map type queuing cqu6
      match qos-group 6

    class-map type queuing class-default
      match qos-group 0



  Type network-qos class-maps
  ===========================

    class-map type network-qos cnq1
      match qos-group 1

    class-map type network-qos cnq6
      match qos-group 6

    class-map type network-qos class-default
      match qos-group 0

switch(config)#
```

This example shows how to display the policy map configuration:

```
switch(config)# show policy-map


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

  policy-map type qos pqos
    class type qos cqos1
      set qos-group 1
    class type qos cqos6
      set qos-group 6
    class type qos class-default
      set qos-group 0
  policy-map type qos default-in-policy
    class type qos class-default
      set qos-group 0

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

  policy-map type queuing pqu
    class type queuing cqu1
      bandwidth percent 10
    class type queuing cqu6
      bandwidth percent 20
    class type queuing class-default
      bandwidth percent 70
  policy-map type queuing default-out-policy
    class type queuing class-default
      bandwidth percent 100


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

  policy-map type network-qos pnqos
    class type network-qos cnq1
      mtu 1500
      set cos 4
    class type network-qos cnq6
      mtu 1500
      set cos 5
      congestion-control random-detect ecn
    class type network-qos class-default
      mtu 9216
  policy-map type network-qos default-nq-policy
    class type network-qos class-default
      mtu 1500
switch(config)#
```

This example shows how to display all active policy maps in the system:

```
switch(config)# show policy-map system


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

  policy-map type network-qos pnqos
    class type network-qos cnq1      match qos-group 1

      mtu 1500
      set cos 4
    class type network-qos cnq6      match qos-group 6

      mtu 1500
      set cos 5
      congestion-control random-detect ecn
    class type network-qos class-default      match qos-group 0

      mtu 9216
```

```
  Service-policy (qos) input:   pqos
    policy statistics status:   disabled

    Class-map (qos):   cqos1 (match-all)
      Match: cos 1
      set qos-group 1

    Class-map (qos):   cqos6 (match-all)
      Match: cos 6
      set qos-group 6

    Class-map (qos):   class-default (match-any)
      Match: any
      set qos-group 0

  Service-policy (queuing) output:   pqu
    policy statistics status:   disabled

    Class-map (queuing):   cqu1 (match-any)
      Match: qos-group 1
      bandwidth percent 10

    Class-map (queuing):   cqu6 (match-any)
      Match: qos-group 6
      bandwidth percent 20

    Class-map (queuing):   class-default (match-any)
      Match: qos-group 0
      bandwidth percent 70

switch(config)#
```

This example shows how to display the service policy maps configured on the interfaces:

```
switch(config)# show policy-map interface ethernet 1/1


Global statistics status :   disabled

Ethernet1/1

  Service-policy (qos) input:   pqos
    policy statistics status:   disabled

    Class-map (qos):   cqos1 (match-all)
      Match: cos 1
      set qos-group 1

    Class-map (qos):   cqos6 (match-all)
      Match: cos 6
      set qos-group 6

    Class-map (qos):   class-default (match-any)
      Match: any
      set qos-group 0

  Service-policy (queuing) output:   pqu
    policy statistics status:   disabled

    Class-map (queuing):   cqu1 (match-any)
      Match: qos-group 1
      bandwidth percent 10

    Class-map (queuing):   cqu6 (match-any)
      Match: qos-group 6
      bandwidth percent 20

    Class-map (queuing):   class-default (match-any)
      Match: qos-group 0
      bandwidth percent 70

switch(config)#
```

This example shows how to display the queuing information for a specific interface:

```
switch(config)# show queuing interface ethernet 1/1
Ethernet1/1 queuing information:
  TX Queuing
    qos-group  sched-type  oper-bandwidth
        0       WRR             70
        1       WRR             10
        6       WRR             20

  RX Queuing
    qos-group 0
    HW MTU: 1500 (1500 configured)
    drop-type: drop, xon: 0, xoff: 0
    Statistics:
        Ucast pkts sent over the port           : 0
        Ucast bytes sent over the port          : 0
        Mcast pkts sent over the port           : 0
        Mcast bytes sent over the port          : 0
        Ucast pkts dropped                      : 0
        Ucast bytes dropped                     : 0
        Mcast pkts dropped                      : 0
        Mcast bytes dropped                     : 0
    qos-group 1
    HW MTU: 1500 (1500 configured)
    drop-type: drop, xon: 0, xoff: 0
    Statistics:
        Ucast pkts sent over the port           : 0
        Ucast bytes sent over the port          : 0
        Mcast pkts sent over the port           : 0
        Mcast bytes sent over the port          : 0
        Ucast pkts dropped                      : 0
        Ucast bytes dropped                     : 0
        Mcast pkts dropped                      : 0
        Mcast bytes dropped                     : 0
    qos-group 6
    HW MTU: 1500 (1500 configured)
    drop-type: drop, xon: 0, xoff: 0
    Statistics:
        Ucast pkts sent over the port           : 0
        Ucast bytes sent over the port          : 0
        Mcast pkts sent over the port           : 0
        Mcast bytes sent over the port          : 0
        Ucast pkts dropped                      : 0
        Ucast bytes dropped                     : 0
        Mcast pkts dropped                      : 0
        Mcast bytes dropped                     : 0
switch(config)#
```

INDEX

R

S

T

V