Send document comments to nexus7k-docfeedback@cisco.com.

Cisco Nexus 7000 Series NX-OS Quality of Service Configuration Guide, Release 4.0

November 5, 2009

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-12927-02

Cisco Nexus 7000 Series NX-OS Quality of Service Configuration Guide, Release 4.0

CONTENTS

Preface

This preface describes the audience, organization, and conventions of the *Cisco Nexus 7000 Series NX-OS Quality of Service Configuration Guide, Release 4.0*. It also provides information on how to obtain related documentation.

This chapter includes the following sections:

Audience

This guide is for experienced network administrators who configure and maintain NX-OS devices.

Organization

This publication is organized as follows:

Chapter	Description
Chapter 1, “Overview”	Provides an overview of QoS features.
Chapter 2, “Using Modular QoS CLI”	Describes how to use CPL to define QoS policies.
Chapter 3, “Configuring Classification”	Describes how to configure the classification feature.
Chapter 4, “Configuring Marking”	Describes how to configure the marking feature.
Chapter 5, “Configuring Mutation Mapping”	Describes how to configure the mutation feature.
Chapter 6, “Configuring Policing”	Describes how to configure the policing feature.
Chapter 7, “Configuring Queuing and Scheduling”	Describes how to configure the queuing and scheduling feature.
Chapter 8, “Monitoring QoS Statistics”	Describes how to view QoS statistics.

Chapter	Description
Appendix A, "Configuration Limits for Cisco NX-OS Quality of Service Configuration Features, Release 4.0"	Lists information related to numerical limitations in implementing QoS.
Appendix B, "Additional References"	Lists related documents and RFCs, as well as how to get Cisco support help.

Document Conventions

This publication 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.

Tip Means *the following information will help you solve a problem*.

Command descriptions use these conventions:

boldface font	Commands and keywords are in boldface.
italic font	Arguments for which you supply values are in italics.
{ }	Elements in braces are required choices.
[ ]	Elements in square brackets are optional.
x | y | x	Alternative, mutually exclusive elements are separated by vertical bars.

Screen examples use these conventions:

`screen font`	Terminal sessions and information that the switch displays are in screen font.
`boldface screen font`	Information that 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 number sign (#) at the beginning of a line of code indicates a comment line.

Send document comments to nexus7k-docfeedback@cisco.com.

Related Documentation

Cisco NX-OS documentation is available at the following URL:

http://www.cisco.com/en/US/products/ps9372/tsd_products_support_series_home.html

The documentation set for Cisco NX-OS includes the following documents:

Release Notes

Cisco Nexus 7000 Series NX-OS Release Notes, Release 4.0

NX-OS Configuration Guides

Cisco Nexus 7000 Series NX-OS Getting Started with Virtual Device Contexts, Release 4.0

Cisco Nexus 7000 Series NX-OS Fundamentals Configuration Guide, Release 4.0

Cisco Nexus 7000 Series NX-OS Interfaces Configuration Guide, Release 4.0

Cisco Nexus 7000 Series NX-OS Layer 2 Switching Configuration Guide, Release 4.0

Cisco Nexus 7000 Series NX-OS Quality of Service Configuration Guide, Release 4.0

Cisco Nexus 7000 Series NX-OS Unicast Routing Configuration Guide, Release 4.0

Cisco Nexus 7000 Series NX-OS Multicast Routing Configuration Guide, Release 4.0

Cisco Nexus 7000 Series NX-OS Security Configuration Guide, Release 4.0

Cisco Nexus 7000 Series NX-OS Virtual Device Context Configuration Guide, Release 4.0

Cisco Nexus 7000 Series NX-OS Software Upgrade Guide, Release 4.0

Cisco Nexus 7000 Series NX-OS Licensing Guide, Release 4.0

Cisco Nexus 7000 Series NX-OS High Availability and Redundancy Guide, Release 4.0

Cisco Nexus 7000 Series NX-OS System Management Configuration Guide, Release 4.0

Cisco Nexus 7000 Series NX-OS XML Management Interface User Guide, Release 4.0

Cisco Nexus 7000 Series NX-OS System Messages Reference

Cisco Nexus 7000 Series NX-OS MIB Quick Reference

NX-OS Command References

Cisco Nexus 7000 Series NX-OS Command Reference Master Index, Release 4.0

Cisco Nexus 7000 Series NX-OS Fundamentals Command Reference, Release 4.0

Cisco Nexus 7000 Series NX-OS Interfaces Command Reference, Release 4.0

Cisco Nexus 7000 Series NX-OS Layer 2 Switching Command Reference, Release 4.0

Cisco Nexus 7000 Series NX-OS Quality of Service Command Reference, Release 4.0

Cisco Nexus 7000 Series NX-OS Unicast Routing Command Reference, Release 4.0

Cisco Nexus 7000 Series NX-OS Multicast Routing Command Reference, Release 4.0

Cisco Nexus 7000 Series NX-OS Security Command Reference, Release 4.0

Cisco Nexus 7000 Series NX-OS Virtual Device Context Command Reference, Release 4.0

Cisco NX-OS High Availability and Redundancy Command Reference, Release 4.0

Send document comments to nexus7k-docfeedback@cisco.com.

Cisco Nexus 7000 Series NX-OS System Management Command Reference, Release 4.0

Other Software Document

Cisco Nexus 7000 Series NX-OS Troubleshooting Guide, Release 4.0

Obtaining Documentation, Obtaining Support, and Security Guidelines

For information on obtaining documentation, submitting a service request, and gathering additional information, see the monthly *What's New in Cisco Product Documentation*, which also lists all new and revised Cisco technical documentation, at:

http://www.cisco.com/en/US/docs/general/whatsnew/whatsnew.html

Subscribe to the *What's New in Cisco Product Documentation* as a Really Simple Syndication (RSS) feed and set content to be delivered directly to your desktop using a reader application. The RSS feeds are a free service and Cisco currently supports RSS version 2.0.

CHAPTER 1

Overview

This chapter describes the configurable Cisco NX-OS Quality of Service (QoS) features on the device.

QoS allows you to classify the network traffic, police and prioritize the traffic flow, and provide congestion avoidance.

This chapter includes the following sections:

Information About 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, police and 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 the policing feature and the marking features.
- Queuing policies use the queuing and scheduling features as well as a limited set of the marking feature.

Note The system-defined QoS features and values that are discussed in Chapter 2, “Using Modular QoS CLI” apply globally to the entire switch and cannot be modified. See the *Cisco Nexus 7000 Series NX-OS Virtual Device Context Configuration Guide, Release 4.0* for complete information on VDCs.

This section includes the following topics:

Using QoS

Traffic is processed based on how you classify it and the policies that you create and apply to traffic classes.

To configure QoS features, you use the following steps:

1. Create traffic classes by classifying the incoming and outgoing packets that match criteria such as IP address or QoS fields.
2. Create policies by specifying actions to take on the traffic classes, such as limiting, marking, or dropping packets.
3. Apply policies to a port, port channel, VLAN, or a subinterface.

You use MQC to create the traffic classes and policies of the QoS features. For more information, see Chapter 2, “Using Modular QoS CLI.”

Note The queuing and scheduling operations of the overall QoS feature use IPv6 and IPv4; the rest of the feature uses only IPv4.

Classification

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), Layer 2 to Layer 4 parameters, and the packet length.

The values used to classify traffic are called match criteria. When you define a traffic class, you can specify multiple match criteria, you can choose to not match on a particular criterion, or you can determine 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.

For more information about configuring classification, see Chapter 3, “Configuring Classification.”

Marking

Marking is the setting of QoS information that is related to a packet. You can set the value of standard QoS fields IP precedence, DSCP and Class of Service (CoS), and internal labels that can be used in subsequent actions. Marking is used to identify the traffic type for use in policing, queuing, and scheduling traffic (only CoS is used in scheduling).

For more information about configuring marking, see Chapter 4, “Configuring Marking.”

Mutation

Mutation is the changing of packet header QoS fields. You can map IP precedence, DSCP, or CoS values to all incoming or outgoing packets. You can use mutation in policies that contain policing commands, but you cannot use mutation in queuing and scheduling commands. You use configurable, user-defined table maps for mutation.

For more information about configuring mutation, see Chapter 5, “Configuring Mutation Mapping.”

Policing

Policing is the monitoring of data rates for a particular class of traffic. The device can also monitor associated burst sizes.

Three “colors,” or conditions, are determined by the policer depending on the data rate parameters supplied: conform (green), exceed (yellow), or violate (red). You can configure only one action for each condition. When the data rate exceeds the user-supplied values, packets are either marked down or dropped. You can define single-rate, dual-rate, and color-aware policers.

Single-rate policers monitor the specified committed information rate (CIR) of traffic. Dual-rate policers monitor both CIR and peak information rate (PIR) of traffic. Color-aware policers assume that traffic has been previously marked with a color.

For more information about configuring policing, see Chapter 6, “Configuring Policing.”

Queuing and Scheduling

The queuing and scheduling process allows you to control the bandwidth allocated to traffic classes, so you achieve the desired trade-off between throughput and latency.

You can apply weighted random early detection (WRED) to a class of traffic, which allows packets to be dropped based on the Class of Service (CoS) field. The WRED algorithm allows you to perform proactive queue management to avoid traffic congestion.

You can schedule traffic by imposing a maximum data rate on a class of traffic so that excess packets are retained in a queue to smooth (constrain) the output rate.

For information about configuring queuing and scheduling, see Chapter 7, “Configuring Queuing and Scheduling.”

Sequencing of QoS Actions

The following are the two types of policies:

- **qos**—Defines MQC objects that you can use for marking and policing.
- **queuing**—Defines MQC objects that you can use for queuing and scheduling as well as a limited set of the marking objects.

Note The default type of policy is **qos**. You cannot apply QoS policies on egress interfaces.

The Cisco NX-OS device processes the QoS policies that you define based on whether they are applied to ingress or egress packets. The system performs actions for QoS policies only if you define them under the type **qos** service policies.

Note You can apply only ingress traffic actions for QoS policies on Layer 2 interfaces. You can apply both ingress and egress traffic actions on Layer 3 interfaces.

This section includes the following topics:

Sequencing of Ingress Traffic Actions

The sequence of QoS actions on ingress traffic is as follows:

1. Queuing and scheduling
2. Mutation
3. Classification
4. Marking
5. Policing

Sequencing of Egress Traffic Actions

The sequencing of QoS actions on egress traffic is as follows:

1. Classification
2. Marking
3. Policing
4. Mutation
5. Queuing and scheduling

Note Mutation happens much closer to the beginning of the traffic actions on the ingress packets, and any further classification and policing is based on the changed QoS values. Mutation happens at the end of the traffic actions on the egress packets, right before queuing and scheduling.

High Availability Requirements for QoS Features

The Cisco NX-OS QoS software recovers its previous state after a software restart, and it is able to switch over from the active supervisor to the standby supervisor without a loss of state.

Note See the *Cisco Nexus 7000 Series NX-OS High Availability and Redundancy Guide, Release 4.0* for complete information on high availability.

QoS Feature Configuration with MQC

You use MQC to configure QoS features. The MQC configuration commands are shown in Table 1-1.

Table 1-1 MQC Configuration Commands

MQC Command	Description
class-map	Defines a class map that represents a class of traffic.
table-map	Defines a table map that represents a mapping from one set of field values to another set of field values. You can reference a table map from a policy map.
policy-map	Defines a policy map that represents a set of policies to be applied to a set of class maps. Policy maps can reference table maps.

You can modify or delete MQC objects, except system-defined objects, when the objects are not associated with any interfaces. See Chapter 2, "Using Modular QoS CLI" for information on system-defined MQC objects.

Once a QoS policy is defined, you can attach the policy map to an interface using the interface configuration command shown in Table 1-2.

Table 1-2 Interface Command to Attach a Policy Map to an Interface

Interface Command	Description
service-policy	Applies the specified policy map to input or output packets on the interface.

For information about using MQC, see Chapter 2, "Using Modular QoS CLI."

QoS Statistics

Statistics are maintained for each policy, class action, and match criteria per interface. You can enable or disable the collection of statistics, you can display statistics using the **show policy-map** interface command, and you can clear statistics based on an interface or policy map with the **clear qos statistics** command. Statistics are enabled by default and can be disabled globally.

For information about monitoring QoS statistics, see Chapter 8, "Monitoring QoS Statistics."

Default QoS Behavior

The QoS queuing features are enabled by default. Specific QoS-type features, policing and marking, are enabled only when a policy is attached to an interface. Specific policies are enabled when that policy is attached to an interface.

By default, the device always enables a system default queuing policy, or system-defined queuing policy map, on each port and port channel. When you configure a queuing policy and apply the new queuing policy to specified interfaces, the new queuing policy replaces the default queuing policy and those rules now apply. For more information on the system-defined, default queuing policies and the default values that apply to each interface, see Chapter 2, “Using Modular QoS CLI.”

The device enables other QoS features, policing and marking, only when you apply a policy map to an interface.

CHAPTER 2

Using Modular QoS CLI

This chapter describes how to configure Modular QoS CLI (MQC) objects that can be used for configuring QoS features.

This chapter includes the following sections:

Information About MQC

MQC provides a language to define QoS policies.

Note MQC commands are included in the *Cisco Nexus 7000 Series NX-OS Quality of Service Command Reference, Release 4.0*

You configure QoS policies using three steps:

1. Define traffic classes.
2. Associate policies and actions with each traffic class.
3. Attach policies to logical or physical interfaces and VLANs.

MQC provides three 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.
- **table-map**—Defines a table map that represents a mapping from one set of packet field values to another set of packet fields. Table maps are referenced in policy maps.
- **policy-map**—Defines a policy map that represents a set of policies to be applied on a class-by-class basis to class maps.

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

- **qos**—Defines MQC objects that you can use for marking and policing.

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

Note The **qos** type is the default.

You can attach policies to ports, port channels, VLANs, subinterfaces, or tunnels by using the **service-policy** interface configuration command.

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

Note The system may accept QoS and ACL commands if you are working in the interface configuration mode and the module on which the interface is present is reloaded, even if that module is not present in the system at that time.

Licensing Requirements for Using MQC Objects

The following table shows the licensing requirements for this feature:

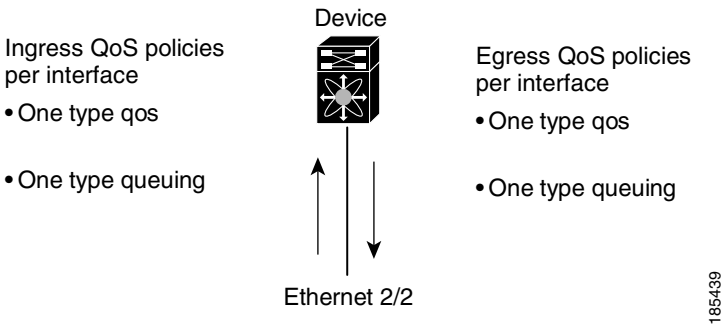
Product	License Requirement
NX-OS	QoS requires no license. Any feature not included in a license package is bundled with the Cisco NX-OS system images and is provided at no extra charge to you. For a complete explanation of the NX-OS licensing scheme, see the *Cisco Nexus 7000 Series NX-OS Licensing Guide, Release 4.0*.

However, using VDCs requires an Advanced Services license.

Using an MQC Object

You configure QoS and queuing policies using the MQC class-map, policy-map, and table-map objects. You cannot use table maps in queuing policies. After you configure class maps and policy maps, you can attach one policy map of each type to each of the ingress or egress directions of an interface. Figure 2-1 lists the maximum QoS and queuing policies that you can define on each interface.

Figure 2-1 Maximum QoS Policies Per Interface



A policy map contains either a QoS policy or queuing policy. The policy map references the names of class maps that represent traffic classes. For each class of traffic, the device applies the policies on the interface or VLAN that you select.

A packet is matched sequentially to a class of traffic starting from the first traffic class definition. When a match is found, the policy actions for that class are applied to the packet.

The reserved class map receives all traffic that is not matched in type qos policies, and the device applies the policy actions as it would for any other traffic class. You use class-default to perform mutations (mutation is a method for translating QoS values in the packet header prior to traffic classification).

Note You can access user-defined MQC objects only in the virtual device context (VDC) in which they were created. You can access the system-defined MQC objects in all VDCs.

This section includes the following topics:

Type qos Policies

You use type qos policies to mark, to apply mutations, to set the ingress port trust state, and to police packets.

Figure 2-2 shows the QoS policy structure with the associated MQC objects of type qos without mutation, and Figure 2-3 shows the QoS policy structure with mutation. The MQC objects are shown in bold.

Figure 2-2 QoS Policy Diagram Showing Type qos MQC Object Usage without Mutation

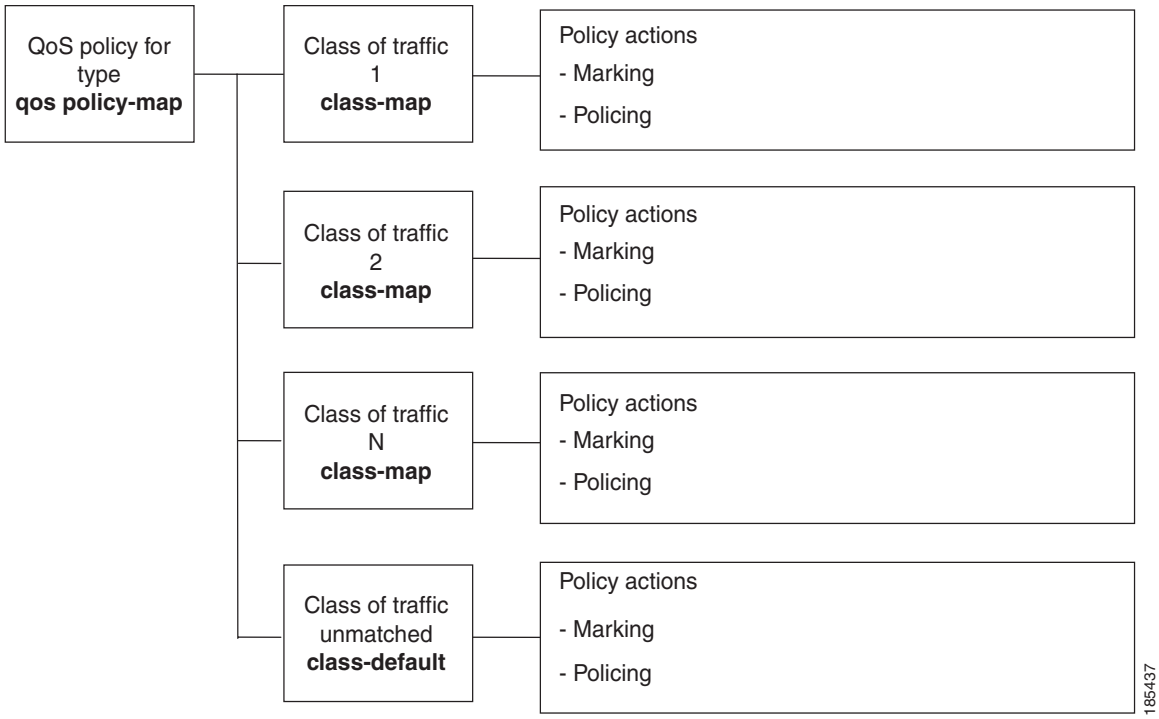
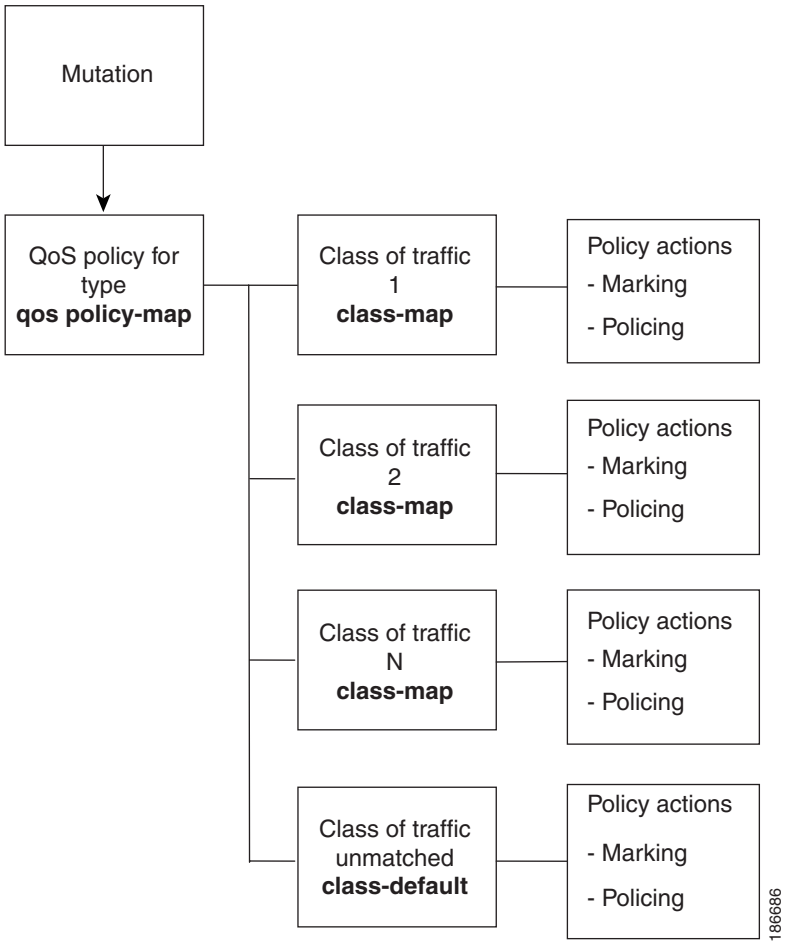


Figure 2-3 QoS Policy Diagram Showing Type qos MQC Object Usage with Mutation



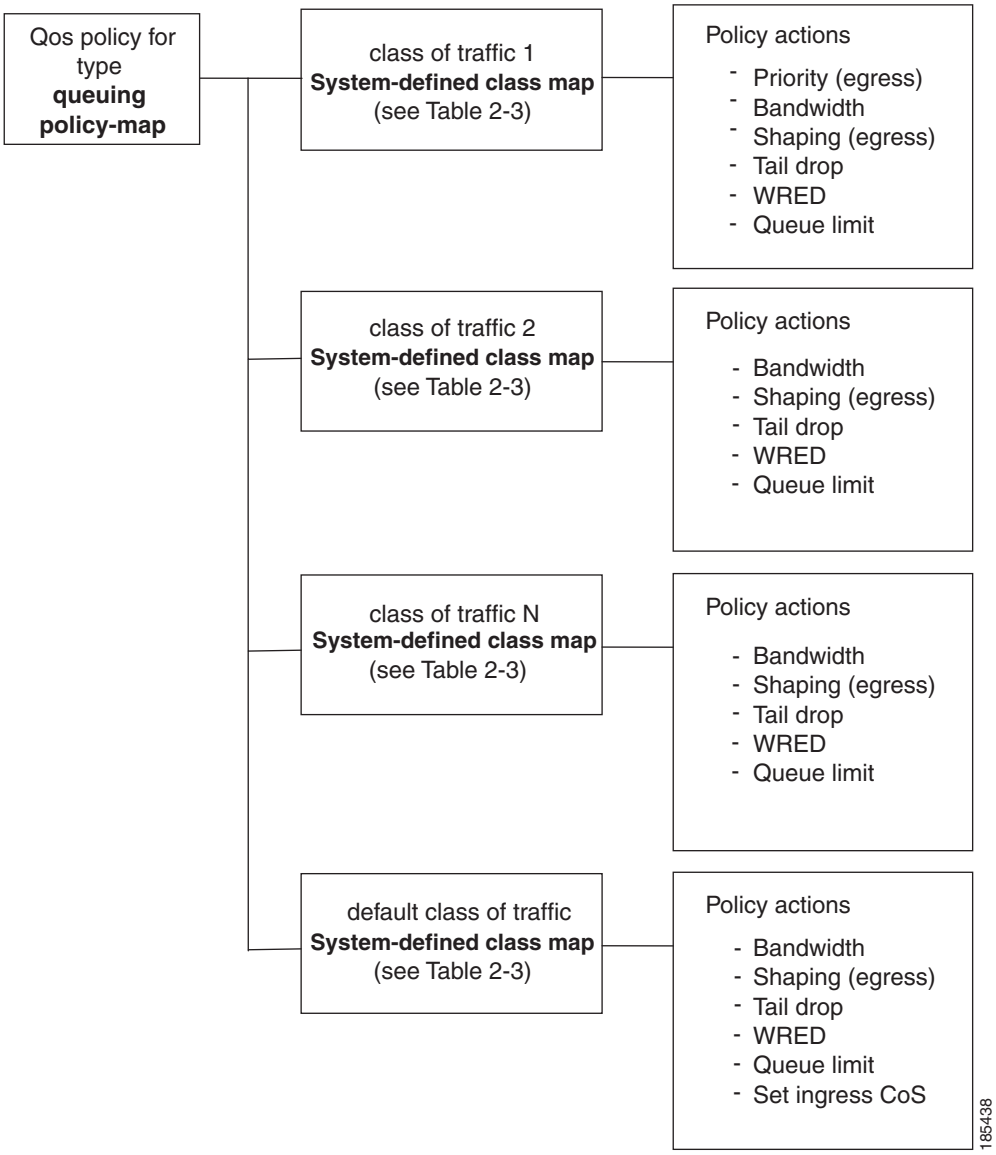
Type queuing Policies

You use type queuing policies to mark, shape, and queue packets. Marking is limited to the CoS field and does not support the use of table maps.

Figure 2-4 shows the QoS policy structure with associated MQC objects of type queuing. The MQC objects are shown in bold.

Note MQC table-map objects cannot be used in policies of type queuing.

Figure 2-4 QoS Policy Diagram Showing Type queuing MQC Object Usage



Note: See Chapter 5, "Queuing and Scheduling," for information on configuring these parameters.

System-Defined MQC Objects

Note These are the default MQC objects. All of these values apply across all VDCs.

When you configure QoS features, and the systems requests one of these MQC objects, you can use these system-defined objects.The system-defined MQC objects are shown in Table 2-1. See the tables listed next to the object for information on these system-defined objects.

Type qos class maps that are defined by the system are listed in Table 2-2.

Note You cannot reference the conform-color-in, conform-color-out, exceed-color-in, or exceed-color-out class maps in a policy map.

Table 2-2 System-Defined Type qos Class Maps

Class Map Name	Description
class-default	Type qos class map that is assigned to all packets that match none of the criteria of traffic classes that you define in a type qos policy map. You can use class-default for mutation.
conform-color-in	Type qos conform color class map in the input direction. This color-aware class map makes a policer color-aware for conform action.
conform-color-out	Type qos conform color class map in the output direction. This color-aware class map makes a policer color-aware for conform action.
exceed-color-in	Type qos exceed color class map in the input direction. This color-aware class map makes a policer color-aware for exceed action.
exceed-color-out	Type qos exceed color class map in the output direction. This color-aware class map makes a policer color-aware for exceed action.

Type queuing class maps that are defined by the system are listed in Table 2-3.

Table 2-3 System-Defined Type queuing Class Maps

Class Map Queue Name	Description	Default CoS Values
1 Gigabit Module Ingress: 2 queues with 4 thresholds per queue		
2q4t-in-q1	Ingress queue 1 of 2q4t type	5-7
2q4t-in-q-default	Ingress default queue of 2q4t type	0-4
1 Gigabit Module Egress: 1 strict priority queue and 3 normal queues with 4 thresholds per queue		
1p3q4t-out-pq1 [1]	Egress priority queue of 1p3q4t type	5-7
1p3q4t-out-q2	Egress queue 2 of 1p3q4t type	–

Table 2-3 System-Defined Type queuing Class Maps (continued)

Class Map Queue Name	Description	Default CoS Values
1p3q4t-out-q3	Egress queue 3 of 1p3q4t type	–
1p3q4t-out-q-default	Egress default queue of 1p3q4t type	0-4
10 Gigabit Module Ingress: 8 queues with 2 thresholds per queue		
8q2t-in-q1	Ingress queue 1 of 8q2t type	5-7
8q2t-in-q2	Ingress queue 2 of 8q2t type	–
8q2t-in-q3	Ingress queue 3 of 8q2t type	–
8q2t-in-q4	Ingress queue 4 of 8q2t type	–
8q2t-in-q5	Ingress queue 5 of 8q2t type	–
8q2t-in-q6	Ingress queue 6 of 8q2t type	–
8q2t-in-q7	Ingress queue 7 of 8q2t type	–
8q2t-in-q-default	Ingress default queue of 8q2t type	0-4
10 Gigabit Module Egress: 1 strict priority queue and 7 normal queues with 4 thresholds per queue		
1p7q4t-out-pq1 [1]	Egress priority queue of 1p7q4t type	5-7
1p7q4t-out-q2	Egress queue 2 of 1p7q4t type	–
1p7q4t-out-q3	Egress queue 3 of 1p7q4t type	–
1p7q4t-out-q4	Egress queue 4 of 1p7q4t type	–
1p7q4t-out-q5	Egress queue 5 of 1p7q4t type	–
1p7q4t-out-q6	Egress queue 6 of 1p7q4t type	–
1p7q4t-out-q7	Egress queue 7 of 1p7q4t type	–
1p7q4t-out-q-default	Egress default queue of 1p7q4t type	0-4

1. These are either priority or normal queues. If you use the priority keyword in your configuration, these are used as priority queues. Otherwise, they are used as normal queues.

Table maps that are defined by the system are listed in Table 2-4. The default mapping of values in the tables maps is contained in RFC 2597.

Table 2-4 System-Defined Table Maps

Table Map Name	Description
cir-markdown-map	Table map used to mark down packets that exceed the committed information rate (CIR). **Note** Enter the **show table-map** command to display the default mapping.
pir-markdown-map	Table map used to mark down packets that violate the peak information rate (PIR). **Note** Enter the **show table-map** command to display the default mapping.
cos-discard-class-map	Table map used to map the CoS value to the discard-class value.

Table 2-4 System-Defined Table Maps (continued)

Table Map Name	Description
cos-dscp-map	Table map used to map the CoS value to the DSCP value.
cos-precedence-map	Table map used to map the CoS value to the precedence value.
dscp-cos-map	Table map used to map the DSCP value to the CoS value.
dscp-precedence-map	Table map used to map the DSCP value to the precedence value.
dscp-discard-class-map	Table map used to map the DSCP value to the discard-class value.
precedence-dscp-map	Table map used to map the precedence value to the DSCP value.
precedence-cos-map	Table map used to map the precedence value to the CoS value.
precedence-discard-class-map	Table map used to map the precedence value to the discard-class value.
discard-class-cos-map	Table map used to map the discard-class value to the CoS value.
discard-class-prec-map	Table map used to map the discard-class value to the precedence value.
discard-class-dscp-map	Table map used to map the discard-class value to the DSCP value.

Policy maps that are defined by the system are listed in Table 2-5.

Configuring an MQC Object

When you specify a MQC object command, the device creates the object if it does not exist and then enters map mode.

To remove a class-map, table-map, or policy-map object, use the **no** form of the command that you used to create the object.

For the commands that you can use in the MQC object mode, see the following configuration chapters:

- Chapter 3, “Configuring Classification”
- Chapter 4, “Configuring Marking”
- Chapter 5, “Configuring Mutation Mapping”
- Chapter 6, “Configuring Policing”
- Chapter 7, “Configuring Queuing and Scheduling”

This section includes the following topics:

- Configuring or Modifying a Policy Map, page 2-15

Configuring or Modifying a Class Map

You can create or modify a class map. You can then reference class maps in policy maps.

Note You cannot create a queuing class map; you must use one of the system-defined queuing class maps listed in Table 2-3.

SUMMARY STEPS

1. **config t**
2. **class-map** [**type qos**] [**match-any** | **match-all**] *class-map-name*
3. **exit**
4. **class-map** [**type qos**] {**conform-color-in** | **conform-color-out** | **exceed-color-in** | **exceed-color-out**}
5. **exit**
6. **class-map type queuing match-any** *class-queuing-name*
7. **exit**
8. **show class-map [type qos]** [*class-map-name* | **conform-color-in** | **conform-color-out** | **exceed-color-in** | **exceed-color-out**]
9. **show class-map type queuing** [*class-queuing-name*]
10. **copy running-config startup-config**

DETAILED STEPS

	Command	Purpose
Step 1	**config t** **Example:** switch# config t switch(config)#	Enters configuration mode.
Step 2	**class-map** [**type qos**] [**match-any** | **match-all**] *class-map-name* **Example:** switch(config)# class-map class1 switch(config-cmap-qos)#	Creates or accesses the class map of type qos, and then enters class-map qos mode. Class-map names can contain alphabetic, hyphen, or underscore characters, are case sensitive, and can be up to 40 characters.
Step 3	**exit** **Example:** switch(config-cmap-qos)# exit switch(config)#	Exits class-map qos mode and enters configuration mode.
Step 4	**class-map** [**type qos**] {**conform-color-in** | **conform-color-out** | **exceed-color-in** | **exceed-color-out**} **Example:** switch(config)# class-map exceed-color-in switch(config-color-map)#	(Optional) Accesses the class map of type qos for one of the system-defined color maps, and then enters color-map mode. **Note** This is only used when color-aware policing is required.
Step 5	**exit** **Example:** switch(config-color-map)# exit switch(config)#	Exits color-map mode, and then enters configuration mode.
Step 6	**class-map type queuing match-any** [*class-queuing-name*] **Example:** switch(config)# class-map type queuing match-any 1p3q4t-out-pq1 switch(config-cmap-que)#	Creates or accesses the class map of type queuing, and then enters class-map queuing mode. Class queuing names are listed in Table 2-3.

	Command	Purpose

Step 7	**exit**
	
	Example:
	`switch(config-cmap-que)# exit`
	`switch(config)#`
		Exits class-map queuing mode and enters configuration mode.

Step 8	**show class-map** [**type qos**] [*class-map-name* | **conform-color-in** | **conform-color-out** | **exceed-color-in** | **exceed-color-out**] **Example:** `switch(config)# show class-map`	(Optional) Displays information about all configured class maps or a selected class map of type qos.
Step 9	**show class-map type queuing** [*class-queuing-name*] **Example:** `switch(config)# show class-map type queuing`	(Optional) Displays information about all configured class maps or a selected class map of type queuing. Class queuing names are listed in Table 2-3.
Step 10	**copy running-config startup-config** **Example:** `switch(config)# copy running-config startup-config`	(Optional) Saves the running configuration to the startup configuration.

Configuring or Modifying a Table Map

You can create or modify a table map that you can reference in policy maps. See Chapter 4, "Configuring Marking" for information on configuring table maps.

SUMMARY STEPS

1. **config t**
2. **table-map** *table-map-name*
3. **exit**
4. **table-map** {**cir-markdown-map** | **pir-markdown-map** | **cos-discard-class-map** | **cos-dscp-map** | **cos-precedence-map** | **dscp-cos-map** | **dscp-precedence-map** | **dscp-discard-class-map** | **precedence-dscp-map** | **precedence-cos-map** | **precedence-discard-class-map** | **discard-class-cos-map** | **discard-class-prec-map** | **discard-class-dscp-map**}
5. **exit**
6. **show table-map** [*table-map-name* | **cir-markdown-map** | **pir-markdown-map** | **cos-discard-class-map** | **cos-dscp-map** | **cos-precedence-map** | **dscp-cos-map** | **dscp-precedence-map** | **dscp-discard-class-map** | **precedence-dscp-map** | **precedence-cos-map** | **precedence-discard-class-map** | **discard-class-cos-map** | **discard-class-prec-map** | **discard-class-dscp-map**}
7. **copy running-config startup-config**

DETAILED STEPS

	Command	Purpose
Step 1	**config t** **Example:** switch# config t switch(config)#	Enters configuration mode.
Step 2	**table-map** *table-map-name* **Example:** switch(config)# table-map table1 switch(config-tmap)#	Creates or accesses the table map and then enters table-map mode. Table map names can contain alphabetic, hyphen, or underscore characters, are case sensitive, and can be up to 40 characters.
Step 3	**exit** **Example:** switch(config-tmap)# exit switch(config)#	Exits table-map mode and enters configuration mode.
Step 4	**table-map** {**cir-markdown-map** | **pir-markdown-map** | **cos-discard-class-map** | **cos-dscp-map** | **cos-precedence-map** | **dscp-cos-map** | **dscp-precedence-map** | **dscp-discard-class-map** | **precedence-dscp-map** | **precedence-cos-map** | **precedence-discard-class-map** | **discard-class-cos-map** | **discard-class-prec-map** | **discard-class-dscp-map**} **Example:** switch(config)# table-map cir-markdown-map switch(config-mrkdwn-map)#	Accesses one of the system-defined markdown table maps, and then enters markdown-map mode.
Step 5	**exit** **Example:** switch(config-mrkdwn-map)# exit switch(config)#	Exits table-map mode and enters configuration mode.
Step 6	**show table-map** [*table-map-name* | **cir-markdown-map** | **pir-markdown-map** | **cos-discard-class-map** | **cos-dscp-map** | **cos-precedence-map** | **dscp-cos-map** | **dscp-precedence-map** | **dscp-discard-class-map** | **precedence-dscp-map** | **precedence-cos-map** | **precedence-discard-class-map** | **discard-class-cos-map** | **discard-class-prec-map** | **discard-class-dscp-map**] **Example:** switch(config)# show table-map	(Optional) Displays information about all configured table maps or a selected table map.
Step 7	**copy running-config startup-config** **Example:** switch(config)# copy running-config startup-config	(Optional) Saves the running configuration to the startup configuration.

Configuring or Modifying a Policy Map

You can create or modify a policy map that you can use to define actions to perform on class maps.

SUMMARY STEPS

1. **config t**
2. **policy-map** [**type qos**] [**match-first**] *policy-map-name*
3. **exit**
4. **policy-map type queuing** [**match-first**] *policy-map-name*
5. **exit**
6. **show policy-map** [**type qos**] [*policy-map-name*]
7. **show policy-map type queuing** [*policy-map-name*]
8. **copy running-config startup-config**

DETAILED STEPS

	Command	Purpose

Step 1	**config t**
	
	Example:
	`switch# config t`
	`switch(config)#`
		Enters configuration mode.

Step 2	**policy-map** [**type qos**] [**match-first**] *policy-map-name*
	
	Example:
	`switch(config)# policy-map policy1`
	`switch(config-pmap-qos)#`
		Creates or accesses the policy map of type qos and then enters policy-map mode. Policy-map names can contain alphabetic, hyphen, or underscore characters, are case sensitive, and can be up to 40 characters.

Step 3	**exit**
	
	Example:
	`switch(config-tmap)# exit`
	`switch(config)#`
		Exits policy-map mode and enters configuration mode.

Step 4	**policy-map type queuing** [**match-first**] *policy-map-name*
	
	Example:
	`switch(config)# policy-map type queuing policy_queue1`
	`switch(config-pmap-que)#`
		Creates or accesses the policy map of type queuing and then enters policy-map mode. You can specify a policy-map name. Policy-map names can contain alphabetic, hyphen, or underscore characters, are case sensitive, and can be up to 40 characters.

	Command	Purpose
Step 5	**exit** **Example:** switch(config-tmap)# exit switch(config)#	Exits policy-map mode and enters configuration mode.
Step 6	**show policy-map** [**type qos**] [*policy-map-name*] **Example:** switch(config)# show policy-map	(Optional) Displays information about all configured policy maps or a selected policy map of type qos.
Step 7	**show policy-map type queuing** [*policy-map-name*] **Example:** switch(config)# show policy-map type queuing	(Optional) Displays information about all configured policy maps or a selected policy map of type queuing.
Step 8	**copy running-config startup-config** **Example:** switch(config)# copy running-config startup-config	(Optional) Saves the running configuration to the startup configuration.

Applying Descriptions to MQC Objects

You can apply the **description** command to any MQC object.

SUMMARY STEPS

1. **config t**
2. **class-map** [**type qos**] [**match-any** | **match-all**] *class-map-name*
   or
   table-map *table-map-name*
   or
   policy-map [**type qos**] [**match-first**] *policy-map-name*
3. **description** *string*
4. **exit**
5. **copy running-config startup-config**

DETAILED STEPS

	Command	Purpose
Step 1	**config t** **Example:** switch# config t switch(config)#	Enters configuration mode.
Step 2	**class-map** [**type qos**] [**match-any** | **match-all**] *class-map-name* **Example:** switch(config-cmap)# class-map class1 switch(config-cmap)#	Creates or accesses the class map, and then enters class-map mode. The class-map name can contain alphabetic, hyphen, or underscore characters, is case sensitive, and can be up to 40 alphanumeric characters.
	table-map *table-map-name* **Example:** switch(config-tmap)# table-map table1 switch(config-tmap)#	Creates or accesses the table map, and then enters table-map mode.The table-map name can contain alphabetic, hyphen, or underscore characters, is case sensitive, and can be up to 40 characters
	policy-map [**type qos**] [**match-first**] *policy-map-name* **Example:** switch(config)# policy-map policy1 switch(config-pmap)#	Creates or accesses the policy map, 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	**description** *string* **Example:** switch(config-cmap)# description my traffic class switch(config-cmap)#	Adds a description string to the MQC object. The description can be up to 200 alphanumeric characters. **Note** You cannot modify the description of system-defined queuing class maps.
Step 4	**exit** **Example:** switch(config-cmap)# exit switch(config)#	Exits table-map mode and enters configuration mode.
Step 5	**copy running-config startup-config** **Example:** switch(config)# copy running-config startup-config	(Optional) Saves the running configuration to the startup configuration.

Verifying an MQC Object

To display MQC object configuration information, perform one of the following tasks:

Command	Purpose
show class-map [**type qos**] [*class-map-name* | **conform-color-in** | **conform-color-out** | **exceed-color-in** | **exceed-color-out**]	Displays information about all configured class maps or a selected class map of type qos.
show class-map type queuing [*class-queuing-name*]	Displays information about all configured class maps or a selected class map of type queuing. Class queuing names are listed in Table 2-3.
show table-map [*table-map-name* | **cir-markdown-map** | **pir-markdown-map** | **cos-discard-class-map** | **cos-dscp-map** | **cos-precedence-map** | **dscp-cos-map** | **dscp-precedence-map** | **dscp-discard-class-map** | **precedence-dscp-map** | **precedence-cos-map** | **precedence-discard-class-map** | **discard-class-cos-map** | **discard-class-prec-map** | **discard-class-dscp-map**]	Displays information about all configured table maps or a selected table map.
show policy-map [**type qos**] [*policy-map-name*]	Displays information about all configured policy maps or a selected policy map of type qos.
show policy-map type queuing [*policy-map-name*]	Displays information about all configured policy maps or a selected policy map of type queuing.

For detailed information about the fields in the output from these commands, see the *Cisco NX-OS Quality of Service Command Reference*.

Attaching and Detaching a QoS Policy Action from an Interface

The software does not allow you to enable or disable QoS features with a configuration command. To enable or disable QoS features, you must attach or detach QoS policies to or from interfaces, VLANs, or tunnels as described in this section.

Note You must enable the **tunnel feature** by entering the feature tunnel command and configure the tunnel before you attach policies.

The system-defined type queuing class maps (see Table 2-3) are attached to each interface unless you specifically attach a different class map.

Note The device restricts QoS policies to one per interface per direction (ingress or egress) for each of the policy types qos and queuing.

Policies that are defined at multiple interfaces have the following restrictions:

- A QoS policy attached to the physical port will take effect when the port is not a member of a port channel.
- A QoS policy attached to a port channel will take effect even when policies are attached to member ports.
- A QoS policy attached to a VLAN interface is applied to all ports in that VLAN that do not have other policies specifically applied.
- One ingress policy type queuing is supported for each Layer 2 port- and Layer 2 port-channel interface in both the ingress and egress direction. Egress type qos policies are not allowed on Layer 2 port or Layer 2 port-channel interfaces.
- One ingress and one egress QoS policy are supported for each Layer 3 and Layer 3 port-channel interface.
- One ingress and one egress QoS policy are supported for each VLAN.
- One ingress and one egress queuing policy are supported for each Layer 2 port-, Layer 2 port-channel, Layer 3 port-, and Layer 3 port-channel interface.
- When a VLAN or port channel, or both, touches multiple forwarding engines, all policies that enforce a rate are enforced per forwarding engine.

  For example, a policer configured on a specific VLAN that limits the rate for the VLAN to 100 Mbps and has one switch port in the VLAN on one module and has another switch port in the VLAN on another module, each forwarding engine enforces the 100-Mbps rate. In this case, you could actually have up to 200 Mbps in the VLAN you configured to limit the rate to 100 Mbps.

Note Default queuing policies are active, unless you configure and apply another policy. See Table 2-5 for the default queuing policies.

The interface where a QoS policy is applied is summarized in Table 2-6. Each row represents the interface levels. The entry descriptions are as follows:

- Applied—Interface where an attached policy is applied.
- Present—Interface where a policy is attached but not applied.
- Not present—Interface where no policy is attached.
- Present or not—Interface where a policy is either attached or not, but not applied.

Table 2-6 QoS Policy Interfaces

Port Policy	Port-Channel Policy	VLAN Policy
Applied	Not present	Present or not
Present or not	Applied	Present or not
Not present	Not present	Applied

To attach a policy map to an interface, use the **service-policy** interface command mode or the VLAN command mode. You specify whether the policies defined in the policy map are applied to the input or output stream of packets on the interface.

To detach a policy map from an interface or VLAN, use the **no** form of the **service-policy** interface command mode or the VLAN command mode.

SUMMARY STEPS

1. **config t**
2. **interface** {[**ethernet** *slot/port*] | [**port-channel** *channel-number*] | [**vlan** *vlan-id*] | [**tunnel** *number*]}
3. **service-policy** [**type qos**] {**input** | **output**} *policy-map-name* [**no-stats**]
4. **show policy-map [interface** *interface* | **vlan** *vlan_id*] [**input** | **output**] [**type qos** | **queuing**] [**class** [**type qos** | **queuing**] *class-map-name*]
5. **copy running-config startup-config**

DETAILED STEPS

	Command	Purpose

Step 1	**config t**
	
	Example:
	`switch# config t`
	`switch(config)#`
		Enters configuration mode.

Step 2	**interface** {[**ethernet** *slot/port*]	[**port-channel** *channel-number*]	[**vlan** *vlan-id*]	[**tunnel** *number*]}
				
				Example:
				`switch(config)# interface ethernet 1/1`
				`switch(config-if)#`
					Enters interface mode on the interface specified.

Step 3	**service-policy** [**type qos**] {**input**	**output**} *policy-map-name* [**no-stats**]
		
		Example:
		`switch(config-if)# service-policy input policy1`
		`switch(config-if)#`
			Adds the policy map to the input or output packets of an interface or VLAN. Only one input policy and one output policy can be attached to an interface or VLAN.
			
			This example adds policy1 to the input interface.

Step 4	**show policy-map** [**interface** *interface*	**vlan** *vlan-id*] [**input**	**output**] [**type qos**	**queuing**] [**class** [**type qos**	**queuing**] *class-map-name*] **Example:** `switch(config)# show policy-map interface ethernet 1/1`	(Optional) Displays information about policy maps applied to all interfaces or the specified interface. You can limit what the device displays to input or output policies, qos or queuing polices, and to a specific class. This example shows all policy maps on the Ethernet 1/1 interface.
Step 5	**copy running-config startup-config** **Example:** `switch(config)# copy running-config startup-config`	(Optional) Saves the running configuration to the startup configuration.

Feature History for Using Modular QoS CLI

Table 2-7 lists the release history for this feature.

Table 2-7 Feature History for Modular QoS CLI

Feature Name	Releases	Feature Information
Tunnels	4.0(3)	Support was added for tunnels. You can now apply QoS policies to tunneled interfaces.
Type queuing default-in-policy	4.0(3)	Changed WRR from 50/50 to 80/20 for the type queuing default-in-policy only.
Modular QoS CLI	4.0(1)	This feature was introduced.

CHAPTER 3

Configuring Classification

This chapter describes how to configure classification on the device.

This chapter includes the following sections:

Information About Classification

Classification is the separation of packets into traffic classes. You configure the device to take specific action on the specified classified traffic, such as poling or marking down, or other actions.

You can create class maps to represent each traffic class by matching packet characteristics with the classification criteria in Table 3-1.

Table 3-1 Classification Criteria

Classification Criteria	Description
CoS	Class of Service (CoS) field in the IEEE 802.1Q header.
IP precedence	Precedence value within the Type of Service (TOS) byte of the IP header.
Differentiated Services Code Point (DSCP)	DSCP value within the DIffServ field of the IP header.
QoS group	Locally significant QoS values that can be manipulated and matched within the system. The range is from 0 to 126.
Discard class	Locally significant values that can be matched and manipulated within the system. The range is from 0 to 63.

Table 3-1 Classification Criteria (continued)

Classification Criteria	Description
ACL	IP ACL or MAC ACL name.
Protocol	Standard Layer 2 protocol such as Address Resolution Protocol (ARP) or Connectionless Network Service (CLNS).
Packet length	Size range of Layer 3 packet lengths.
IP RTP	Identify applications using Real-time Transport Protocol (RTP) by UDP port number range.
Class map	Criteria specified in a named class-map object.

You can specify multiple match criteria, you can choose to not match on a particular criterion, or you can determine traffic class by matching any or all criteria.

Note However, if you match on an ACL, no other match criteria, except packet length, can be specified in a match-all class. In a match-any class, you can match on ACLs and any other match criteria.

Some match criteria relate only to ingress or egress traffic. For example, the internal label QoS group has no meaning on ingress traffic because it has not yet been assigned a value.

Traffic that fails to match any class in a QoS policy map is assigned to a default class of traffic called class-default. The class class-default can be referenced in a QoS policy map to select this unmatched traffic.

You can reuse class maps within the same VDC when defining the QoS policies for different interfaces that process the same types of traffic.

Note See Chapter 2, “Using Modular QoS CLI” for more information on class maps.

Licensing Requirements for Classification

The following table shows the licensing requirements for this feature:

Product	License Requirement
NX-OS	QoS requires no license. Any feature not included in a license package is bundled with the Cisco NX-OS system images and is provided at no extra charge to you. For a complete explanation of the NX-OS licensing scheme, see the *Cisco Nexus 7000 Series NX-OS Licensing Guide, Release 4.0*.

However, using VDCs requires an Advanced Services license.

Prerequisites for Classification

Classification has the following prerequisites:

- You must be familiar with Chapter 2, "Using Modular QoS CLI."
- You are logged on to the switch.
- You are in the correct virtual device context (VDC). A VDC is a logical representation of a set of system resources. You can use the **switchto vdc** command with a VDC number.

Guidelines and Limitations

Classification has the following guidelines and limitations:

- You can specify a maximum of 1024 match criteria in a class map.
- You can configure a maximum of 4096 classes for use in a single policy map.
- When you match on an ACL, the only other match you can specify is the Layer 3 packet length in a match-all class.
- You can classify traffic on Layer 2 ports based on either the port policy or VLAN policy of the incoming packet, but not both. Either the port policy or the VLAN policy takes effect, but not both; if both are present, the device acts on the port policy and ignores the VLAN policy.

Configuring Traffic Classes

This section includes the following topics.

Configuring ACL Classification

Note The device does not support the **not** form of this command.

You can classify traffic by matching packets based on existing ACLs. The permit and deny ACL keywords are ignored in the matching. QoS does not use the permit-deny functions of ACLs.

Note Tunneled IP packets will not be matched unless the tunneling protocol is also IP, and then the match applies to the outer IP header and not the encapsulated IP header.

SUMMARY STEPS

1. **config t**
2. **class-map** [**type qos**] [**match-any** | **match-all**] *class-map-name*
3. **match access-group name** *acl-name*

DETAILED STEPS

	Command	Purpose
Step 1	**config t** **Example:** switch# config t switch(config)#	Enters configuration mode.
Step 2	**class-map** [**type qos**] [**match-any** | **match-all**] *class-map-name* **Example:** switch(config)# class-map class_acl	Creates or accesses the class map named *class-map-name*, and then enters class-map mode. Class map name can contain alphabetic, hyphen, or underscore characters, is case sensitive, and can be up to 40 characters.
Step 3	**match access-group name** *acl-name* **Example:** switch(config-cmap-qos)# match access-group name my_acl	Configures traffic class by matching packets based on *acl-name*. The permit and deny ACL keywords are ignored in the matching. **Note** The device does not support the **not** form of this command.

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

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

Configuring DSCP Classification

You can classify traffic based on the DSCP value in the DiffServ field of the IP header. The standard DSCP values are found in Table 3-2.

Table 3-2 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	AF40 dscp (011100)—decimal value 28

Table 3-2 Standard DSCP Values (continued)

Value	List of DSCP Values
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

Note Tunneled IP packets will not be matched unless the tunneling protocol is also IP, and then the match applies to the outer IP header and not the encapsulated IP header.

SUMMARY STEPS

1. **config t**
2. **class-map** [**type qos**] [**match-any** | **match-all**] *class-map-name*
3. **match** [**not**] **dscp** *dscp-list*
4. **exit**
5. **copy running-config startup-config**

`DETAILED STEPS

	Command	Purpose
Step 1	**config t** **Example:** switch# config t switch(config)#	Enters configuration mode.
Step 2	**class-map** [**type qos**] [**match-any** | **match-all**] *class-map-name* **Example:** switch(config)# class-map class_dscp	Creates or accesses the class map named *class-map-name*, and then enters class-map mode. The class-map name can contain alphabetic, hyphen, or underscore characters, is case sensitive, and can be up to 40 characters.
Step 3	**match** [**not**] **dscp** *dscp-list* **Example:** switch(config-cmap-qos)# match dscp af21, af32	Configures the traffic class by matching packets based on *dscp-values*. The standard DSCP values are shown in Table 3-2. Use the **not** keyword to match on values that do not match the specified range.
Step 4	**exit** **Example:** switch(config-cmap-qos)# exit switch(config)#	Exits class-map queuing mode, and enters configuration mode.
Step 5	**copy running-config startup-config** **Example:** switch(config)# copy running-config startup-config	(Optional) Saves the running configuration to the startup configuration.

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

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

Configuring IP Precedence Classification

You can classify traffic based on the precedence value in the Type of Service (TOS) byte field of the IP header. Table 3-3 shows the precedence values.

Table 3-3 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)

Table 3-3 Precedence Values (continued)

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

Note Tunneled IP packets will not be matched unless the tunneling protocol is also IP, and then the match applies to the outer IP header and not the encapsulated IP header.

SUMMARY STEPS

1. **config t**
2. **class-map [type qos] [match-any | match-all]** *class-map-name*
3. **match [not] precedence** *precedence-values*
4. **exit**
5. **copy running-config startup-config**

`DETAILED STEPS

	Command	Purpose

Step 1	**config t**
	
	Example:
	`switch# config t`
	`switch(config)#`
		Enters configuration mode.

Step 2	**class-map** [**type qos**] [**match-any** | **match-all**] *class-map-name* **Example:** `switch(config)# class-map class_ip_precedence`	Creates or accesses the class map named *class-map-name*, and then enters class-map mode. The class-map name can contain alphabetic, hyphen, or underscore characters, is case sensitive, and can be up to 40 characters.
Step 3	**match** [**not**] **precedence** *precedence-values* **Example:** `switch(config-cmap-qos)# match precedence 1-2, 5-7`	Configures the traffic class by matching packets based on *precedence-values*. Values are shown in Table 3-3. Use the **not** keyword to match on values that do not match the specified range.

Step 4	**exit**
	
	Example:
	`switch(config-cmap-qos)# exit`
	`switch(config)#`
		Exits class-map queuing mode and enters configuration mode.

Step 5	**copy running-config startup-config** **Example:** `switch(config)# copy running-config startup-config`	(Optional) Saves the running configuration to the startup configuration.

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

```
switch# show class-map class_ip_precedence
```

Configuring Protocol Classification

For Layer 3 protocol traffic, you can use the ACL classification match (see "Configuring ACL Classification" section on page 3-3).

You can classify traffic based on the protocol arguments described in Table 3-4.

Table 3-4 match Command Protocol Arguments

Argument	Description
arp	Address Resolution Protocol (ARP)
bridging	Bridging
cdp	Cisco Discovery Protocol (CDP)
clns	Connectionless Network Service (CLNS)
clns_es	CLNS End Systems
clns_is	CLNS Intermediate System
dhcp	Dynamic Host Configuration (DHCP)
isis	Intermediate system to intermediate system (IS-IS)
ldp	Label Distribution Protocol (LDP)
netbios	NetBIOS Extended User Interface (NetBEUI)

Note A maximum of eight different protocols (in Table 3-4) can be matched at one time.

SUMMARY STEPS

1. **config t**
2. **class-map** [**type qos**] [**match-any** | **match-all**] *class-map-name*
3. **match [not] protocol** {**arp** | **bridging** | **clns** | **clns_is** | **dhcp** | **isis** | **netbios** | **cdp** | **clns_es** | **ldp**}
4. **exit**
5. **copy running-config startup-config**

DETAILED STEPS

	Command	Purpose
Step 1	**config t** **Example:** switch# config t switch(config)#	Enters configuration mode.
Step 2	**class-map** [**type qos**] [**match-any** | **match-all**] *class-map-name* **Example:** switch(config)# class-map class_protocol	Creates or accesses the class map named *class-map-name*, and then enters class-map mode. The class-map name can contain alphabetic, hyphen, or underscore characters, is case sensitive, and can be up to 40 characters.
Step 3	**match** [**not**] **protocol** {**arp** | **bridging** | **cdp** | **clns** | **clns_is** | **dhcp** | **isis** | **netbios** | **clns_es** | **ldp**} switch(config-cmap-qos)# match protocol isis	Configures the traffic class by matching packets based on the specified protocol. Use the **not** keyword to match on protocols that do not match the protocol specified.
Step 4	**exit** **Example:** switch(config-cmap-qos)# exit switch(config)#	Exits class-map queuing mode and enters configuration mode.
Step 5	**copy running-config startup-config** **Example:** switch(config)# copy running-config startup-config	(Optional) Saves the running configuration to the startup configuration.

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

```
switch# show class-map class_protocol
```

Configuring QoS Group Classification

You can classify traffic based on the value of the QoS group internal label, which is not part of the packet payload or any packet header. You can set the value of the QoS group within a policy map using the **set qos-group** command as described in the "Configuring QoS Group Marking" section on page 4-8.

Note You match on the QoS group only in egress policies because its value is undefined until you set it in an ingress policy.

SUMMARY STEPS

1. **config t**
2. **class-map** [**type qos**] [**match-any** | **match-all**] *class-map-name*
3. **match** [**not**] **qos-group** *multi-range-qos-group-values*
4. **exit**
5. **copy running-config startup-config**

DETAILED STEPS

	Command	Purpose
Step 1	**config t** **Example:** switch# config t switch(config)#	Enters configuration mode.
Step 2	**class-map** [**type qos**] [**match-any** | **match-all**] *class-map-name* **Example:** switch(config)# class-map class_qos_group	Creates or accesses the class map named *class-map-name*, and then enters class-map mode. The class-map name can contain alphabetic, hyphen, or underscore characters, is case sensitive, and can be up to 40 characters.
Step 3	**match** [**not**] **qos-group** *multi-range-qos-group-values* **Example:** switch(config-cmap-qos)# match qos-group 4, 80-90	Configures the traffic class by matching packets based on a list of QoS group values. Values can range from 0 to 126. The default QoS group value is 0. Use the **not** keyword to match on values that do not match the specified range.
Step 4	**exit** **Example:** switch(config-cmap-qos)# exit switch(config)#	Exits class-map queuing mode and enters configuration mode.
Step 5	**copy running-config startup-config** **Example:** switch(config)# copy running-config startup-config	(Optional) Saves the running configuration to the startup configuration.

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

```
switch# show class-map class_qos_group
```

Configuring Discard Class Classification

You can classify traffic based on the value of the discard class internal label, which is not part of the packet payload or any packet header. You can set the value of the discard class within a policy map using the **set discard-class** command as described in the "Configuring Discard Class Marking" section on page 4-9.

Note You match on the discard class only in egress policies because its value is undefined until you set it in an ingress policy.

SUMMARY STEPS

1. **config t**
2. **class-map** [**type qos**] [**match-any** | **match-all**] *class-map-name*
3. **match** [**not**] **discard-class** *multi-range-discard-class-values*

4. **exit**
5. **copy running-config startup-config**

DETAILED STEPS

	Command	Purpose
Step 1	**config t** **Example:** switch# config t switch(config)#	Enters configuration mode.
Step 2	**class-map** [**type qos**] [**match-any** | **match-all**] *class-map-name* **Example:** switch(config)# class-map class_discard_class	Creates or accesses the class map named *class-map-name*, and then enters class-map mode. The class-map name can contain alphabetic, hyphen, or underscore characters, is case sensitive, and can be up to 40 characters.
Step 3	**match** [**not**] **discard-class** *multi-range-discard-class-values* **Example:** switch(config-cmap-qos)# match discard-class 4, 60-62	Configures the traffic class by matching packets based on the list of discard-class values. Values can range from 0 to 63. The default discard class value is 0. Use the **not** keyword to match on values that do not match the specified range.
Step 4	**exit** **Example:** switch(config-cmap-qos)# exit switch(config)#	Exits class-map queuing mode and enters configuration mode.
Step 5	**copy running-config startup-config** **Example:** switch(config)# copy running-config startup-config	(Optional) Saves the running configuration to the startup configuration.

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

```
switch# show class-map class_discard_class
```

Configuring Layer 3 Packet Length Classification

You can classify Layer 3 traffic based on various packet lengths.

Note This feature is designed for IP packets only.

SUMMARY STEPS

1. **config t**
2. **class-map** [**type qos**] [**match-any** | **match-all**] *class-map-name*
3. **match** [**not**] **packet length min** *packet-length-list*
4. **exit**

5. **copy running-config startup-config**

DETAILED STEPS

	Command	Purpose
Step 1	**config t** **Example:** switch# config t switch(config)#	Enters configuration mode.
Step 2	**class-map** [**type qos**] [**match-any** | **match-all**] *class-map-name* **Example:** switch(config)# class-map class_packet_length	Creates or accesses the class map named *class-map-name*, and then enters class-map mode. The class-map name can contain alphabetic, hyphen, or underscore characters, is case sensitive, and can be up to 40 characters.
Step 3	**match** [**not**] **packet length** *packet-length-list* **Example:** switch(config-cmap-qos)# match packet length 2000	Configures the traffic class by matching packets based on various packet lengths. Values can range from 1 to 9198. Use the **not** keyword to match on values that do not match the specified range.
Step 4	**exit** **Example:** switch(config-cmap-qos)# exit switch(config)#	Exits class-map queuing mode and enters configuration mode.
Step 5	**copy running-config startup-config** **Example:** switch(config)# copy running-config startup-config	(Optional) Saves the running configuration to the startup configuration.

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

```
switch# show class-map class_packet_length
```

Configuring CoS Classification

You can classify traffic based on 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.

SUMMARY STEPS

1. **config t**
2. **class-map** [**type qos**] [**match-any** | **match-all**] *class-map-name*
3. **match** [**not**] **cos** *cos-list*
4. **exit**
5. **copy running-config startup-config**

DETAILED STEPS

	Command	Purpose
Step 1	**config t** **Example:** switch# config t switch(config)#	Enters configuration mode.
Step 2	**class-map** [**type qos**] [**match-any** | **match-all**] *class-map-name* **Example:** switch(config)# class-map class_cos	Creates or accesses the class map named *class-map-name*, and then enters class-map mode. The class-map name can contain alphabetic, hyphen, or underscore characters, is case sensitive, and can be up to 40 characters.
Step 3	**match** [**not**] **cos** *cos-list* **Example:** switch(config-cmap-qos)# match cos 4, 5-6	Configures the traffic class by matching packets based on list of CoS values. Values can range from 0 to 7. Use the **not** keyword to match on values that do not match the specified range.
Step 4	**exit** **Example:** switch(config-cmap-qos)# exit switch(config)#	Exits class-map queuing mode and enters configuration mode.
Step 5	**copy running-config startup-config** **Example:** switch(config)# copy running-config startup-config	(Optional) Saves the running configuration to the startup configuration.

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

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

Configuring IP RTP Classification

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 uses an even port and the next higher odd port is used for RTP Control Protocol (RTCP) communications.

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

SUMMARY STEPS

1. **config t**
2. **class-map** [**type qos**] [**match-any** | **match-all**] *class-map-name*
3. **match** [**not**] **ip rtp** *udp-port-values*
4. **exit**
5. **copy running-config startup-config**

DETAILED STEPS

	Command	Purpose
Step 1	**config t** **Example:** switch# config t switch(config)#	Enters configuration mode.
Step 2	**class-map** [**type qos**] [**match-any** | **match-all**] *class-map-name* **Example:** switch(config)# class-map class_rtp	Creates or accesses the class map named *class-map-name*, and then enters class-map mode. The class-map name can contain alphabetic, hyphen, or underscore characters, is case sensitive, and can be up to 40 characters.
Step 3	**match** [**not**] **ip rtp** *udp-port-value* **Example:** switch(config-cmap-qos)# match ip rtp 2000-2100, 4000-4100	Configures the traffic class by matching packets based on range of lower and upper UDP port numbers, which is likely to target applications using RTP. Values can range from 2000 to 65535. Use the **not** keyword to match on values that do not match the specified range.
Step 4	**exit** **Example:** switch(config-cmap-qos)# exit switch(config)#	Exits class-map queuing mode and enters configuration mode.
Step 5	**copy running-config startup-config** **Example:** switch(config)# copy running-config startup-config	(Optional) Saves the running configuration to the startup configuration.

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

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

Configuring Class Map Classification

You can classify traffic based on the match criteria in another class map. You can reference the same class map in multiple policies.

Note

- The referenced class map must be created prior to its reference.
- You can configure only one level of nesting of class maps. You cannot reference a class map that references another class map.

Use the following guidelines to configure class-map classification:

- To perform a logical OR with the class map specified in the **match class-map** command, use the **match-any** keyword. The **match-any** or **match-all** specification of the matched class map is ignored.

- To perform a logical AND with the class map specified in the **match class-map** command, use the **match-all** keyword. The **match-any** or **match-all** specification of the matched class map is ignored.
- Before you delete a referenced class map, you should delete all references to that class map.

SUMMARY STEPS

1. **config t**
2. **class-map** [**type qos**] [**match-any** | **match-all**] *class-map-name*
3. **match** [**not**] **class-map** *class-map-name*
4. **exit**
5. **copy running-config startup-config**

DETAILED STEPS

	Command	Purpose
Step 1	**config t** **Example:** switch# config t switch(config)#	Enters configuration mode.
Step 2	**class-map** [**type qos**] [**match-any** | **match-all**] *class-map-name* **Example:** switch(config)# class-map class_class_map	Creates or accesses the class map named *class-map-name*, and then enters class-map mode. The class-map name can contain alphabetic, hyphen, or underscore characters, is case sensitive, and can be up to 40 characters.
Step 3	**match** [**not**] **class-map** *class-map-name* **Example:** switch(config-cmap-qos)# match class-map class_map3	Configures the traffic class by matching packets based on match criteria in another class map. Because match-all is the default for the **class-map** command, match criteria specified in class_map3 are ANDed with match criteria in class_class_map. Use the **not** keyword to match on values that do not match the specified range.
Step 4	**exit** **Example:** switch(config-cmap-qos)# exit switch(config)#	Exits class-map queuing mode and enters configuration mode.
Step 5	**copy running-config startup-config** **Example:** switch(config)# copy running-config startup-config	(Optional) Saves the running configuration to the startup configuration.

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

```
switch# show class-map class_class_map
```

Verifying Classification Configuration

Use the **show class-map** command to verify the class-map configuration. This command displays all class maps.

```
switch# show class-map
...
```

Example Configuration

The following example shows how to configure classification for two classes of traffic:

```
class-map class_dscp
  match dscp af21, af32
exit
class-map class_cos
  match cos 4, 5-6
  exit
```

CHAPTER 4

Configuring Marking

This chapter describes how to configure the marking features that you can use to define the class of traffic to which the packet belongs to.

This chapter includes the following sections:

Information About Marking

Marking is a method that you use to modify the QoS fields of the incoming and outgoing packets. The QoS fields that you can mark are CoS in Layer 2, and IP precedence and DSCP in Layer 3. QoS group and discard class are two labels local to the system that you can assign intermediate marking values, which you can then use to determine the final values marked in a packet.

You can use marking commands in traffic classes that are referenced in a policy map. The marking features that you can configure are listed in Table 4-1.

Table 4-1 Configurable Marking Features

Marking Feature	Description
DSCP	Layer 3 Differentiated Service Code Point (DSCP). **Note** If you manipulate this **dscp** value, you cannot manipulate **discard class** values, and vice-versa.
IP precedence	Layer 3 IP precedence. **Note** IP precedence uses only the lower 3 bits of the Type of Service (TOS) field. The device overwrites the first 3 bits of the TOS field to 0.

Table 4-1 Configurable Marking Features (continued)

Marking Feature	Description
CoS	Layer 2 Class of Service (CoS).
QoS group	Locally significant QoS values that can be manipulated and matched within the system. The range is from 0 to 126.
Discard class	Locally significant values that can be matched and manipulated within the system. The range is from 0 to 63. **Note** If you manipulate this **discard class** value, you cannot manipulate **dscp** values, and vice-versa.
Ingress and egress ports	Status of the marking applies to incoming or outgoing packets.
Using table maps	Method to use table maps for marking.

Unless noted as a restriction, you can apply marking features to both incoming and outgoing packets.

Licensing Requirements for Marking

The following table shows the licensing requirements for this feature:

Product	License Requirement
NX-OS	QoS requires no license. Any feature not included in a license package is bundled with the Cisco NX-OS system images and is provided at no extra charge to you. For a complete explanation of the NX-OS licensing scheme, see the *Cisco Nexus 7000 Series NX-OS Licensing Guide, Release 4.0*.

However, using VDCs requires an Advanced Services license.

Prerequisites for Marking

Marking has the following prerequisites:

- You must be familiar with Chapter 2, “Using Modular QoS CLI.”
- You are logged on to the switch.
- You are in the correct virtual device context (VDC). A VDC is a logical representation of a set of system resources. You can use the **switchto vdc** command with a VDC number.

Guidelines and Limitations

Use the following guidelines to configure marking:

- The **set cos** command is applicable only to 802.1Q interfaces, and you can only use it in egress policies.
- You can only use the **set qos-group** command in ingress policies.
- You can only use the **set discard-class** command in ingress policies.

Configuring Marking

You can combine one or more of the marking features in a policy map to control the setting of QoS values. You can then apply policies to either incoming or outgoing packets on an interface.

This section includes the following topics:

Note Do not press Enter after you use the **set** command and before you add the rest of the command. If you press Enter directly after entering the **set** keyword, you will be unable to continue to configure with the QoS configuration.

Configuring DSCP Marking

Note If you configure this value, you cannot configure the discard-class value (see the "Configuring Discard Class Marking" section on page 4-9).

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, as well as the standard DSCP values shown in Table 4-2.

Table 4-2 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	AF40 dscp (011100)—decimal value 28
af33	AF33 dscp (011110)—decimal value 30

Table 4-2 Standard DSCP Values (continued)

Value	List of DSCP Values
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

For more information about DSCP, see RFC 2475.

SUMMARY STEPS

1. **config t**
2. **policy-map** [**type qos**] [**match-first**] *policy-map-name*
3. **class** [**type qos**] {*class-map-name* | **class-default**} [**insert-before** *before-class-map-name*]
4. **set dscp** *dscp-value*

DETAILED STEPS

	Command	Purpose
Step 1	**config t** **Example:** switch# config t switch(config)#	Enters configuration mode.
Step 2	**policy-map** [**type qos**] [**match-first**] *policy-map-name* **Example:** switch(config)# policy-map policy1 switch(config-pmap-qos)#	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**} [**insert-before** *before-class-map-name*] **Example:** switch(config-pmap)# class class1 switch(config-pmap-c-qos)#	Creates a reference to *class-map-name*, and enters policy-map class configuration mode. The class is added to the end of the policy map unless **insert-before** is used to specify the class to insert before. Specify **class-default** to select all traffic not currently matched by classes in the policy map.
Step 4	**set dscp** *dscp-value* **Example:** switch(config-pmap-c-qos)# set dscp af31 switch(config-pmap-c-qos)#	Sets the DSCP value to *dscp-value*. Standard values are shown in Table 4-2.

Use the **show policy-map** command to display the policy-map configuration:

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

Configuring IP Precedence Marking

You can set the value of the IP precedence field in bits 0–2 of the IPv4 Type of Service (ToS) field of the IP header.

Note The device rewrites the last 3 bits of the ToS field to 0 for packets that match this class.

Table 4-3 shows the precedence values.

Table 4-3 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)

Table 4-3 Precedence Values (continued)

Value	List of Precedence Values
immediate	Immediate precedence (2)
internet	Internetwork control precedence (6)
network	Network control precedence (7)
priority	Priority precedence (1)
routine	Routine precedence (0)

SUMMARY STEPS

1. **config t**
2. **policy-map** [**type qos**] [**match-first**] *policy-map-name*
3. **class** [**type qos**] {*class-map-name* | **class-default**} [**insert-before** *before-class-map-name*]
4. **set precedence** *precedence-value*

DETAILED STEPS

	Command	Purpose

Step 1	**config t**
	
	Example:
	`switch# config t`
	`switch(config)#`
		Enters configuration mode.

Step 2	**policy-map** [**type qos**] [**match-first**] *policy-map-name*
	
	Example:
	`switch(config)# policy-map policy1`
	`switch(config-pmap-qos)#`
		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**} [**insert-before** *before-class-map-name*]
	
	Example:
	`switch(config-pmap-qos)# class class1`
	`switch(config-pmap-c-qos)#`
		Creates a reference to *class-map-name* and enters policy-map class configuration mode. The class is added to the end of the policy map unless **insert-before** is used to specify the class to insert before. Specify **class-default** to select all traffic not currently matched by classes in policy map.

Step 4	**set precedence** *precedence-value*
	
	Example:
	`switch(config-pmap-c-qos)# set precedence 3`
	`switch(config-pmap-c-qos)#`
		Sets the IP precedence value to *precedence-value*. The value can range from 0 to 7. You can enter one of the values shown in Table 4-3.

Use the **show policy-map** command to display the policy-map configuration:

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

Configuring CoS Marking

You can set the value of the CoS field in the high-order three bits of the VLAN ID Tag field in the IEEE 802.1Q header.

Note You can set CoS only in egress policies.

SUMMARY STEPS

1. **config t**
2. **policy-map** [**type qos**] [**match-first**] *policy-map-name*
3. **class** [**type qos**] {*class-map-name* | **class-default**} [**insert-before** *before-class-map-name*]
4. **set cos** *cos-value*

DETAILED STEPS

	Command	Purpose
Step 1	**config t** **Example:** switch# config t switch(config)#	Enters configuration mode.
Step 2	**policy-map** [**type qos**] [**match-first**] *policy-map-name* **Example:** switch(config)# policy-map policy1 switch(config-pmap-qos)#	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**} [**insert-before** *before-class-map-name*] **Example:** switch(config-pmap-qos)# class class1 switch(config-pmap-c-qos)#	Creates a reference to *class-map-name*, and enters policy-map class configuration mode. The class is added to the end of the policy map unless **insert-before** is used to specify the class to insert before. Specify **class-default** to select all traffic not currently matched by classes in the policy map.
Step 4	**set cos** *cos-value* **Example:** switch(config-pmap-c-qos)# set cos 3 switch(config-pmap-c-qos)#	Sets the CoS value to *cos-value*. The value can range from 0 to 7. You can use this command only in egress policies.

Use the **show policy-map** command to display the policy-map configuration:

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

Configuring QoS Group Marking

You can set the value of the internal label QoS group, which is only locally significant. You can reference this value in subsequent policy actions or classify traffic that is referenced in egress policies by using the **match qos-group** class-map command.

Note You can set QoS group only in ingress policies.

SUMMARY STEPS

1. **config t**
2. **policy-map** [**type qos**] [**match-first**] *policy-map-name*
3. **class** [**type qos**] {*class-map-name* | **class-default**} [**insert-before** *before-class-map-name*]
4. **set qos-group** *qos-group-value*

DETAILED STEPS

	Command	Purpose

Step 1	**config t**
	
	Example:
	`switch# config t`
	`switch(config)#`
		Enters configuration mode.

Step 2	**policy-map** [**type qos**] [**match-first**] *policy-map-name*
	
	Example:
	`switch(config)# policy-map policy1`
	`switch(config-pmap-qos)#`
		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**} [**insert-before** *before-class-map-name*]
	
	Example:
	`switch(config-pmap-qos)# class class1`
	`switch(config-pmap-c-qos)#`
		Creates a reference to *class-map-name*, and enters policy-map class configuration mode. The class is added to the end of the policy map unless **insert-before** is used to specify the class to insert before. Specify **class-default** to select all traffic not currently matched by classes in the policy map.

Step 4	**set qos-group** *qos-group-value*
	
	Example:
	`switch(config-pmap-c-qos)# set qos-group 100`
	`switch(config-pmap-c-qos)#`
		Sets the QoS group value to *qos-group-value*. The value can range from 0 to 126.

Use the **show policy-map** command to display the policy-map configuration:

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

Configuring Discard Class Marking

Note If you configure this value, you cannot configure the DSCP value (see the "Configuring DSCP Marking" section on page 4-3).

You can set the value of the internal label discard class, which is locally significant only. You can reference this value in subsequent policy actions or classify traffic that is referenced in egress policies by using the **match discard-class** class-map command.

Note You can set the discard class only in ingress policies.

SUMMARY STEPS

1. **config t**
2. **policy-map** [**type qos**] [**match-first**] *policy-map-name*
3. **class** [**type qos**] {*class-map-name* | **class-default**} [**insert-before** *before-class-map-name*]
4. **set discard-class** *discard-class-value*

DETAILED STEPS

	Command	Purpose
Step 1	**config t** **Example:** switch# config t switch(config)#	Enters configuration mode.
Step 2	**policy-map** [**type qos**] [**match-first**] *policy-map-name* **Example:** switch(config)# policy-map policy1 switch(config-pmap-qos)#	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**} [**insert-before** *before-class-map-name*] **Example:** switch(config-pmap-qos)# class class1 switch(config-pmap-c-qos)#	Creates a reference to *class-map-name*, and enters policy-map class configuration mode. The class is added to the end of the policy map unless **insert-before** is used to specify the class to insert before. Specify **class-default** to select all traffic not currently matched by classes in the policy map.
Step 4	**set discard-class** *discard-class-value* **Example:** switch(config-pmap-c-qos)# set discard-class 40 switch(config-pmap-c-qos)#	Sets the discard class value to *discard-class-value*. The value can range from 0 to 63. **Note** See "Configuring Marking Using Table Maps" section on page 4-13 for information on using table maps with marking.

Use the **show policy-map** command to display the policy-map configuration:

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

Configuring Ingress and Egress Marking

You can apply the marking instructions in a QoS policy map to ingress or egress packets by attaching that QoS policy map to an interface. To select ingress or egress, you specify either the **input** or **output** keyword in the **service-policy** command. For detailed instructions, see the "Attaching and Detaching a QoS Policy Action from an Interface" section on page 2-18.

Configuring DSCP Port Marking

You can set the DSCP value for each class of traffic defined in a specified ingress policy map.

The default behavior of the device is to preserve the DSCP value, or to trust DSCP. To make the port untrusted, change the DSCP value. Unless you configure a QoS policy and attach that policy to specified interfaces, the DSCP value is preserved.

Note You can attach only one policy type qos map to each interface in each direction.

SUMMARY STEPS

1. **config t**
2. **policy-map** [**type qos**] [**match-first**] *policy-map-name*
3. **class** [**type qos**] {*class-map-name* | **class-default**} [**insert-before** *before-class-map-name*]
4. **set dscp**-*value*
5. **exit**
6. **class** [**type qos**] {*class-map-name* | **class-default**} [**insert-before** *before-class-map-name*]
7. **set dscp**-*value*
8. **exit**
9. **class** [**type qos**] {*class-map-name* | **class-default**} [**insert-before** *before-class-map-name*]
10. **set dscp**-*value*
11. **exit**
12. **exit**
13. **{[interface ethernet** *slot/port*] | vlan-*id*]}
14. **service-policy** [**type qos**] {**input** | **output**} *policy-map-name* [**no-stats**]

DETAILED STEPS

	Command	Purpose
Step 1	**config t** **Example:** switch# config t switch(config)#	Enters configuration mode.
Step 2	**policy-map** [**type qos**] [**match-first**] *policy-map-name* **Example:** switch(config)# policy-map policy1 switch(config-pmap-qos)#	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**} [**insert-before** *before-class-map-name*] **Example:** switch(config-pmap)# class class1 switch(config-pmap-c-qos)#	Creates a reference to *class-map-name* and enters policy-map class configuration mode. The class is added to the end of the policy map unless **insert-before** is used to specify the class to insert before. Specify **class-default** to select all traffic not matched by classes in the policy map so far.
Step 4	**set dscp**-*value* **Example:** switch(config-pmap-c-qos)# set dscp af31 switch(config-pmap-c-qos)#	Sets the DSCP value to *dscp-value*. Valid values are shown in Table 4-2.
Step 5	exit **Example:** switch(config-pmap-c-qos)# exit switch(config-pmap-qos)#	Returns to policy-map configuration mode.
Step 6	**class** [**type qos**] {*class-map-name* | **class-default**} [**insert-before** *before-class-map-name*] **Example:** switch(config-pmap-qos)# class class2 switch(config-pmap-c-qos)#	Creates a reference to *class-map-name*, and enters policy-map class configuration mode. The class is added to the end of the policy map unless **insert-before** is used to specify the class to insert before. Specify **class-default** to select all traffic not currently matched by classes in the policy map.
Step 7	**set dscp**-*value* **Example:** switch(config-pmap-c-qos)# set dscp af13 switch(config-pmap-c-qos)#	Sets the DSCP value to *dscp-value*. Valid values are shown in Table 4-2.
Step 8	exit **Example:** switch(config-pmap-c-qos)# exit switch(config-pmap-qos)#	Returns to policy-map configuration mode.
Step 9	**class** [**type qos**] {*class-map-name* | **class-default**} [**insert-before** *before-class-map-name*] **Example:** switch(config-pmap-qos)# class class-default switch(config-pmap-c-qos)#	Creates a reference to *class-map-name*, and enters policy-map class configuration mode. The class is added to the end of the policy map unless **insert-before** is used to specify the class to insert before. Specify **class-default** to select all traffic not currently matched by classes in policy map.

	Command	Purpose
Step 10	**set dscp**-*value* **Example:** switch(config-pmap-c-qos)# set dscp af22 switch(config-pmap-c-qos)#	Sets the DSCP value to *dscp-value*. Valid values are shown in Table 4-2.
Step 11	**exit** **Example:** switch(config-pmap-c-qos)# exit switch(config-pmap-qos)#	Returns to policy-map configuration mode.
Step 12	**exit** **Example:** switch(config-pmap)# exit switch(config)#	Returns to configuration mode.
Step 13	**interface ethernet** *slot/port* **Example:** switch(config)# interface ethernet 1/1 switch(config-if)#	Enters interface mode to configure the Ethernet interface.
	vlan-*id* **Example:** switch(config)# vlan 101 switch(config-if)#	(Optional) Enters the VLAN mode on the specified VLAN.
Step 14	**service-policy** [**type qos**] {**input** | **output**} *policy-map-name* [**no-stats**] **Example:** switch(config-if)# service-policy input policy1 switch(config-if)#	Adds *policy-map-name* to the input packets of the interface. You can attach only one input policy and one output policy to an interface.

Use the **show policy-map** command to display the policy-map configuration:

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

Configuring Table Maps for Use in Marking

You can configure the system-defined table maps to define the mapping of values from a source QoS field to a destination QoS field. The source and destination fields are determined by the context of the table map in the **set** and **police** commands. For information about table maps, see the "Configuring Marking Using Table Maps" section on page 4-13.

Use the **default** command to define the destination value of unmapped source values. By default, unmapped values are copied to the destination value, so that the destination value is the same as the source value. In Cisco NX-OS Release 4.0.2 and later releases, the *ignore* variable for the **default** command is no longer supported.

Note You can use only one of the system-defined, table maps in this procedure. For information on the system-defined table maps, see Chapter 2, "Using Modular QoS CLI."

SUMMARY STEPS

1. **config t**
2. **table-map** *table-map-name*
3. **from** *source-value* **to** *dest-value*
4. Repeat Step 3 to map other values.
5. **default** {*value* | **copy**}
6. **exit**

DETAILED STEPS

	Command	Purpose

Step 1	**config t**
	
	Example:
	`switch# config t`
	`switch(config)#`
		Enters configuration mode.

Step 2	**table-map** *table-map-name*
	
	Example:
	`switch(config)# table-map cos-dscp-map`
	`switch(config-tmap)#`
		Accesses the default, or system-defined, table map named *table-map-name*, and then enters table-map mode.

Step 3	**from** *source-value* **to** *dest-value*
	
	Example:
	`switch(config-tmap)# from 0 to 2`
	`switch(config-tmap)#`
		Defines a mapping from *source-value* to *dest-value*. Up to 64 mappings can be defined. The mapping values can be from 0 to 63.

Step 4	Repeat Step 3 to define more values.	—

Step 5	**default** {*value* | **copy**}
	
	Example:
	`switch(config-tmap)# default 18`
	`switch(config-tmap)#`
		Defines the default value to use for unmapped source values. The *value* must be in the range 0 to 63. Entering *copy* specifies the default copy action.
		Note In Cisco NX-OS 4.0.2 and later releases, the *ignore* variable for this is not supported.

Step 6	**exit**
	
	Example:
	`switch(config-tmap)# exit`
	`switch(config)#`
		Exits table-map mode and enters configuration mode.

Use the **show table-map** command to display the table-map configuration:

```
switch# show table-map cos-dscp-map
```

Configuring Marking Using Table Maps

You can use the system-defined table maps to perform marking in the **set** and **police** policy map class commands.

Note See Chapter 2, "Using Modular QoS CLI" for the list of system-defined table maps.

A source field and destination field are specified in the command that maps to the source and destination values supplied in the referenced table map. The QoS fields that can be used in these commands are listed in Table 4-4.

Table 4-4 QoS Table Map Fields

QoS Table Map Field	Description
CoS	Class of Service field in the 802.1Q header.
DSCP	Differentiated Services Code Point in the IP header.
IP precedence	Bits 0–2 of the IPv4 ToS field.
Discard class	Locally significant values that can be matched and manipulated within the system. The range is from 0 to 63.

You can use the system-defined markdown table maps for the **exceed** or **violate** action of the **police** command by using the same syntax as the **set** command.

Note
- The internal label QoS group is not supported through table maps.
- Marking down in the **police** command requires the use of a table map.

For information on the **police** command, see Chapter 6, "Configuring Policing."

SUMMARY STEPS

1. **config t**
2. **policy-map** [**type qos**] [**match-first**] *policy-map-name*
3. **class** [**type qos**] {*class-map-name* | **class-default**} [**insert-before** *before-class-map-name*]
4. **set** {**cos** | **dscp** | **discard-class** | **precedence** | **discard-class**} {**cos** | **dscp** | **discard-class** | **precedence** | **discard-class**} *table-map-name*
5. **exit**

DETAILED STEPS

	Command	Purpose
Step 1	**config t** **Example:** switch# config t switch(config)#	Enters configuration mode.
Step 2	**policy-map** [**type qos**] [**match-first**] *policy-map-name* **Example:** switch(config)# policy-map policy1 switch(config-pmap-qos)#	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**} [**insert-before** *before-class-map-name*] **Example:** switch(config-pmap-qos)# class class1 switch(config-pmap-c-qos)#	Creates a reference to *class-map-name* and enters policy-map class configuration mode. The class is added to the end of the policy map unless **insert-before** is used to specify the class to insert before. Specify **class-default** to select all traffic not currently matched by classes in the policy map.
Step 4	**set** {**cos** | **dscp** | **discard-class** | **precedence** | **discard-class**} {**cos** | **dscp** | **discard-class**| **precedence** | **discard-class**} *table-map-name* **Example:** switch(config-pmap-c-qos)# set cos dscp cos-dscp-map switch(config-pmap-c-qos)#	Sets the first packet field to the value of the second packet field based on the mapping values specified in the referenced *table-map-name*. **Note** The *table-map-name* must be the name of one of the system-defined table maps listed in Chapter 2, "Using Modular QoS CLI." You cannot use the name of a user-defined table in this procedure. The example shows that CoS is replaced by DSCP based on the system-defined cos-dscp-map.
Step 5	**exit** **Example:** switch(config-pmap-c)# exit switch(config-pmap-qos)#	Returns to policy-map configuration mode.

Use the **show policy-map** and **show table-map cos-dscp-map** command to display the policy1 policy-map configuration:

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

Verifying the Marking Configuration

Use the **show table-map** and **show policy-map** commands to verify the marking configuration.

This command displays all table maps:

```
switch# show table-map
...
```

This command displays all policy maps:

```
switch# show policy-map
...
```

Example Configuration

The following example shows how to configure marking:

```
config t
  policy-map type qos untrust_dcsp
   class class-default
     set dscp 0
  policy-map type queuing untrust_1Gport_policy
   class type queuing 2q4t-in-q-default
     set cos 0
  policy-map type queuing untrust_10Gport_policy
   class type queuing 8q2t-in-q-default
     set cos 0
```

Feature History for Marking

Table 4-5 lists the release history for this feature.

Table 4-5 Feature History for Marking

Feature Name	Releases	Feature Information
Default command	4.0(2)	The *ignore* variable is no longer supported for the **default** command.
QoS Marking	4.0(1)	This feature was introduced.

CHAPTER 5

Configuring Mutation Mapping

This chapter describes how to configure the mutation of packet values used to define traffic classes.

This chapter includes the following sections:

Information About Mutation Mapping

Mutation mapping is a method of modifying a QoS field in all packets on an interface. On ingress, mutation mapping occurs before traffic classification and all other actions. On egress, mutation mapping occurs after traffic classification and before the other actions. You can apply mutation mapping to packet fields CoS, DSCP, or IP precedence, or to the internal field discard class.

You use a hierarchical policy map to configure mutation mapping. In the mutation mapping policy map you specify the field to mutate and the policy map to apply with the mutation.

Note The device supports hierarchical policies only for mutation mapping.

Mutation Mapping in Sequence of Traffic Actions

The sequence of QoS actions on ingress traffic is as follows:

1. Queuing and scheduling
2. Mutation
3. Classification
4. Marking
5. Policing

The sequencing of QoS actions on egress traffic is as follows:

1. Classification
2. Marking
3. Policing
4. Mutation
5. Queuing and scheduling

Note Mutation happens much closer to the beginning of the traffic actions on the ingress packets, and any further classification and policing is based on the changed QoS values. Mutation happens at the end of the traffic actions on the egress packets, right before queuing and scheduling.

Licensing Requirements for Mutation Mapping

The following table shows the licensing requirements for this feature:

Product	License Requirement
NX-OS	QoS requires no license. Any feature not included in a license package is bundled with the Cisco NX-OS system images and is provided at no extra charge to you. For a complete explanation of the NX-OS licensing scheme, see the *Cisco Nexus 7000 Series NX-OS Licensing Guide, Release 4.0*.

However, using VDCs requires an Advanced Services license.

Prerequisites for Mutation Mapping

Mutation mapping has the following prerequisites:

- You must be familiar with Chapter 2, “Using Modular QoS CLI.”
- You are logged on to the switch.
- You are in the correct virtual device context (VDC). A VDC is a logical representation of a set of system resources. You can use the **switchto vdc** command with a VDC number.

Guidelines and Limitations

Use the following guidelines to configure mutation mapping:

- You use a hierarchical policy for mutation mapping. Hierarchical policies are not supported for any other use.
- The device supports only one level of hierarchy.
- You can configure up to 14 table maps for use in ingress interfaces and up to 15 table maps for use in egress interfaces.
- Before you delete a referenced policy map, you must first remove all references to that policy map.

Configuring Mutation Mapping

To configure mutation mapping, you create a hierarchical policy map that uses the **class-default** traffic class to capture all packets and apply mutation mapping to them. You use the **service-policy** command to specify the policy map to apply with mutation mapping.

To configure mutation mapping, follow these steps:

Step 1 Create the policy map to apply in the mutation mapping hierarchical policy. For information about configuring policy maps, see Chapter 6, "Configuring Policing" or Chapter 7, "Configuring Queuing and Scheduling."

Step 2 Create the table map to use in the mutation mapping hierarchical policy. For information about configuring table maps, see the "Configuring Marking Using Table Maps" section on page 4-13.

Step 3 Configure the mutation mapping hierarchical policy as described in this section.

Step 4 Apply the service policy to the interface. For information about attaching policies to interfaces, see Chapter 2, "Using Modular QoS CLI."

SUMMARY STEPS

1. **config t**
2. **policy-map** [**type qos**] [**match-first**] *policy-map-name*
3. **class class-default**
4. **set** {**cos** | **discard-class** | **dscp** | **precedence**} {**cos** | **discard-class** | **dscp** | **precedence**} **table** *table-map-name*
5. **service-policy** [**type qos**] *policy-map-name* [**no-stats**]
6. **show policy-map** [**type** {**qos** | **queuing**}] [*policy-map-name*]
7. **copy running-config startup-config**

DETAILED STEPS

	Command	Purpose
Step 1	**config t** **Example:** switch# config t switch(config)#	Enters configuration mode.
Step 2	**policy-map** [**type qos**] [**match-first**] *policy-map-name* **Example:** switch(config)# policy-map policy1 switch(config-pmap-qos)#	Creates or accesses the specified policy map 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 class-default** **Example:** switch(config-pmap-qos)# class class-default switch(config-pmap-c-qos)#	Configures **class-default** to capture all traffic in this policy map.
Step 4	**set** {**cos** | **discard-class** | **dscp** | **precedence**} {**cos** | **discard-class** | **dscp** | **precedence**} **table** *table-map-name* **Example:** switch(config-pmap-c-qos)# set dscp dscp table dscp_mutation switch(config-pmap-c-qos)#	Sets the first packet field to the value of the second packet field based on the mapping values in the specified table map. For mutation mapping, both fields must have the same value. The specified table map must already exist. The example shows how to use mutation mapping on the DSCP field based on the mapping values in table map dscp_mutation.
Step 5	**service-policy** [**type qos**] *policy-map-name* [**no-stats**] **Example:** switch(config-pmap-c-qos)# service-policy testpolicy switch(config-pmap-c-qos)#	Defines the policy map to apply with the mutation map. The specified policy map must already exist and cannot contain a **service-policy** command. **Note** Classification within this service policy is based on the mutated value, not on the original value in the packet. **Note** The service-policy command can only be used for mutation mapping.
Step 6	**show policy-map** [**type** {**qos** | **queuing**}] [*policy-map-name*] **Example:** switch(config-pmap-c-qos)# show policy-map policy1	(Optional) Displays information about all configured policy maps or the specified policy map.
Step 7	**copy running-config startup-config** **Example:** switch(config-pmap-c-qos)# copy running-config startup-config	(Optional) Saves the running configuration to the startup configuration.

Verifying the Mutation Mapping Configuration

To display the mutation mapping configuration information, perform one of the following tasks:

Command	Purpose
show policy-map [**type** {**qos** | **queuing**}] [*policy-map-name*]	Displays information about all configured policy maps or the specified policy map.

For detailed information about the fields in the output from these commands, see the *Cisco Nexus 7000 Series NX-OS Quality of Service Command Reference, Release 4.0*.

Example Configuration

The following example shows a mutation configuration:

```
config t
  class-map type qos match-all dscp0-12
   match dscp0-12
  class-map type qos match-all dscp13-60
   match dscp13-60
  table-map mutate_dscp
   default 11
   from 0 to 0
   from 1 to 1
   from 2 to 1
   from 63 to 46
  policy-map type qos child_policy
   class dscp0-12
    police cir 10 mbps bc 200 ms pir 20 mbps be 200 ms conform transmit exceed set dscp
dscp table cir-markdown-map violate drop
  markdown-map violate drop
  rop
   class dscp13-63
    police cir 20 mbps bc 200 ms pir 40 mbps be 200 ms conform transmit exceed set dscp
dscp table cir-markdown-map violate drop
  markdown-map violate drop
  rop
   class class-default
    police cir 5 mbps bc 200 ms conform transmit violate drop
  policy-map type qos parent_policy_for_mutation
   class class-default
   set dscp dscp table mutate_dscp
   service-policy type qos child_qos_policy
```

CHAPTER 6

Configuring Policing

This chapter describes how to configure policing of traffic classes.

This chapter includes the following sections:

Information About Policing

Policing is the monitoring of the data rates for a particular class of traffic. When the data rate exceeds user-configured values, marking or dropping of packets occurs immediately. Policing does not buffer the traffic, so transmission delay is not affected. When traffic exceeds the data rate, you instruct the system to either drop the packets or mark QoS fields in them.

You can define single-rate, dual-rate, and color-aware policers.

Single-rate policers monitor the committed information rate (CIR) of traffic. Dual-rate policers monitor both CIR and peak information rate (PIR) of traffic. In addition, the system monitors associated burst sizes. Three “colors,” or conditions, are determined by the policer for each packet depending on the data rate parameters supplied: conform (green), exceed (yellow), or violate (red).

You can configure only one action for each condition. For example, you might police for traffic in a class to conform to the data rate of 256000 bits per second, with up to 200 millisecond bursts. The system would apply the conform action to traffic that falls within this rate, and it would apply the violate action to traffic that exceeds this rate.

Color-aware policers assume that traffic has been previously marked with a color. This information is then used in the actions taken by this type of policer.

For more information about policers, see RFC 2697 and RFC 2698.

Shared Policers

QoS applies the bandwidth limits specified in a shared policer cumulatively to all flows in the matched traffic. A shared policer applies the same policer to more than one interface simultaneously.

For example, if you configure a shared policer to allow 1 Mbps for all TFTP traffic flows on VLAN 1 and VLAN 3, the device limits the TFTP traffic for all flows combined on VLAN 1 and VLAN 3 to 1 Mbps.

The following are guidelines for configuring shared policers:

- You create named shared policers by entering the **qos shared-policer** command. If you create a shared policer and create a policy using that shared policer and attach the policy to multiple ingress ports, the device polices the matched traffic from all the ingress ports to which it is attached.
- You define shared policers in a policy map class within the police command. If you attach a named shared policer to multiple ingress ports, the device polices the matched traffic from all the ingress ports to which it is attached.
- Shared policing works independently on each module.

Licensing Requirements for Policing

The following table shows the licensing requirements for this feature:

Product	License Requirement
NX-OS	QoS requires no license. Any feature not included in a license package is bundled with the Cisco NX-OS system images and is provided at no extra charge to you. For a complete explanation of the NX-OS licensing scheme, see the *Cisco Nexus 7000 Series NX-OS Licensing Guide, Release 4.0*.

However, using VDCs requires an Advanced Services license.

Prerequisites for Policing

Policing has the following prerequisites:

- You must be familiar with Chapter 2, “Using Modular QoS CLI.”
- You are logged on to the switch.
- You are in the correct virtual device context (VDC). A VDC is a logical representation of a set of system resources. You can use the **switchto vdc** command with a VDC number.

Guidelines and Limitations

Use the following guidelines to configure policing:

- Each module polices independently, which might affect QoS features that are being applied to traffic that is distributed across more than one module. The following are examples of these QoS features:
  - Policers applied to a port channel interface.

  - Egress policers applied to a Layer 3 interface. Note that the device performs egress policing decisions at the ingress interface, on the ingress module.
  - Policers applied to a VLAN.
- All policers in either the ingress or egress direction must use the same mode. For example, if color-aware mode is needed for a class, all classes in that policy in the same direction must be in color-aware mode.

Configuring Policing

You can configure a single- or dual-rate policer.

This section includes the following topics:

Configuring 1-Rate and 2-Rate, 2-Color and 3-Color Policing

The type of policer created by the device is based on a combination of the **police** command arguments described in Table 6-1.

Note Specify the identical value for **pir** and **cir** to configure 1-rate 3-color policing.

Table 6-1 Arguments to the police Command

Argument	Description
cir	Committed information rate, or desired bandwidth, specified as a bit rate or a percentage of the link rate. Although a value for **cir** is required, the argument itself is optional. The range of values is 1 to 80000000000; the range of policing values that are mathematically significant is 8000 to 80 Gbps.
percent	Specifies the rate as a percentage of the interface rate. The range of values is 1 to 100%.
bc	Indication of how much the **cir** can be exceeded, either as a bit rate or an amount of time at **cir**. The default is 200 milliseconds of traffic at the configured rate. The default data rate units are bytes, and the Gigabit per second (gbps) rate is not supported for this parameter.
pir	Peak information rate, specified as a PIR bit rate or a percentage of the link rate. There is no default. The range of values is 1 to 80000000000; the range of policing values that are mathematically significant is 8000 to 80 Gbps. The range of percentage values is 1 to 100%.

Table 6-1 Arguments to the police Command (continued)

Argument	Description
be	Indication of how much the **pir** can be exceeded, either as a bit rate or an amount of time at **pir**. When the **bc** value is not specified, the default is 200 milliseconds of traffic at the configured rate. The default data rate units are bytes, and the Gigabit per second (gbps) rate is not supported for this parameter. **Note** You must specify a value for **pir** before the device displays this argument.
conform	Single action to take if the traffic data rate is within bounds. The basic actions are transmit or one of the **set** commands listed in Table 6-4. The default is transmit.
exceed	Single action to take if the traffic data rate is exceeded. The basic actions are drop or markdown. The default is drop.
violate	Single action to take if the traffic data rate violates the configured rate values. The basic actions are drop or markdown. The default is drop.

Note For information on the color-aware **police** command arguments, see the "Configuring Color-Aware Policing" section on page 6-8.

Although all the arguments in Table 6-1 are optional, you must specify a value for **cir**. In this section, **cir** indicates what is its value but not necessarily the keyword itself. The combination of these arguments and the resulting policer types and actions are shown in Table 6-2.

Table 6-2 Policer Types and Actions from Police Arguments Present

Police Arguments Present	Policer Type	Policer Action
cir, but not **pir**, **be**, or **violate**	1-rate, 2-color	<= **cir**, then **conform**; else **violate**
cir and **pir**	1-rate, 3-color	<= **cir**, then **conform**; <= **pir**, then **exceed**; else **violate** **Note** You must specify identical values for **cir** and **pir**.
cir and **pir**	2-rate, 3-color	<= **cir**, then **conform**; <= **pir**, then **exceed**; else **violate**

The policer actions that you can specify are described in Table 6-3 and Table 6-4.

Note The policer can only drop or markdown packets that exceed or violate the specified parameters. See Chapter 4, "Configuring Marking" for information on marking down packets.

The data rates used in the **police** command are described in Table 6-5.

Table 6-5 The Data Rates for the police Command

Rate	Description
bps	Bits per second (default)
kbps	1,000 bits per seconds
mbps	1,000,000 bits per second
gbps	1,000,000,000 bits per second

Burst sizes used in the **police** command are described in Table 6-6.

SUMMARY STEPS

Note Specify the identical value for **pir** and **cir** to configure 1-rate 3-color policing.

1. **config t**
2. **policy-map** [**type qos**] [**match-first**] *policy-map-name*
3. **class** [**type qos**] {*class-map-name* | **class-default**} [**insert-before** *before-class-map-name*]
4. **police** [**cir**] {*committed-rate* [*data-rate*] | **percent** *cir-link-percent*} [**bc** *committed-burst-rate* [*link-speed*]] [**pir**] {*peak-rate* [*data-rate*] | **percent** *cir-link-percent*} [**be** *peak-burst-rate* [*link-speed*]] {**conform** {**transmit** | **set-prec-transmit** | **set-dscp-transmit** | **set-cos-transmit** | **set-qos-transmit** | **set-discard-class-transmit**} [**exceed** {**drop** | **set dscp dscp table** {*cir-markdown-map*}} [**violate** {**drop** | **set dscp dscp table** {*pir-markdown-map*}}]]}
5. **exit**
6. **exit**
7. **show policy-map** [**type qos**] [*policy-map-name*]
8. **copy running-config startup-config**

DETAILED STEPS

	Command	Purpose
Step 1	**config t** **Example:** switch# config t switch(config)#	Enters configuration mode.
Step 2	**policy-map** [**type qos**] [**match-first**] *policy-map-name* **Example:** switch(config)# policy-map policy1 switch(config-pmap-qos)#	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**} [**insert-before** *before-class-map-name*] **Example:** switch(config-pmap-qos)# class class-default switch(config-pmap-c-qos)#	Creates a reference to *class-map-name*, and enters policy-map class configuration mode. The class is added to the end of the policy map unless **insert-before** is used to specify the class to insert before. Specify **class-default** to select all traffic that is not matched by classes in the policy map so far.
Step 4	**police** [**cir**] {*committed-rate* [*data-rate*] | **percent** *cir-link-percent*} [[**bc** *committed-burst-rate* [*link-speed*]][**pir**] {*peak-rate* [*data-rate*] | **percent** *cir-link-percent*} [[**be** *peak-burst-rate* [*link-speed*]] [**conform** {**transmit** | **set-prec-transmit** | **set-dscp-transmit** | **set-cos-transmit** | **set-qos-transmit** | **set-discard-class-transmit**} [**exceed** {**drop** | **set dscp dscp table** {*cir-markdown-map*}} [**violate** {**drop** | **set dscp dscp table** {*pir-markdown-map*}}]]] **Example #1:** switch(config-pmap-c-qos)# police cir 256000 conform transmit violate set dscp dscp table pir-markdown-map switch(config-pmap-c-qos)# **Example #2:** switch(config-pmap-c-qos)# police cir 256000 pir 256000 conform transmit exceed set dscp dscp table cir-markdown-map violate drop switch(config-pmap-c-qos)#	Polices **cir** in bits or as a percentage of the link rate. The **conform** action is taken if the data rate is <= **cir**. If **be** and **pir** are not specified, all other traffic takes the **violate** action. If **be** or **violate** are specified, then the **exceed** action is taken if the data rate <= **pir**, and the **violate** action is taken otherwise. The actions are described in Table 6-3 and Table 6-4. The data rates and link speeds are described in Table 6-5 and Table 6-6. This first example shows a 1-rate, 2-color policer that transmits if the data rate is within 200 milliseconds of traffic at 256000 bps and marks IP precedence to 6 if the data rate is exceeded. This second example shows a 1-rate, 3-color policer that transmits if the data rate is within 200 milliseconds of traffic at 256000 bps, marks DSCP to 6 if the data rate is within 300 milliseconds of traffic at 256000 bps, and drops packets otherwise. **Note** You must specify identical values for **cir** and **pir.**

	Command	Purpose

Step 5	**exit**
	
	Example:
	`switch(config-pmap-c-qos)# exit`
	`switch(config-pmap-qos)#`
		Exits policy-map class configuration mode and enters policy-map mode.

Step 6	**exit**
	
	Example:
	`switch(config-pmap-qos)# exit`
	`switch(config)#`
		Exits policy-map mode and enters configuration mode.

Step 7	**show policy-map** [**type qos**] [*policy-map-name*] **Example:** `switch(config)# show policy-map`	(Optional) Displays information about all configured policy maps or a selected policy map of type qos.
Step 8	**copy running-config startup-config** **Example:** `switch(config)# copy running-config startup-config`	(Optional) Saves the running configuration to the startup configuration.

Use the **show policy-map** command to display the policy1 policy-map configuration:

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

Configuring Color-Aware Policing

Color-aware policing implies that the QoS DSCP field in a class of traffic has been previously marked with values that you can use in a policer. This feature allows you to mark traffic at one node in a network and then take action based on this marking at a subsequent node.

Note For information on the **police** command, see the "Configuring 1-Rate and 2-Rate, 2-Color and 3-Color Policing" section on page 6-3.

You can use one or more of the four police command class maps **conform-color** or **exceed-color** to perform color-aware policing. These keywords require a class-map name that is used to classify packets. Based on the match criteria that you specify in the class maps, the traffic is classified into one of these two classes or class-default if there is no match. The policer then takes the following action:

- Packets that belong to the **conform-color** class are policed with the **cir** and **pir** arguments to the **police** command.
- Packets that belong to the **exceed-color** class are policed only against the **pir** argument to the **police** command. If **pir** is not specified, then the **cir** values are used.
- Packets that end up in class-default because they fail to match either the **conform-color** or **exceed-color** class will immediately take the violate action.

Note A color other than class-default cannot be assigned to the violate action because according to RFC 2697 and RFC 2698, all packets must be assigned a color.

You can set the DSCP value for color-aware policing to a specified value. The list of valid DSCP values is shown in Table 6-7.

Table 6-7 Color-Aware Policing Valid 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	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

After you apply color-aware policing, all matching packets in the device will be policed according to the specifications of the color-aware policer.

To configure color-aware policing, follow these steps:

Step 1 Create the class map. For information about configuring class maps, see Chapter 3, “Configuring Classification.”

Step 2 Create a policy map. For information about policy maps, see this chapter and Chapter 2, “Using Modular QoS CLI,”

Step 3 Configure the color-aware class map as described in this section.

Step 4 Apply the service policy to the interfaces. For information about attaching policies to interfaces, see Chapter 2, “Using Modular QoS CLI.”

Note The rates specified in the shared policer are shared by the number of interfaces to which you apply the service policy. Each interface does not have its own dedicated rate as specified in the shared policer.

SUMMARY STEPS

1. **config t**
2. **class-map** {**conform-color-in** | **conform-color-out** | **exceed-color-in** | **exceed-color-out**}
3. **match dscp** *dscp-value*
4. **policy-map** [**type qos**] [**match-first**] *policy-map-name*
5. **class** [**type qos**] {*class-map-name* | **class-default**} [**insert-before** *before-class-map-name*]
6. **police** [**cir**] {*committed-rate* [*data-rate*] | **percent** *cir-link-percent*} [**bc** *committed-burst-rate* [*link-speed*]] [**pir**] {*peak-rate* [*data-rate*] | **percent** *cir-link-percent*} [**be** *peak-burst-rate* [*link-speed*]] {**conform** {**transmit** | **set-prec-transmit** | **set-dscp-transmit** | **set-cos-transmit** | **set-qos-transmit** | **set-discard-class-transmit**} [**exceed** {**drop** | **set dscp dscp table** {*cir-markdown-map*}} [**violate** {**drop** | **set dscp dscp table** {*pir-markdown-map*}}]]}
7. **exit**
8. **show policy-map** *policy-map-name*
9. **copy running-config startup-config**

DETAILED STEPS

	Command	Purpose

Step 1	**config t**
	
	Example:
	`switch# config t`
	`switch(config)#`
		Enters configuration mode.

Step 2	**class-map** {**conform-color-in** | **conform-color-out** | **exceed-color-in** | **exceed-color-out**}
	
	Example:
	`switch(config)# class-map conform-color-in`
	`switch(config-color-map)#`
		Accesses the color-aware class map, and enters color-map mode. When you enter this command, the system returns the following message:
		`Warning: Configuring match for any DSCP values in this class-map will make ALL policers in the system color-aware for those DSCP values.`

Step 3	**match dscp** *dscp-value*
	
	Example:
	`switch(config-color-map)# match dscp af22`
	`switch(config-color-map)#`
		Specifies the DSCP value to match for color-aware policers. See Table 6-7 for list of valid values.

Step 4	**policy-map** [**type qos**] [**match-first**] *policy-map-name*
	
	Example:
	`switch(config)# policy-map policy1`
	`switch(config-pmap-qos)#`
		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.

	Command	Purpose
Step 5	**class** [**type qos**] {*class-map-name* | **class-default**} [**insert-before** *before-class-map-name*] **Example:** switch(config-pmap-qos)# class class-default switch(config-pmap-c-qos)#	Creates a reference to *class-map-name* and enters policy-map class configuration mode. The class is added to the end of the policy map unless **insert-before** is used to specify the class to insert before. Specify **class-default** to select all traffic that is not matched by classes in the policy map so far.
Step 6	**police** [**cir**] {*committed-rate* [*data-rate*] | **percent** *cir-link-percent*} [[**bc** *committed-burst-rate* [*link-speed*]][**pir**] {*peak-rate* [*data-rate*] | **percent** *cir-link-percent*} [[**be** *peak-burst-rate* [*link-speed*]] [**conform** {**transmit** | **set-prec-transmit** | **set-dscp-transmit** | **set-cos-transmit** | **set-qos-transmit** | **set-discard-class-transmit**} [**exceed** {**drop** | **set dscp dscp table** {*cir-markdown-map*}} [**violate** {**drop** | **set dscp dscp table** {*pir-markdown-map*}}]]] **Example #1:** switch(config-pmap-c-qos)# police cir 256000 be 300 ms conform-class my_conform_class_map exceed-class my_exceed_class_map conform transmit exceed set dscp dscp table cir-markdown-map violate drop switch(config-pmap-c-qos)# **Example #2:** switch(config-pmap-c-qos)# police cir 256000 pir 512000 conform-class my_conform_class_map exceed-class my_exceed_class_map conform transmit exceed set dscp dscp table cir-markdown-map violate drop switch(config-pmap-c-qos)#	Polices **cir** in bits or as a percentage of the link rate. The **conform** action is taken if the data rate is <= **cir**. If **be** and **pir** are not specified, all other traffic takes the **violate** action. If **be** or **violate** are specified, then the **exceed** action is taken if the data rate <= **pir**, and the **violate** action is taken otherwise. The actions are described in Table 6-3 and Table 6-4. The data rates and link speeds are described in Table 6-5 and Table 6-6. This first example shows a 1-rate, 3-color color-aware policer that transmits if conform-class the data rate is within 200 milliseconds of traffic at 256000 bps, marks DSCP to 6 if the exceed-class the data rate is within 300 milliseconds of traffic at 256000 bps, and drops packets otherwise. This second example shows a 2-rate, 3-color color-aware policer that transmits if the data rate is within 200 milliseconds of traffic at 256000 bps, marks CoS to 5 if the data rate exceeds 200 milliseconds of traffic at 512 bps, and drops packets otherwise
Step 7	**exit** **Example:** switch(config-color-map)# exit switch(config)#	Exits color-map mode and then enters configuration mode.
Step 8	**show policy-map** [**type qos**] [*policy-map-name*] **Example:** switch(config)# show policy-map	(Optional) Displays information about all configured policy maps or a selected policy map of type qos.
Step 9	**copy running-config startup-config** **Example:** switch(config)# copy running-config startup-config	(Optional) Saves the running configuration to the startup configuration.

Use the **show policy-map** command to display the policy1 policy-map configuration:

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

Configuring Ingress and Egress Policing

You can apply the policing instructions in a QoS policy map to ingress or egress packets by attaching that QoS policy map to an interface. To select ingress or egress, you specify either the **input** or **output** keyword in the **service-policy** command. For more information, see the "Attaching and Detaching a QoS Policy Action from an Interface" section on page 2-18.

Configuring Markdown Policing

Markdown policing is the setting of a QoS field in a packet when traffic exceeds or violates the policed data rates. You can configure markdown policing by using the **set** commands for policing action described in Table 6-3 and Table 6-4.

The example in this section shows you how to use a table map to perform markdown.

SUMMARY STEPS

1. **config t**
2. **policy-map** [**type qos**] [**match-first**] *policy-map-name*
3. **class** [**type qos**] {*class-map-name* | **class-default**} [**insert-before** *before-class-map-name*]
4. **police** [**cir**] {*committed-rate* [*data-rate*] | **percent** *cir-link-percent*} [**bc** *committed-burst-rate* [*link-speed*]] [**pir**] {*peak-rate* [*data-rate*] | **percent** *cir-link-percent*} [**be** *peak-burst-rate* [*link-speed*]] {**conform** *conform-action* [**exceed** {**drop** | **set dscp dscp table** *cir-markdown-map*} [**violate** {**drop** | **set dscp dscp table** *pir-markdown-map*}]]}}
5. **exit**
6. **exit**
7. **show policy-map** [**type qos**] [*policy-map-name*]
8. **copy running-config startup-config**

DETAILED STEPS

	Command	Purpose
Step 1	**config t** **Example:** switch# config t switch(config)#	Enters configuration mode.
Step 2	**policy-map** [**type qos**] [**match-first**] *policy-map-name* **Example:** switch(config)# policy-map policy1 switch(config-pmap-qos)#	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**} [**insert-before** *before-class-map-name*] **Example:** switch(config-pmap-qos)# class class-default switch(config-pmap-c-qos)#	Creates a reference to *class-map-name*, and enters policy-map class configuration mode. The class is added to the end of the policy map unless **insert-before** is used to specify the class to insert before. Specify **class-default** to select all traffic not matched by classes in the policy map so far.
Step 4	**police** [**cir**] {*committed-rate* [*data-rate*] | **percent** cir-*link-percent*} [[**bc** | **burst**] *burst-rate* [*link-speed*]] [[**be** | **peak-burst**] *peak-burst-rate* [*link-speed*]] [**conform** *conform-action* [**exceed set dscp dscp table** *cir-markdown-map* [**violate set dscp dscp table** *pir-markdown-map*]]] **Example:** switch(config-pmap-c-qos)# police cir 256000 be 300 ms conform transmit exceed set dscp dscp table cir-markdown-map violate drop switch(config-pmap-c-qos)#	Polices **cir** in bits or as a percentage of the link rate. The **conform** action is taken if the data rate is <= **cir**. If **be** and **pir** are not specified, all other traffic takes the **violate** action. If **be** or **violate** are specified, then the **exceed** action is taken if the data rate <= **pir**, and the **violate** action is taken otherwise. The actions are described in Table 6-3 and Table 6-4. The data rates and link speeds are described in Table 6-5 and Table 6-6. This example shows a 1-rate, 3-color policer that transmits if the data rate is within 200 milliseconds of traffic at 256000 bps; marks down DSCP using the system-defined table map if the data rate is within 300 milliseconds of traffic at 256000 bps; and drops packets otherwise.

	Command	Purpose

Step 5	**exit**
	
	Example:
	`switch(config-pmap-c-qos)# exit`
	`switch(config-pmap-qos)#`
		Exits policy-map class configuration mode and enters policy-map mode.

Step 6	**exit**
	
	Example:
	`switch(config-pmap-qos)# exit`
	`switch(config)#`
		Exits policy-map mode and enters configuration mode.

Step 7	**show policy-map** [**type qos**] [*policy-map-name*] **Example:** `switch(config)# show policy-map`	(Optional) Displays information about all configured policy maps or a selected policy map of type qos.
Step 8	**copy running-config startup-config** **Example:** `switch(config)# copy running-config startup-config`	(Optional) Saves the running configuration to the startup configuration.

Use the **show policy-map** command to display the policy1 policy-map configuration:

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

Configuring Shared Policers

The shared-policer feature allows you to apply the same policing parameters to several interfaces simultaneously. You create a shared policer by assigning a name to a policer, and then applying that policer to a policy map that you attach to the specified interfaces. The shared policer is also referred to as the named aggregate policer in other Cisco documentation.

Note After you configure the shared policer, you can use the shared-policer name to configure any type of shared policing, as described in the following sections: "Configuring 1-Rate and 2-Rate, 2-Color and 3-Color Policing" section on page 6-3, "Configuring Color-Aware Policing" section on page 6-8, "Configuring Ingress and Egress Policing" section on page 6-12, and - "Configuring Markdown Policing" section on page 6-12.

To configure shared policing, follow these steps:

Step 1 Configure the shared policer as described in this section.

Step 2 Create the class map. For information about configuring class maps, see Chapter 3, "Configuring Classification."

Step 3 Create a policy map. For information about policy maps, see this chapter and Chapter 2, "Using Modular QoS CLI."

Step 4 Reference the shared policer to the policy map as described in this section.

Step 5 Apply the service policy to the interfaces. For information about attaching policies to interfaces, see Chapter 2, "Using Modular QoS CLI."

Note The rates specified in the shared policer are shared by the number of interfaces to which you apply the service policy. Each interface does not have its own dedicated rate as specified in the shared policer.

SUMMARY STEPS

1. **config t**
2. **qos shared-policer** [**type qos**] *shared-policer-name* [**cir**] {*committed-rate* [*data-rate*] | **percent** *cir-link-percent*} [**bc** *committed-burst-rate* [*link-speed*]] [**pir**] {*peak-rate* [*data-rate*] | **percent** *cir-link-percent*} [**be** *peak-burst-rate* [*link-speed*]] {{**conform** *conform-action* [**exceed** {**drop** | **set dscp dscp table** *cir-markdown-map*} [**violate** {**drop** | **set dscp dscp table** *pir-markdown-map*}]]}
3. **policy-map** [**type qos**] [**match-first**] *policy-map-name*
4. **class** [**type qos**] {*class-map-name* | **class-default**} [**insert-before** *before-class-map-name*]
5. **police aggregate** *shared-policer-name*
6. **exit**
7. **exit**
8. **show qos shared-policer** *shared-policer-name*]
9. **copy running-config startup-config**

DETAILED STEPS

	Command	Purpose
Step 1	**config t** **Example:** switch# config t switch(config)#	Enters configuration mode.
Step 2	**qos shared-policer** [**type qos**] *shared-policer-name* [**cir**] {*committed-rate* [*data-rate*] | **percent** *cir-link-percent*} [[**bc** *committed-burst-rate* [*link-speed*]][**pir**] {*peak-rate* [*data-rate*] | **percent** *cir-link-percent*} [[**be** *peak-burst-rate* [*link-speed*]] [**conform** *conform-action* [**exceed set dscp dscp table** *cir-markdown-map* [**violate set dscp dscp table** *pir-markdown-map*]]] **Example:** switch(config)# qos shared-policer test1 cir 10 mbps switch(config)#	Creates or accesses the shared policer. The shared-policer-name can contain alphabetic, hyphen, or underscore characters, is case sensitive, and can be up to 40 characters. Polices **cir** in bits or as a percentage of the link rate. The **conform** action is taken if the data rate is <= **cir**. If **be** and **pir** are not specified, all other traffic takes the **violate** action. If **be** or **violate** are specified, then the **exceed** action is taken if the data rate <= **pir**, and the **violate** action is taken otherwise. The actions are described in Table 6-3 and Table 6-4. The data rates and link speeds are described in Table 6-5 and Table 6-6.
Step 3	**policy-map** [**type qos**] [**match-first**] *policy-map-name* **Example:** switch(config)# policy-map policy1 switch(config-pmap-qos)#	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 4	**class** [**type qos**] {*class-map-name* | **class-default**} [**insert-before** *before-class-map-name*] **Example:** switch(config-pmap-qos)# class class1 switch(config-pmap-c-qos)#	Creates a reference to *class-map-name* and enters policy-map class configuration mode. The class is added to the end of the policy map unless **insert-before** is used to specify the class to insert before. Specify **class-default** to select all traffic that is currently not matched by classes in the policy map.
Step 5	**police aggregate** *shared-policer-name* **Example:** switch(config-pmap-c-qos)# police aggregate test1 switch(config-pmap-c-qos)#	Creates a reference in the policy map to *shared-policer-name*.

	Command	Purpose

Step 6	**exit**
	
	Example:
	`switch(config-pmap-c-qos)# exit`
	`switch(config-pmap-qos)#`
		Exits policy-map class configuration mode and enters policy-map mode.

Step 7	**exit**
	
	Example:
	`switch(config-pmap-qos)# exit`
	`switch(config)#`
		Exits policy-map mode and enters configuration mode.

Step 8	**show qos shared-policer** [**type**] [*shared-policer-name*] **Example:** `switch(config)# show qos shared-policer test1`	(Optional) Displays information about the configuration of all shared policers.
Step 9	**copy running-config startup-config** **Example:** `switch(config)# copy running-config startup-config`	(Optional) Saves the running configuration to the startup configuration.

Use the **show qos shared-policer** command to display the test1 shared-policer configurations:

```
switch# show qos shared-policer test1
```

Verifying the Policing Configuration

Use these command to verify the policing configuration.

show policy-map	Displays information about policy maps and policing.
show qos shared-policer [**type qos**] [*policer-name*]	Displays information about all shared policing.

Example Configurations

The following are examples of how to configure policing:

- 1-rate, 2-color policer:

```
config t
  policy-map policy1
    class one_rate_2_color_policer
      police cir 256000 conform transmit violate drop
```

- 1-rate, 2-color policer with DSCP markdown:

```
config t
  policy-map policy2
    class one_rate_2_color_policer_with_dscp_markdown
```

```
      police cir 256000 conform transmit violate drop
```

- 1-rate, 3-color policer:

```
config t
  policy-map policy3
    class one_rate_3_color_policer
      police cir 256000 pir 256000 conform transmit exceed set dscp dscp table
cir-markdown-map violate drop
```

- 2-rate, 3-color policer:

```
config t
  policy-map policy4
    class two_rate_3_color_policer
      police cir 256000 pir 256000 conform transmit exceed set dscp dscp table
cir-markdown-map violate drop
```

- Color-aware policer for specified DSCP values:

```
config t
    class-map conform-color-in
      match dscp 0-10
  policy-map policy5
    class one_rate_2_color_policer
      police cir 256000 conform transmit violate drop
```

- Shared policer:

```
config t
  qos shared-policer type qos udp_policer type cir 10 mbps pir 20 mbps conform transmit
exceed set dscp dscp table cir-markdown-map violate drop
  policy-may type qos udp_policy
   class type qos udp_qos
      police aggregate udp_1mbps
```

CHAPTER 7

Configuring Queuing and Scheduling

This chapter describes how to configure the QoS queuing and scheduling features on the device.

This chapter includes the following sections:

Information About Queuing and Scheduling

Traffic queuing is the ordering of packets and applies to both input and output of data. Device modules can support multiple queues, which you can use to control the sequencing of packets in different traffic classes. You can also set weighted random early detection (WRED) and taildrop thresholds. The device drops packets only when the configured thresholds are exceeded.

Traffic scheduling is the methodical output of packets at a desired frequency to accomplish a consistent flow of traffic. You can apply traffic scheduling to different traffic classes to weight the traffic by priority.

The queuing and scheduling processes allow you to control the bandwidth that is allocated to the traffic classes, so that you achieve the desired trade-off between throughput and latency for your network.

Table 7-1 describes the system-defined queues that you can use to perform queuing and scheduling.

Table 7-1 System-Defined Queue Types

Queue Type	Direction	Description
2q4t	Input	2 queues with 4 WRED or tail drop thresholds per queue
1p3q4t	Output	1 strict priority plus 3 normal queues or 4 normal queues with 4 WRED or tail-drop thresholds per queue

Table 7-1 System-Defined Queue Types (continued)

Queue Type	Direction	Description
8q2t	Input	8 queues with 2 tail drop thresholds per queue
1p7q4t	Output	1 strict priority queue plus 7 normal queues or 8 normal queues with 4 WRED or tail drop thresholds per queue

The queues match on the Class of Service (CoS) field. The device ensures that every CoS value from 0 to 7 maps to a queue for each queue type. Only one queue for a queue type can be assigned a specific CoS value. For more information about the system-defined queues, see Table 2-3.

This section includes the following topics:

Setting Ingress Port CoS

You can set the CoS field in all ingress packets for untrusted ports. By default, ports are trusted and the CoS field is not modified. (Use this method to configure the port state to trusted or untrusted.)

For information about configuring ingress port CoS, see the "Configuring Ingress Port CoS" section on page 7-5.

Modifying Class Maps

You can modify the CoS values that are matched by system-defined queuing class maps, which modifies the CoS-to-queue mapping. Table 2-3 on page 2-7 lists the default system-defined CoS values. Each CoS value appears only once in the queues of the same type.

Note When you modify a system-defined class queuing map, the changes occur immediately and may disrupt traffic on all VDCs.

For information about configuring class maps, see the "Modifying Queuing Class Maps" section on page 7-6.

Congestion Avoidance

You can use the following methods to proactively avoid traffic congestion on the device:

- Apply WRED to a class of traffic, which allows the device to drop packets based on the CoS field. WRED is designed to work with TCP traffic.
- Apply tail drop to a class of traffic, which allows the device to drop packets based on the Class of Service (CoS) field.

For information about configuring congestion avoidance, see the "Configuring Congestion Avoidance" section on page 7-8.

Congestion Management

For ingress packets, you can configure congestion management by specifying a bandwidth that allocates a minimum data rate to a queue.

For egress packets, you can choose one of the following congestion management methods:

- Specify a bandwidth that allocates a minimum data rate to a queue.
- Impose a maximum data rate on a class of traffic so that excess packets are retained in a queue to shape the output rate.
- Allocate all data for a class of traffic to a priority queue. The device distributes the remaining bandwidth among the other queues.

For information about configuring congestion management, see the "Configuring Congestion Management" section on page 7-13.

Virtualization Support

A virtual device context (VDC) is a logical representation of a set of system resources. Other than configuring class maps, queuing and scheduling apply only to the VDC where the commands are entered. For information about configuring class maps, see the "Modifying Queuing Class Maps" section on page 7-6.

For information about configuring VDCs, see the *Cisco NX-OS Virtual Device Context Configuration Guide*.

Licensing Requirements for Queuing and Scheduling

The following table shows the licensing requirements for this feature:

Product	License Requirement
NX-OS	QoS requires no license. Any feature not included in a license package is bundled with the Cisco NX-OS system images and is provided at no extra charge to you. For a complete explanation of the NX-OS licensing scheme, see the *Cisco Nexus 7000 Series NX-OS Licensing Guide, Release 4.0*.

However, using VDCs requires an Advanced Services license.

Prerequisites for Queuing and Scheduling

Queuing and scheduling have the following prerequisites:

- You must be familiar with Chapter 2, "Using Modular QoS CLI."
- You are logged on to the switch.

- You are in the correct virtual device context (VDC). A VDC is a logical representation of a set of system resources. You can use the **switchto vdc** command with a VDC number.

Guidelines and Limitations

Use the following guidelines to configure queuing and scheduling:

- Configure system-defined class maps with care as the changes occur immediately and traffic may be disrupted on all VDCs.
- When you are working with 10-Gigabit Ethernet ports in shared mode, the egress queuing policy applies to all the ports in the port group. With 10-Gigabit Ethernet ports in shard mode, all the ports in the port group must be in the same VDC. See the *Cisco NX-OS Interfaces Configuration Guide* for information on shared and dedicated mode, as well as the *Cisco Nexus 7000 Series Hardware Installation and Reference Guide* for information about the port groups.
- You cannot set either the queue limit or WRED on ingress 10-Gigabit Ethernet ports.

Configuring Queuing and Scheduling

You configure queuing and scheduling by creating policy maps of type queuing that you apply to either traffic direction of an interface. You can modify system-defined class maps, which are used in policy maps to define classes of traffic to which you want to apply policies. For information about configuring policy maps and class maps, see Chapter 2, “Using Modular QoS CLI.”

You can configure the congestion-avoidance features (which include tail drop and WRED) in any queue. You can configure one of the egress congestion management features (which include priority, shaping, and bandwidth) in output queues, and bandwidth in input queues.

We recommend that you modify the CoS value *before* you create a policy map. You can modify the CoS values that are matched by device-defined class map queues. You must assign each CoS value from 0 to 7 to one or more of the queues for each queue type. Each CoS value can be used only once in each queue type.

The system-defined policy maps default-in-policy and default-out-policy are attached to all ports to which you do not apply a queuing policy map. The default policy maps cannot be configured. For more information about the default policy maps, see Table 2-5.

In Cisco NX-OS 4.0(3) and later releases, the WRR for the default-in-policy changed from 50/50 to 80/20.

If you downgrade from Release 4.0(3) to Release 4.0(2) and enter the **show running-configuration** command, the input default queuing policy has an unknown enum in the display, as follows:

```
switch# show running-config
version 4.0(2)

…

    …
    policy-map type queuing default-in-policy
    class type queuing unknown enum 0
    queue-limit percent 50
    bandwidth percent 80
    class type queuing unknown enum 0
    queue-limit percent 50
    bandwidth percent 20
```

If you copy and paste this configuration into any NX-OS software release, the device sends errors while executing all the commands starting from the **policy-map type queuing default-in-policy** command. These errors are harmless and will not affect the running of the device.

This section includes the following topics:

Configuring Ingress Port CoS

To make a port untrusted, set the CoS value to a static value.

Note By default, ports are trusted (trust CoS) and the CoS field is not modified. When you configure the ingress port CoS value, the port becomes untrusted.

You use the ingress default queues from the system-defined queue classes for the type of module to which you want to apply the policy map. See Table 2-3 on page 2-7 for the list of system-defined class maps for each type of module.

The CoS values set using this procedure apply to all packets ingressing the specified interfaces, not just to the class-default packets. If you set the CoS value, the device modifies the value before ingress queuing and scheduling so the CoS-modified packets are classified differently.

Note If you want to change the system-defined queuing class-maps, you must either modify the configured queuing policies or create new queuing policies and attach these to the affected interfaces. If you do not do this, you can render the default queuing or the configured queuing policies invalid, which may affect interfaces in multiple VDCs.

SUMMARY STEPS

1. **config t**
2. **policy-map type queuing** [**match-first**] *policy-map-name*
3. **class type queuing** *class-queuing-name*
4. **set cos** *value*
5. **exit**
6. **show policy-map type queuing** *policy-map-name*
7. **copy running-config startup-config**

DETAILED STEPS

	Command	Purpose
Step 1	**config t** **Example:** switch# config t switch(config)#	Enters configuration mode.
Step 2	**policy-map type queuing** [**match-first**] *policy-map-name* **Example:** switch(config)# policy-map type queuing untrusted_port_cos switch(config-pmap-que)#	Configures the policy map of type queuing, and then enters policy-map mode for the policy-map name that you specify. Policy-map names can contain alphabetic, hyphen, or underscore characters, are case sensitive, and can be up to 40 characters.
Step 3	**class type queuing** *class-queuing-name* **Example:** switch(config)# class type queuing 2q4t-in-q-default switch(config-pmap-c-que)#	Configures the class map of type queuing, and then enters policy-map class queuing mode. Class queuing names are listed in Table 2-3. **Note** To configure port CoS, you can use only an ingress default system-defined queue type.
Step 4	**set cos** *value* **Example:** switch(config-pmap-c-que)# set cos 5	Sets the CoS field in all ingress packet to the value specified. The range is from 0 to 7.
Step 5	**exit** **Example:** switch(config-cmap-que)# exit switch(config)#	Exits policy-map queue mode, and enters configuration mode.
Step 6	**show policy-map type queuing** *policy-map-name* **Example:** switch(config)# show policy-map type untrusted_port_cos	(Optional) Displays information about all configured policy maps or a selected policy map of type queuing.
Step 7	**copy running-config startup-config** **Example:** switch(config)# copy running-config startup-config	(Optional) Saves the running configuration to the startup configuration.

Modifying Queuing Class Maps

You can modify the CoS values that are matched by system-defined class maps. Table 2-3 on page 2-7 lists the default system-defined CoS values.

The system-defined class maps can only be changed from the default VDC. Changes occur immediately and are applied to all ports on all VDCs that use the modified class map.

Note When you modify a system-defined class map, the changes occur immediately and may disrupt traffic on all VDCs that use the modified class map.

The device automatically modifies the CoS values that you configured in other queues so that each CoS value appears only once in the queues of the same type.

Note If you want to change the system-defined queuing class-maps, you must either modify the configured queuing policies or create new queuing policies and attach these to the affected interfaces. If you do not do this, you can render the default queuing or the configured queuing policies invalid, which may affect interfaces in multiple VDCs.

BEFORE YOU BEGIN

Ensure you are in the default VDC for the device.

SUMMARY STEPS

1. **config t**
2. **class-map type queuing match-any** *class-queuing-name*
3. **match cos** *value-range*
4. Repeat Steps 2 and 3 to modify CoS values for additional queues
5. **exit**
6. **show class-map type queuing** [*class-queuing-name*]
7. **copy running-config startup-config**

DETAILED STEPS

	Command	Purpose
Step 1	**config t** **Example:** switch# config t switch(config)#	Enters configuration mode.
Step 2	**class-map type queuing match-any** *class-queuing-name* **Example:** switch(config)# class-map type queuing match-any 1p3q4t-out-pq1 switch(config-cmap-que)#	Configures the class map of type queuing, and then enters class-map queuing mode. Class queuing names are listed in Table 2-3.
Step 3	**match cos** *value-range* **Example:** switch(config-cmap-que)# match 0-3,7	Sets the CoS value range matched by this queue. You can specify a range of values by using a hyphen between the beginning and ending values and a comma between values. The range is from 0 to 7.

	Command	Purpose
Step 4	Repeat Steps 2 and 3 to modify CoS values for additional queues.	—

Step 5	**exit**
	
	Example:
	`switch(config-cmap-que)# exit`
	`switch(config)#`
		Exits class-map queue mode and enters configuration mode.

Step 6	**show class-map type queuing** [*class-queuing-name*] **Example:** `switch(config)# show class-map type queuing`	(Optional) Displays information about all configured class maps or a selected class map of type queuing. Class queuing names are listed in Table 2-3.
Step 7	**copy running-config startup-config** **Example:** `switch(config)# copy running-config startup-config`	(Optional) Saves the running configuration to the startup configuration.

Configuring Congestion Avoidance

You can configure congestion avoidance with tail drop or WRED features. Both features can be used in ingress and egress policy maps.

Note WRED and tail drop cannot be configured in the same class.

This section includes the following topics:

- Configuring Tail Drop, page 7-8
- Configuring WRED, page 7-10

Configuring Tail Drop

You can configure tail drop on both ingress and egress queues by setting thresholds by CoS values. The device will drop packets that exceed the thresholds. You can specify a threshold based on queue size or buffer memory used by the queue.

Note You cannot configure the queue size on ingress 10-Gigabit Ethernet ports.

You use the system-defined queue classes for the type of module to which you want to apply the policy map. See Table 2-3 on page 2-7.

Note WRED and tail drop cannot be configured in the same class.

SUMMARY STEPS

1. **config t**

2. **policy-map type queuing** [**match-first**] *policy-map-name*
3. **class type queuing** *class-queuing-name*
4. **queue-limit cos** *value* {*threshold* [**packets** | **bytes** | **kbytes** | **mbytes** | **ms** | **us**] | **percent** *percent_of_queuelimit*}
5. Repeat Step 4 to assign tail drop thresholds for other CoS values.
6. Repeat Steps 3 through 5 to assign tail drop thresholds for other queue classes.
7. **exit**
8. **show policy-map type queuing** *policy-map-name*
9. **copy running-config startup-config**

DETAILED STEPS

	Command	Purpose

Step 1	**config t**
	
	Example:
	`switch# config t`
	`switch(config)#`
		Enters configuration mode.

Step 2	**policy-map type queuing** [**match-first**] *policy-map-name*
	
	Example:
	`switch(config)# policy-map type queuing shape_queues`
	`switch(config-pmap-que)#`
		Configures the policy map of type queuing, and then enters policy-map mode for the policy-map name you specify. Policy-map names can contain alphabetic, hyphen, or underscore characters, are case sensitive, and can be up to 40 characters.

Step 3	**class type queuing** *class-queuing-name*
	
	Example:
	`switch(config)# class type queuing 1p3q4t-out-pq1`
	`switch(config-pmap-c-que)#`
		Configures the class map of type queuing, and then enters policy-map class queuing mode. Class queuing names are listed in Table 2-3.

Step 4	**queue-limit cos** *value* {*threshold* [**packets** | **bytes** | **kbytes** | **mbytes** | **ms** | **us**] | **percent** *percent_of_queuelimit*} **Example:** `switch(config-pmap-c-que)# queue-limit cos 5 10 mbytes`	Assigns a tail drop threshold based on queue size or percentage of the buffer memory used by the queue. The device will drop packets that exceed the specified threshold. You can configure the threshold by the number of packets, number of bytes, or the duration of time at the underlying interface minimum guaranteed link rate. The default threshold is in packets. The size is from 1 to 83886080. The duration is from 1 to 83886080. The percentage is from 1 to 100. The example shows how to set a tail drop threshold for packets with a CoS of 5 to a maximum size of 10 MB.
Step 5	(Optional) Repeat Step 4 to assign tail drop thresholds for other CoS values.	—

	Command	Purpose
Step 6	(Optional) Repeat Steps 3 through 5 to assign tail drop thresholds for other queue classes.	—

Step 7	**exit**
	
	Example:
	`switch(config-cmap-que)# exit`
	`switch(config)#`
		Exits policy-map queue mode and enters configuration mode.

Step 8	**show policy-map type queuing** *policy-map-name* **Example:** `switch(config)# show policy-map type queuing shape_queues`	(Optional) Displays information about all configured policy maps or a selected policy map of type queuing.
Step 9	**copy running-config startup-config** **Example:** `switch(config)# copy running-config startup-config`	(Optional) Saves the running configuration to the startup configuration.

Configuring WRED

Before configuring WRED, ensure that the CoS values are there (see the "Modifying Queuing Class Maps" section on page 7-6).

You can configure WRED on both ingress and egress queues to set minimum and maximum packet drop thresholds. The frequency of dropped packets increases as the queue size exceeds the minimum threshold. When the maximum threshold is exceeded, all packets for the CoS value are dropped.

Note You cannot configure WRED on ingress 10-Gigabit Ethernet ports.

You can configure WRED thresholds by CoS value, and configure a single WRED threshold to use on all CoS values that you do not specifically configure.

Note WRED and tail drop cannot be configured in the same class.

You use the system-defined queue classes for the type of module to which you want to apply the policy map. See Table 2-3 on page 2-7.

SUMMARY STEPS

1. **config t**
2. **policy-map type queuing** [**match-first**] *policy-map-name*
3. **class type queuing** *class-queuing-name*
4. **random-detect cos-based** [**aggregate** [**minimum-threshold**] {*min-threshold* [**packets** | **bytes** | **kbytes** | **mbytes** | **ms** | **us**] | **percent** *min-percent-of-qsize*} [**maximum-threshold**] {*max-threshold* [**packets** | **bytes** | **kbytes** | **mbytes** | **ms** | **us**] | **percent** *max-percent-of-qsize*}]
5. **random-detect** {**cos** *cos-list* [**minimum-threshold**] {*min-threshold* [**packets** | **bytes** | **kbytes** | **mbytes** | **ms** | **us**] | **percent** *min-percent-of-qsize*} [**maximum-threshold**] {*max-threshold* [**packets** | **bytes** | **kbytes** | **mbytes** | **ms** | **us**] | **percent** *max-percent-of-qsize*}}

6. Repeat Step 5 to configure WRED for other CoS values.
7. Repeat Steps 3 through 6 to configure WRED for other queuing classes.
8. **exit**
9. **show policy-map type queuing** *policy-map-name*
10. **copy running-config startup-config**

DETAILED STEPS

	Command	Purpose
Step 1	**config t** **Example:** switch# config t switch(config)#	Enters configuration mode.
Step 2	**policy-map type queuing** [**match-first**] *policy-map-name* **Example:** switch(config)# policy-map type queuing shape_queues switch(config-pmap-que)#	Configures the policy map of type queuing, and then enters policy-map mode for the policy-map name you specify. Policy-map names can contain alphabetic, hyphen, or underscore characters, are case sensitive, and can be up to 40 characters.
Step 3	**class type queuing** *class-queuing-name* **Example:** switch(config)# class type queuing 1p3q4t-out-pq1 switch(config-pmap-c-que)#	Configures the class map of type queuing, and then enters policy-map class queuing mode. Class queuing names are listed in Table 2-3.
Step 4	**random-detect cos-based** [**aggregate** [**minimum-threshold**] {*min-threshold* [**packets** | **bytes** | **kbytes** | **mbytes** | **ms** | **us**] | **percent** *min-percent-of-qsize*} [**maximum-threshold**] {*max-threshold* [**packets** | **bytes** | **kbytes** | **mbytes** | **ms** | **us**] | **percent** *max-percent-of-qsize*}]	Configures WRED for all CoS values not configured by a CoS-specific **random-detect** command. You can specify minimum and maximum thresholds used to drop packets from the queue. You can configure thresholds by the number of packets, number of bytes, the duration of time at the underlying interface minimum guaranteed link rate, or as the percentage of queue size. Minimum and maximum thresholds must be of the same type. If no aggregate arguments are supplied, then no aggregate WRED is configured. The default threshold is in packets. The thresholds are from 1 to 83886080. The percentage is from 1 to 100. **Note** You must enter this command, even if you enter the command with no values.
	Example 1: switch(config-pmap-c-que)# random-detect cos-based aggregate 10 mbytes 20 mbytes	Example 1 shows how to set the aggregate WRED thresholds for nonconfigured classes of traffic to a minimum of 10 MB and a maximum of 20 MB.
	Example 2: switch(config-pmap-c-que)# random-detect cos-based aggregate percent 10 percent 20	Example 2 shows how to set the aggregate WRED thresholds for nonconfigured classes of traffic to a minimum of 10 percent and a maximum of 20 percent of the queue size. **Note** You can specify only one **random-detect cos-based** command in a class.

	Command	Purpose
Step 5	**random-detect** {**cos** *cos-list* [**aggregate** [**minimum-threshold**] {*min-threshold* [**packets** | **bytes** | **kbytes** | **mbytes** | **ms** | **us**] | **percent** *min-percent-of-qsize*} [**maximum-threshold**] {*max-threshold* [**packets** | **bytes** | **kbytes** | **mbytes** | **ms** | **us**] | **percent** *max-percent-of-qsize*}}	(Optional) Configures WRED for specific CoS values. You can specify minimum and maximum thresholds used to drop packets from the queue. You can configure thresholds by the number of packets, number of bytes, the duration of time at the underlying interface minimum guaranteed link rate, or as the percentage of the queue size. Minimum and maximum thresholds must be of the same type. The default threshold is in packets. Thresholds are from 1 to 83886080. Percentage is from 1 to 100.
	Example 1: `switch(config-pmap-c-que)# random-detect cos 5,7 15 mbytes 20 mbytes`	Example 1 shows how to set the aggregate WRED thresholds for CoS values 5 and 7 to a minimum of 15 MB and a maximum of 20 MB.
	Example 2: `switch(config-pmap-c-que)# random-detect cos 5 percent 5 percent 15`	Example 2 shows how to set the aggregate WRED thresholds for CoS value 5 to a minimum of 5 percent and a maximum of 15 percent of the queue size.
Step 6	(Optional) Repeat Step 5 to configure WRED for other CoS values.	—
Step 7	(Optional) Repeat Steps 3 through 6 to configure WRED for other queuing classes.	—

Step 8	**exit**
	
	Example:
	`switch(config-cmap-que)# exit`
	`switch(config)#`
		Exits policy-map queue mode and enters configuration mode.

Step 9	**show policy-map type queuing** *policy-map-name* **Example:** `switch(config)# show policy-map type queuing shape_queues`	(Optional) Displays information about all configured policy maps or a selected policy map of type queuing.
Step 10	**copy running-config startup-config** **Example:** `switch(config)# copy running-config startup-config`	(Optional) Saves the running configuration to the startup configuration.

Configuring Congestion Management

You can configure only one of the following congestion management methods in a policy map:

- Allocate a minimum data rate to a queue by using the **bandwidth** and **bandwidth remaining** commands.
- Allocate all data for a class of traffic to a priority queue by using the **priority** command. You can use the **bandwidth remaining** command to distribute remaining traffic among the nonpriority queues. By default, the system evenly distributes the remaining bandwidth among the nonpriority queues.
- Allocate a maximum data rate to a queue by using the **shape** command.

In addition to the congestion management feature that you choose, you can configure one of the following queue features in each class of a policy map:

- Taildrop thresholds based on queue size and queue limit usage. For more information, see the "Configuring Tail Drop" section on page 7-8.
- WRED for preferential packet drops based on CoS. For more information, see the "Configuring WRED" section on page 7-10.

This section includes the following topics:

Configuring Bandwidth and Bandwidth Remaining

You can configure the bandwidth and bandwidth remaining on both ingress and egress queues to allocate a minimum percentage of the interface bandwidth to a queue. You use the system-defined ingress or egress queue class for the type of module to which you want to apply the policy map. See Table 2-3 on page 2-7 for the list of system-defined ingress or egress queue classes for each module.

Note If you configure bandwidth, you cannot configure priority or shaping in the same policy map.

SUMMARY STEPS

1. **config t**
2. **policy-map type queuing** [**match-first**] *policy-map-name*
3. **class type queuing** *class-queuing-name*
4. **bandwidth** {*rate* [**bps** | **kbps** | **mbps** | **gbps**] | **percent** *percent*}
   or
   bandwidth remaining percent *percent*
5. Repeat Steps 3 to 4 to assign bandwidth or bandwidth remaining for other queuing classes.
6. **exit**
7. **show policy-map type queuing** *policy-map-name*
8. **copy running-config startup-config**

DETAILED STEPS

	Command	Purpose
Step 1	**config t** **Example:** switch# config t switch(config)#	Enters configuration mode.
Step 2	**policy-map type queuing** [**match-first**] *policy-map-name* **Example:** switch(config)# policy-map type queuing shape_queues switch(config-pmap-que)#	Configures the policy map of type queuing, and then enters policy-map mode for the policy-map name you specify. Policy-map names can contain alphabetic, hyphen, or underscore characters, are case sensitive, and can be up to 40 characters.
Step 3	**class type queuing** *class-queuing-name* **Example:** switch(config)# class type queuing 1p3q4t-out-pq1 switch(config-pmap-c-que)#	Configures the class map of type queuing, and then enters policy-map class queuing mode. You must select one of the system-defined output queues. Class queuing names are listed in Table 2-3.
Step 4	**bandwidth** {*rate* [**bps** | **kbps** | **mbps** | **gbps**] | **percent** *percent*}	Assigns a minimum rate of the interface bandwidth to an output queue. You can configure a data rate by the bit rate or as the percentage of the underlying interface link rate. The default units are kbps. The data rate is from 1 to 10,000,000,000. The percentage is from 1 to 100. **Note** You can use only the **percent** keyword for interfaces set to autonegotiate.
	Example 1: switch(config-pmap-c-que)# bandwidth 10 mbps	Example 1 shows how to set the bandwidth to a minimum rate of 100 megabits per second (mbps).
	Example 2: switch(config-pmap-c-que)# bandwidth percent 25	Example 2 shows how to set the bandwidth to a minimum of 25 percent of the underlying link rate.
	bandwidth remaining percent *percent* **Example:** switch(config-pmap-c-que)# bandwidth remaining percent 25	(Optional) Assigns the percent of the bandwidth that remains to this queue. The range is from 0 to 100. The example shows how to set the bandwidth for this queue to 25 percent of the remaining bandwidth.

	Command	Purpose
Step 5	(Optional) Repeat Steps 3 to 4 to assign bandwidth or bandwidth remaining for other queuing classes.	—

Step 6	**exit**
	
	Example:
	`switch(config-cmap-que)# exit`
	`switch(config)#`
		Exits policy-map queue mode and enters configuration mode.

Step 7	**show policy-map type queuing** *policy-map-name* **Example:** `switch(config)# show policy-map type queuing shape_queues`	(Optional) Displays information about all configured policy maps or a selected policy map of type queuing.
Step 8	**copy running-config startup-config** **Example:** `switch(config)# copy running-config startup-config`	(Optional) Saves the running configuration to the startup configuration.

Configuring Priority

If you don't specify the priority, the system-defined egress **pq** queues behave like normal queues. (See Chapter 2, "Using Modular QoS CLI" for information on the system-defined type queuing class maps.)

You can configure only one level of priority on an egress priority queue. You use the system-defined priority queue class for the type of module to which you want to apply the policy map. See Table 2-3 on page 2-7 for the list of available system-defined, class maps for each module.

For the nonpriority queues, you can configure how much of the remaining bandwidth to assign to each queue. By default, the device evenly distributes the remaining bandwidth among the nonpriority queues.

Note If you configure priority, you cannot configure bandwidth or shaping in the same policy map.

SUMMARY STEPS

1. **config t**
2. **policy-map type queuing** [**match-first**] *policy-map-name*
3. **class type queuing** *class-queuing-name*
4. **priority** [**level** *value*]
5. **class type queuing** *class-queuing-name*
6. **bandwidth remaining percent** *percent*
7. Repeat Steps 5 to 6 to assign bandwidth remaining for the other nonpriority queues.
8. **exit**
9. **show policy-map type queuing** *policy-map-name*
10. **copy running-config startup-config**

DETAILED STEPS

	Command	Purpose
Step 1	**config t** **Example:** switch# config t switch(config)#	Enters configuration mode.
Step 2	**policy-map type queuing** [**match-first**] *policy-map-name* **Example:** switch(config)# policy-map type queuing priority_queue1 switch(config-pmap-que)#	Configures the policy map of type queuing, and then enters policy-map mode for the policy-map name that you specify. Policy-map names can contain alphabetic, hyphen, or underscore characters, are case sensitive, and can be up to 40 characters.
Step 3	**class type queuing** *class-queuing-name* **Example:** switch(config-pmap-que)# class type queuing 1p3q4t-out-pq1 switch(config-pmap-c-que)#	Configures the class map of type queuing, and then enters policy-map class queuing mode. You must select one of the system-defined priority queues. Class queuing names are listed in Table 2-3.
Step 4	**priority** [**level** *value*] **Example:** switch(config-pmap-c-que)# priority	Selects this queue as a priority queue. Only 1 priority level is supported.
Step 5	**class type queuing** *class-queuing-name* **Example:** switch(config-pmap-c-que)# class type queuing 1p3q4t-out-q2	(Optional) Configures the class map of type queuing, and then enters policy-map class queuing mode. Class queuing names are listed in Table 2-3. Choose a nonpriority queue where you want to configure the remaining bandwidth. By default, the system evenly distributes the remaining bandwidth among the nonpriority queues.
Step 6	**bandwidth remaining percent** *percent* **Example:** switch(config-pmap-c-que)# bandwidth remaining percent 25	(Optional) Assigns the percent of the bandwidth that remains to this queue. The range is from 1 to 100.

	Command	Purpose
Step 7	(Optional) Repeat Steps 5 to 6 to assign the remaining bandwidth for the other nonpriority queues.	—

Step 8	**exit**
	
	Example:
	`switch(config-cmap-que)# exit`
	`switch(config)#`
		Exits policy-map queue mode and enters configuration mode.

Step 9	**show policy-map type queuing** *policy-map-name* **Example:** `switch(config)# show policy-map type queuing priority_queue1`	(Optional) Displays information about all configured policy maps or a selected policy map of type queuing.
Step 10	**copy running-config startup-config** **Example:** `switch(config)# copy running-config startup-config`	(Optional) Saves the running configuration to the startup configuration.

Configuring Shaping

Note The device forces the shape rate to the closest value in the following percentage intervals: 100, 50, 33, 25, 12.5, 6.25, 3.13, or 1.07.

You can configure shaping on an egress queue to impose a maximum rate on it. You use the system-defined egress queue class for the type of module to which you want to apply the policy map. See Table 2-3 on page 2-7 for the list of available system-defined class maps for each module.

Note If you configure shaping, you cannot configure bandwidth or priority in the same policy map.

SUMMARY STEPS

1. **config t**
2. **policy-map type queuing** [**match-first**] *policy-map-name*
3. **class type queuing** *class-queuing-name*
4. **shape** [**average**] {*rate* [**bps** | **kbps** | **mbps** | **gbps**] | **percent** *percent*}
5. Repeat Steps 3 to 4 to configure shaping for other queuing classes.
6. **exit**
7. **show policy-map type queuing** *policy-map-name*
8. **copy running-config startup-config**

DETAILED STEPS

	Command	Purpose

Step 1	**config t**
	
	Example:
	`switch# config t`
	`switch(config)#`
		Enters configuration mode.

Step 2	**policy-map type queuing** [**match-first**] *policy-map-name*
	
	Example:
	`switch(config)# policy-map type queuing shape_queues`
	`switch(config-pmap-que)#`
		Configures the policy map of type queuing, and then enters policy-map mode for the policy-map name you specify. Policy-map names can contain alphabetic, hyphen, or underscore characters, are case sensitive, and can be up to 40 characters.

Step 3	**class type queuing** *class-queuing-name*
	
	Example:
	`switch(config)# class type queuing 1p3q4t-out-pq1`
	`switch(config-pmap-c-que)#`
		Configures the class map of type queuing and then enters policy-map class queuing mode. You must select one of the system-defined output queues. Class queuing names are listed in Table 2-3.

Step 4	**shape** [**average**] {*rate* [**bps** | **kbps** | **mbps** | **gbps**] | **percent** *percent*}
	
	Example 1:
	`switch(config-pmap-c-que)# shape 10 mbps`
	
	Example 2:
	`switch(config-pmap-c-que)# shape percent 25`
		Assigns a maximum rate on an output queue. You can configure a data rate by the bit rate or as a percentage of the underlying interface link rate. The default bit rate is in bits per second (bps). The data rate is from 8000 bps to 10 gbps. The percentage is from 1 to 100.
		Note You can use only the **percent** keyword for interfaces set to autonegotiate.
		Example 1 shows how to shape traffic to a maximum rate of 100 megabits per second (mbps).
		Example 2 shows how to shape traffic to a maximum of 25 percent of the underlying link rate.

Step 5	(Optional) Repeat Steps 3 to 4 to configure shaping for other queuing classes.	—

Step 6	*exit*
	
	Example:
	`switch(config-cmap-que)# exit`
	`switch(config)#`
		Exits policy-map queue mode and enters configuration mode.

Step 7	**show policy-map type queuing** *policy-map-name* **Example:** `switch(config)# show policy-map type queuing shape_queues`	(Optional) Displays information about all configured policy maps or a selected policy map of type queuing.
Step 8	**copy running-config startup-config** **Example:** `switch(config)# copy running-config startup-config`	(Optional) Saves the running configuration to the startup configuration.

Configuring Queue Limits

You can configure the queue limit on both ingress and egress queues. The device drops any packets over the queue limit. You use the system-defined queue classes for the type of module to which you want to apply the policy map. See Table 2-3 on page 2-7.

SUMMARY STEPS

1. **config t**
2. **policy-map type queuing** [**match-first**] *policy-map-name*
3. **class type queuing** *class-queuing-name*
4. **queue-limit** {*threshold* [**packets** | **bytes** | **kbytes** | **mbytes** | **ms** | **us**] | **percent** *percent_of_queuelimit*}
5. **exit**
6. **exit**
7. **show policy-map type queuing** *policy-map-name*
8. **copy running-config startup-config**

DETAILED STEPS

	Command	Purpose
Step 1	**config t** **Example:** switch# config t switch(config)#	Enters configuration mode.
Step 2	**policy-map type queuing** [**match-first**] *policy-map-name* **Example:** switch(config)# policy-map type queuing shape_queues switch(config-pmap-que)#	Configures the policy map of type queuing and then enters policy-map mode for the policy-map name you specify. Policy-map names can contain alphabetic, hyphen, or underscore characters, are case sensitive, and can be up to 40 characters.
Step 3	**class type queuing** *class-queuing-name* **Example:** switch(config)# class type queuing 1p3q4t-out-pq1 switch(config-pmap-c-que)#	Configures the class map of type queuing and then enters policy-map class queuing mode. Class queuing names are listed in Table 2-3.
Step 4	**queue-limit** {*threshold* [**packets** | **bytes** | **kbytes** | **mbytes** | **ms** | **us**] | **percent** *percent_of_queuelimit*} **Example:** switch(config-pmap-c-que)# queue-limit 10 mbytes	Assigns a queue limit based on queue size or percentage of the buffer memory used by the queue. The device will drop packets that exceed the specified threshold. You can configure the threshold by the number of packets, number of bytes, or the duration of time at the underlying interface minimum guaranteed link rate. The default threshold is in packets. The size is from 1 to 83886080. The duration is from 1 to 83886080. The percentage is from 1 to 100. The example shows how to set a queue limit to a maximum size of 10 MB.
Step 5	**exit** **Example:** switch(config-pmap-c-que)# exit switch(config-pmap-que)#	Exits class-map queue mode and enters policy-map queue mode.
Step 6	**exit** **Example:** switch(config-pmap-que)# exit switch(config)#	Exits policy-map queue mode and enters configuration mode.
Step 7	**show policy-map type queuing** *policy-map-name* **Example:** switch(config)# show policy-map type queuing shape_queues	(Optional) Displays information about all configured policy maps or a selected policy map of type queuing.
Step 8	**copy running-config startup-config** **Example:** switch(config)# copy running-config startup-config	(Optional) Saves the running configuration to the startup configuration.

Verifying Queuing and Scheduling Configuration

To display the queuing and scheduling configuration information, perform one of the following tasks:

Command	Purpose
show class-map type queuing [*class-queuing-name*]	Displays information about all configured class maps or a selected class map of type queuing. Class queuing names are listed in Table 2-3.
show policy-map type queuing *policy-map-name*	Displays information about all configured policy maps or a selected policy map of type queuing.

For detailed information about the fields in the output from these commands, see the *Cisco Nexus 7000 Series NX-OS Quality of Service Command Reference, Release 4.0*.

Example Configurations

In this section you can find examples of configuring queuing and scheduling.

This section includes the following topics:

Setting Ingress Port CoS Configuration Example

Note Setting the ingress port CoS value makes the specified interfaces untrusted.

Note Ensure that you are using the default queue for the port type that you are configuring. See Chapter 2, "Using Modular QoS CLI" for information on the default queue for the port types.

The following example shows how to configure ingress port CoS for 1-Gigabit Ethernet ports:

```
config t
  policy-map type queuing untrusted_port_cos
    class type queuing 2q4t-in-q-default
      set cos 5
  interface ethernet 2/1
    service-policy type queuing input untrusted_port_cos
```

The following example shows how to configure ingress port CoS for 10-Gigabit Ethernet ports:

```
config t
  policy-map type queuing untrusted_port_cos
    class type queuing 8q2t-in-q-default
      set cos 5
  interface ethernet 2/1
    service-policy type queuing input untrusted_port_cos
```

Priority and Queue Limit Configuration Example

The following example shows how to configure the priority and queue limit features:

```
config t
  class-map type queuing match-any 1p3q4t-out-pq1
    match cos 5-7
  class-map type queuing match-any 1p3q4t-out-q2
    match cos 3-4
  class-map type queuing match-any 1p3q4t-out-q3
    match cos 0-2
  policy-map type queuing priority_queue1
    class type queue 1p3q4t-out-pq1
      priority
    class type queue 1p3q4t-out-q2
      bandwidth remaining percent 60
      queue-limit 1 mbytes
    class type queue 1p3q4t-out-q3
      bandwidth remaining percent 40
      queue-limit 2 mbytes
```

Shaping and Tail Drop Configuration Example

The following example shows how to configure the shaping and taildrop features:

```
config t
  class-map type queuing match-any 1p3q4t-out-pq1
    match cos 5-7
  class-map type queuing match-any 1p3q4t-out-q2
    match cos 3-4
  policy-map type queuing shape_dt
    class type queue 1p3q4t-out-pq1
      shape percent 50
      queue-limit cos 5 percent 10
      queue-limit cos 6 percent 10
    class type queue 1p3q4t-out-q2
      shape percent 25
      queue-limit cos 4 percent 15
```

Note If the priority keyword is not specified for a **pq1** queue, the queue is just a normal queue, not a priority queue.

Bandwidth and WRED Configuration Example

The following example shows how to configure the bandwidth and WRED features:

```
config t
  class-map type queuing match-any 1p3q4t-out-pq1
```

```
  match cos 5-7
class-map type queuing match-any 1p3q4t-out-q2
  match cos 3-4
policy-map type queuing bandwidth_wred
  class type queuing 1p3q4t-out-pq1
    bandwidth percent 50
    random-detect cos-based
    random-detect cos 5 minimum-threshold percent 10 maximum-threshold percent 30
    random-detect cos 6 minimum-threshold percent 40 maximum-threshold percent 60
  class type queuing 1p3q4t-out-q2
    bandwidth percent 25
    random-detect cos-based
    random-detect cos 4 minimum-threshold percent 20 maximum-threshold percent 40
```

CHAPTER 8

Monitoring QoS Statistics

This chapter describes how to enable, display, and clear QoS statistics on the device.

This chapter includes the following sections:

Information about QoS Statistics

You can display various QoS statistics for the device. Statistics are enabled by default, but you can disable this feature. (See the "Example Display" section on page 8-3 for the statistics displayed.)

Licensing Requirements for Monitoring QoS Statistics

The following table shows the licensing requirements for this feature:

Product	License Requirement
NX-OS	QoS requires no license. Any feature not included in a license package is bundled with the Cisco NX-OS system images and is provided at no extra charge to you. For a complete explanation of the NX-OS licensing scheme, see the *Cisco Nexus 7000 Series NX-OS Licensing Guide, Release 4.0*.

However, using VDCs requires an Advanced Services license.

Prerequisites for Monitoring QoS Statistics

Monitoring QoS statistics has the following prerequisites:

- You must be familiar with Chapter 2, "Using Modular QoS CLI."
- You are logged on to the switch.

- You are in the correct virtual device context (VDC). A VDC is a logical representation of a set of system resources. You can use the **switchto vdc** command with a VDC number.

Enabling Statistics

You can enable or disable QoS statistics for all interfaces on the device. By default, QoS statistics are enabled.

SUMMARY STEPS

1. **config t**
2. **qos statistics**

   no qos statistics
3. **show policy-map interface**
   or
   show policy-map vlan
4. **copy running-config startup-config**

DETAILED STEPS

	Command	Purpose
Step 1	**config t** **Example:** switch# config t switch(config)#	Enters configuration mode.
Step 2	**qos statistics** **Example:** switch(config)# qos statistics	Enables QoS statistics on all interfaces.
	no qos statistics **Example:** switch(config)# no qos statistics	Disables QoS statistics on all interfaces.
Step 3	**show policy-map interface** **Example:** switch(config)# show policy-map interface	(Optional) Displays the statistics status and the configured policy maps on all interfaces.
	show policy-map vlan **Example:** switch(config)# show policy-map vlan	(Optional) Displays the statistics status and the configured policy maps on all VLANs.
Step 4	**copy running-config startup-config** **Example:** switch(config)# copy running-config startup-config	(Optional) Saves the running configuration to the startup configuration.

Displaying Statistics

You can display QoS statistics for all interfaces or a selected interface, data direction, or QoS type.

SUMMARY STEPS

1. **show policy-map** [*policy-map-name*] [**interface**] [**vlan**] [**input** | **output**] [**type** {**qos** | **queuing**}]

DETAILED STEPS

	Command	Purpose
Step 1	**show policy-map** [*policy-map-name*] [**interface**] [**vlan**] [**input** | **output**] [**type** {**qos** | **queuing**}] [{**class** [**type** {**qos** | **queuing**}] **Example:** `switch# show policy-map interface ethernet 2/1`	Displays statistics and the configured policy maps on all interfaces or the specified interface, all VLANs or specified VLANs, data direction, and QoS type.

Clearing Statistics

You can clear QoS statistics for all interfaces or a selected interface, data direction, or QoS type.

SUMMARY STEPS

1. **clear qos statistics** [**interface**] [**vlan**] [**input** | **output**] [**type** {**qos** | **queuing**}]

DETAILED STEPS

	Command	Purpose
Step 1	**clear qos statistics** [**interface**] [**vlan**] [**input** | **output**] [**type** {**qos** | **queuing**}] **Example:** `switch# clear qos statistics type qos`	Displays statistics and the configured policy maps on all interfaces or the specified interface, all VLANs or the specified VLANs, data direction, or QoS type.

Example Display

The following is an example of a display of the QoS statistics:

```
switch(config)# show policy-map interface ethernet 8/1


Global statistics status: enabled

Ethernet8/1

Service-policy (qos) input: pmap
policy statistics status: enabled

Class-map (qos): map (match-all)
```

```
0 packets, 0 bytes
5 minute offered rate 0 bps, drop rate 0 bps
Match: cos 0
police cir 10 mbps bc 200 ms
conformed 0 bytes, 0 bps action: transmit
violated 0 bytes, 0 bps action: drop

Class-map (qos): map1 (match-all)
0 packets, 0 bytes
5 minute offered rate 0 bps, drop rate 0 bps
Match: dscp 0
police cir 10 mbps bc 200 ms
conformed 0 bytes, 0 bps action: transmit
violated 0 bytes, 0 bps action: drop

Class-map (qos): map2 (match-all)
0 packets, 0 bytes
5 minute offered rate 0 bps, drop rate 0 bps
Match: precedence 5
police cir 20 mbps bc 200 ms
conformed 0 bytes, 0 bps action: transmit
violated 0 bytes, 0 bps action: drop

Class-map (qos): map3 (match-all)
0 packets, 0 bytes
5 minute offered rate 0 bps, drop rate 0 bps
Match: cos 3
police cir 30 mbps bc 200 ms
conformed 0 bytes, 0 bps action: transmit
violated 0 bytes, 0 bps action: drop

Class-map (qos): map4 (match-all)
0 packets, 0 bytes
5 minute offered rate 0 bps, drop rate 0 bps
Match: packet length 100
police cir 40 mbps bc 200 ms
conformed 0 bytes, 0 bps action: transmit
violated 0 bytes, 0 bps action: drop

Class-map (qos): map5 (match-all)
0 packets, 0 bytes
5 minute offered rate 0 bps, drop rate 0 bps
Match: access-group foo
police cir 50 mbps bc 200 ms
conformed 0 bytes, 0 bps action: transmit
violated 0 bytes, 0 bps action: drop

Class-map (qos): class-default (match-any)
0 packets, 0 bytes
5 minute offered rate 0 bps, drop rate 0 bps
police cir 60 mbps bc 200 ms
conformed 0 bytes, 0 bps action: transmit
violated 0 bytes, 0 bps action: drop
```

See the *Cisco NX-OS Quality of Service Command Reference* for complete information on the **show policy-map** command.

APPENDIX A

Configuration Limits for Cisco NX-OS Quality of Service Configuration Features, Release 4.0

The features supported by Cisco NX-OS have maximum configuration limits. For some of the features, we have verified configurations that support limits less than the maximum. Table A-1 lists the Cisco verified limits and maximum limits for Quality of Service (QoS) features using *Cisco Nexus 7000 Series NX-OS Release Notes, Release 4.0*.

Table A-1 Cisco NX-OS Release 4.0 QoS Configuration Limits

Feature	Maximum Limit
Policers	16384
Shared policers	16384
Number of policer profiles	4000
Number of classes	4000
Number of class-maps per policy	2000
Number of matches	1024
Number of policies	2000
Mutation table maps for ingress interfaces	14
Mutation table maps for egress interfaces	15
QoS groups	126
Classes in a queuing policy map	64
Classes in a qos marking/policing policy	4000

APPENDIX B

Additional References

This appendix contains additional information related to implementing QoS.

This appendix includes the following sections:

Related Documents

Related Topic	Document Title
VDCs	*Cisco Nexus 7000 Series NX-OS Virtual Device Context Configuration Guide, Release 4.0*
CLI commands	*Cisco Nexus 7000 Series NX-OS Quality of Service Command Reference, Release 4.0*

RFCs

RFCs	Title
RFC 2475	*Architecture for Differentiated Services*
RFC 2697	*A Single Rate Three Color Marker*
RFC 2698	*A Dual Rate Three Color Marker*
RFC 3289	*Management Information Base for the Differentiated Services Architecture*

INDEX

D

E

G

I

L

M

N

P

Q

U

V

W