Modular Quality of Service Command-Line Interface QoS on the Cisco CMTS Routers

First Published: December 18, 2008 Last Updated: July 11, 2012

The Modular Quality of Service Command-Line Interface (MQC) is designed to simplify the configuration of Quality of Service (QoS) on the Cisco CMTS routers by defining a common command syntax and resulting set of QoS behaviors across platforms.

Finding Feature Information

Your software release may not support all the features documented in this module. For the latest feature information and caveats, see the release notes for your platform and software release. To find information about the features documented in this module, and to see a list of the releases in which each feature is supported, see the Feature Information Table at the end of this document.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to http://tools.cisco.com/ITDIT/CFN/. An account on http://www.cisco.com/ is not required.

Contents

Prerequisites for MQC QoS

Table below shows the Cisco cable modem termination system (CMTS) hardware compatibility prerequisites for this feature.

Note The hardware components introduced in a given Cisco IOS Release will be supported in all subsequent releases unless otherwise specified.

Table 1: MQC QoS Support Hardware Compatibility Matrix

CMTS Platform	Processor Engine	Cable Interface Cards
Cisco uBR10012 Universal Broadband Router	**Cisco IOS Release 12.2(33)SCA and later releases releases** • PRE2 **Cisco IOS Release 12.2(33)SCB and later releases releases** • PRE4 **Cisco IOS Release 12.2(33)SCH and later releases** • PRE5	**Cisco IOS Release 12.2(33)SCB and later releases releases** • Cisco uBR10-MC5X20U/H **Cisco IOS Release 12.2(33)SCC and later releases releases** • Cisco UBR-MC20X20V **Cisco IOS Release 12.2(33)SCE and later releases releases** • Cisco uBR-MC3GX60V [1]
Cisco uBR7246VXR Universal Broadband Router	**Cisco IOS Release 12.2(33)SCA and later releases** • NPE-G1 • NPE-G2	**Cisco IOS Release 12.2(33)SCA and later releases** • Cisco uBR-MC28U/X **Cisco IOS Release 12.2(33)SCD and later releases** • Cisco uBR-MC88V [2]

CMTS Platform	Processor Engine	Cable Interface Cards
Cisco uBR7225VXR Universal Broadband Router	**Cisco IOS Release 12.2(33)SCA and later releases** • NPE-G1 **Cisco IOS Release 12.2(33)SCB and later releases** • NPE-G2	**Cisco IOS Release 12.2(33)SCA and later releases** • Cisco uBR-E-28U • Cisco uBR-E-16U • Cisco uBR-MC28U/X **Cisco IOS Release 12.2(33)SCD and later releases** • Cisco uBR-MC88V

1 The Cisco uBR-3GX60V cable interface line card is not compatible with PRE2.

2 The Cisco uBR-MC88V cable interface line card is compatible only with NPE-G2.

Note The combination of PRE4 and Cisco Half-Height Gigabit Ethernet (HHGE) is not supported in the same chassis.

Restrictions for MQC QoS

- The sum of all priority traffic running on a given port must be less than or equal to 90 percent of the port bandwidth.
- The 802.1p and Multiprotocol Label Switching (MPLS) Matching and Marking feature is currently supported only on Cisco uBR10012 routers.
- The MPLS Pseudowire for Cable L2VPN (Any Transport Over MPLS [AToM]) feature does not support matching and marking based on IP header fields. AToM supports only the matching and marking of experimental (EXP) bits.
- The Transport Layer Security (TLS) and cable dot1q L2VPN do not support the Input service policy and IPv6 service policy.
- PRE2 does not support IPv6 class of service (CoS) policy.
- MQC must be configured on the logical interface of a tunnel for the Cisco uBR10012 router.

Information About MQC QoS

Quality of Service (QoS) is supported on WAN interfaces using the standard MQC. The MQC CLI structure allows you to create traffic policies and attach these policies to interfaces. A traffic policy contains a traffic class and QoS features. A traffic class is used to select traffic, while the QoS features in the traffic policy determine how to treat the classified traffic.

Classifying Traffic

The Cisco uBR10012 Universal Broadband Router must differentiate traffic before it can apply appropriate QoS actions to the traffic. You can use an MQC CLI element called a *class map* to define traffic classification rules or criteria.

Class maps organize data packets into specific categories called classes that can receive user-defined QoS policies. The traffic class defines the classification rules for packets received on an interface.

Configuring QoS Policy Actions and Rules

After classifying the traffic, the Cisco uBR10012 Universal Broadband Router must be configured to handle the traffic that meets the matching criteria. The MQC CLI element *policy map* is used to create QoS policies and configure QoS actions and rules to apply to packets that match a particular traffic class.

A policy map associates a traffic class with one or more QoS actions. While configuring a policy map, you can specify a class map name and configure the actions you want the router to take on the matching traffic. However, before creating class policies in a policy map, the class classification criteria must be configured in a class map.

Whenever you modify a class policy of a policy map, class-based weighted fair queuing (CBWFQ) is notified and new classes are installed as part of the policy map in the CBWFQ system.

Attaching Service Policies to an Interface

After creating and configuring a traffic policy, you should attach the policy to an interface. A policy can be applied to packets in either direction, inbound or outbound. An interface can have different service policies for incoming and outgoing packets.

802.1p CoS

The 802.1p CoS feature introduces QoS-based matching and marking to VLAN user priority bits to provide QoS service on the Gigabit Ethernet WAN interface for 802.1q packets.

The 802.1p CoS marking is a QoS action like the “set ip precedence” that sets the user priority bits for traffic prioritization. CoS refers to the three bits in the VLAN header that is used to indicate the IEEE 802.1p priority of the Ethernet frame as it passes through a switched network.

Marking is a way to identify packet flows to differentiate them. Packet marking enables partitioning of the network into multiple priority levels, or classes of service. During network congestion, the priority marked packets are offered a higher priority than normal packets.

The 802.1p input packets are classified at eight different QoS levels (0 to 7) based on the VLAN user priority bits. The packet classification is specified through the MQC using ‘match’ statements within the class-map command.

On the Cisco CMTS router, 802.1p CoS matching is provided only for the input VLAN tagged frames. The user priority bits matching is not available for TLS and dot1q L2VPN packets.

For 802.1q output packets, QoS marking is done at the VLAN header to modify VLAN user priority bits. QoS services use these priority bit settings to gain traffic priority during times of congestion. For upstream TLS and dot1q L2VPN packets, user priority bits marking is done on the WAN interface.

Note For information on QoS, see Cisco IOS Release 12.0 Quality of Service Solutions Configuration Guide .

MPLS Short-Pipe

The MPLS Short-Pipe Mode feature introduces QoS-based matching and marking of MPLS EXP bits to provide QoS service on the WAN interface for MPLS packets. The three bit EXP define QoS treatment for a packet. The EXP bits support up to eight classes of traffic.

When an IP packet is sent from one site to another, the IP precedence field specifies QoS. Based on the IP precedence marking, the packet is given the treatment configured for that QoS. In an MPLS network, IP precedence value is copied to the MPLS EXP field during label imposition by default.

MPLS marking is a QoS action like the “set ip precedence”. Marking sets different values for the MPLS EXP field. This enables service providers to set the priority for packets transported through their networks. The packet classification criteria is specified through the MQC using ‘match’ statements within the class-map command.

MPLS CoS matching provides the QoS classification function based on the EXP bits of the label entry. For MPLS input packets, QoS classification is done to provide different levels of QoS based on the MPLS EXP bits. For MPLS output packets, the QoS marking is done at the MPLS label header to modify EXP bits.

Note IP ToS will be inactive when the MPLS EXP classification is active as both MPLS EXP and IP ToS shares the same field.

MPLS CoS treats AToM packets as general MPLS packets. For upstream AToM packets, marking is done for EXP bits on the imposition label. For downstream AToM packets, classification is done based on the EXP bits.

MPLS Tunneling

Tunneling is the ability of QoS to be transparent from one edge to the other edge of the network. A tunnel starts on label imposition, and ends at label disposition. When the label is stripped off, the packet goes out as an MPLS packet with a different Per-Hop Behavior (PHB) layer underneath or as an IP packet with and IP PHB layer.

MPLS QoS supports the following tunneling modes:

Uniform Mode

In this mode, packets are treated uniformly across the network. All the customers of the MPLS network use the same IP precedence settings. The IP precedence value and the MPLS EXP bits always correspond to the same PHB.

Short Pipe Mode

This mode provides a distinct MPLS PHB layer (on top of the IP PHB layer) across the entire MPLS network, allowing customers of a service provider to implement their own IP PHB marking scheme on the MPLS network.

For more information on tunneling, see DiffServ Tunneling Modes for MPLS Networks at http://www.cisco.com/en/US/tech/tk436/tk428/tech_tech_notes_list.html .

Input MQC Support on the Cable Bundle Interfaces

The Input MQC Support on the Cable Bundle Interfaces feature introduced on the Cisco CMTS from Cisco IOS Release 12.2(33)SCG onwards enables you to differentiate upstream traffic on the cable bundle or sub-bundle interface and set a corresponding 'qos-group'. This 'qos-group' is used at the output WAN interface to classify and set MPLS EXP bits that are different from the ToS and DSCP value of IP packets.

Note The term cable bundle is used to refer to both the cable bundle and sub-bundle interface in this document.

Table below lists the MQC match statements supported by the Input MQC Support on the Cable Bundle Interfaces feature on a cable bundle interface of the Cisco uBR10012 router.

Table 2: MQC Match Statements Supported on a Cable Bundle Interface of the Cisco uBR10012 Router

Match Statement	Purpose
access-group	Matches access group name or number.
any	Matches any packet.
dscp	Matches differentiated services code point (DSCP) for IPv4.
precedence	Matches precedence for IPv4.
input-interface	Matches interface specification. **Note** This option supports only bundle interface.
ip dcsp	Matches DSCP for IPv4.
ip precedence	Matches precedence for IPv4.
ip rtp	Matches Real-Time Transport Protocol (RTP) port numbers for IPv4.
not	Negates match result.

Match Statement	Purpose
qos-group	Matches QoS group. **Note** The 'qos-group' is internal to the router and should not be used to classify upstream traffic from external world.

Table below lists the MQC action statements supported by the Input MQC Support on the Cable Bundle Interfaces feature on a cable bundle interface of the Cisco uBR10012 router.

Table 3: MQC Action Statements Supported on a Cable Bundle Interface of the Cisco uBR10012 Router

Action Statement	Purpose
set dcsp	Sets DCSP value.
set ip dscp	Sets IP DSCP.
set ip precedence	Sets IP precedence.
set precedence	Sets precedence value.
set qos-group	Sets value from 0 to 99.

Table below lists the MQC match statements supported by the Input MQC Support on the Cable Bundle Interfaces feature on a cable bundle interface of the Cisco uBR7200 series routers.

Table 4: MQC Match Statements Supported on a Cable Bundle Interface of the Cisco uBR7200 Series Routers

Match Statement	Purpose
access-group	Matches access group name.
any	Matches any packet.
class-map	Matches class map name.
discard-class	Discards class identifier.
dscp	Matches DSCP in IPv4 and IPv6 packets.
input-interface	Matches an input interface. **Note** This option supports only the bundle interface.
ip dscp	Matches IP DSCP.
ip precedence	Matches IP precedence.

Match Statement	Purpose
ip rtp	Matches RTP port numbers.
mpls experimental	Matches MPLS experimental bits.
not	Negates match result.
packet length	Matches layer 3 packet length.
precedence	Matches precedence in IPv4 and IPv6 packets.
protocol	Matches protocol.
qos-group	Matches QoS group. **Note** The 'qos-group' is internal to the router and should not be used to classify upstream traffic from external world.
source-address	Matches source address.

Table below lists the MQC action statements supported by the Input MQC Support on the Cable Bundle Interfaces feature on a cable bundle interface of the Cisco uBR7200 series routers.

Table 5: MQC Action Statements Supported on a Cable Bundle Interface of the Cisco uBR7200 Series Routers

Action Statement	Purpose
set dscp	Sets DSCP value.
set ip dscp	Sets IP DSCP.
set ip precedence	Sets IP precedence value.
set precedence	Sets precedence value.
set qos-group	Sets value from 0 to 99.

How to Configure MQC QoS on the Cisco CMTS Routers

Note MQC support is applicable only to WAN interfaces as DOCSIS has its own QoS mechanism. However, DOCSIS QoS extends limited MQC support for cable interfaces to limit peer-to-peer (P2P) traffic.

This section describes the following required and optional procedures:

Configuring QoS Features Using MQC

To configure QoS features using the Modular QoS CLI, complete the following basic steps:

Step 1 Define a traffic class using the **class-map** command.

Step 2 Create a traffic policy by associating the traffic class with QoS features using the **policy-map** command.

Step 3 Attach the traffic policy to the interface using the **service-policy** command and specify whether the policy has to be applied to inbound or outbound traffic.

What to Do Next

Each of the above-mentioned steps is accomplished using a user interface command. Specifically, the three steps are accomplished through the use of three abstractions, class map, policy map, and service policy.

Note Service policies are applied to Gigabit Ethernet, Ten Gigabit Ethernet, 802.1Q VLAN subinterfaces, and tunnel interfaces. Tunnel interfaces are virtual interfaces without queues, and service policies applied to tunnels cannot contain queuing actions. The Cisco uBR10012 Universal Broadband Router does not support per-subinterface queues for VLAN subinterfaces. However, the VLANs share the main interface queues.

For more information about MQC, see the "Configuring the Modular Quality of Service Command-Line Interface" chapter of the http://www.cisco.com/en/US/docs/ios/12_2/qos/configuration/guide/qcfmcli2.html Cisco IOS Quality of Service Solutions Configuration Guide, Release 12.2 document.

Note Though MQC is not broadly supported on cable interfaces as most subscriber queue configuration is controlled by parameters in the cable modem configuration file, a subset of MQC is supported on cable interfaces. This allows multiple service operators (MSOs) to classify P2P traffic based on type of service (ToS) bits and send it to a shaped queue. The P2P traffic control feature can configure shape and queue-limit actions on the P2P traffic control policy map. The ToS P2P is supported only on legacy cable interfaces and not on Wideband or modular cable (MC) interfaces.

Configuring QoS Traffic Classes

The **class-map** command is used to create a traffic class. A traffic class contains three major elements: a name, a series of **match** commands, and, if more than one **match** command exists in the traffic class, an instruction on how to evaluate these **match** commands.

The **match** commands are used to specify various criteria for classifying packets. Packets are checked to determine whether they match the criteria specified in the **match** commands; if a packet matches the specified criteria, that packet is considered a member of the class and is forwarded according to the QoS specifications set in the traffic policy. Packets that fail to meet any of the matching criteria are classified as members of the default traffic class.

For more information about the default traffic class, refer to the "Configuring the Modular Quality of Service Command-Line Interface" chapter of the http://www.cisco.com/en/US/docs/ios/12_2/qos/configuration/guide/qcfmdcli.html Cisco IOS Quality of Service Solutions Configuration Guide, Release 12.2 document.

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**class-map** [**match-all** | **match-any**] *class-name* **Example:** `Router(config)# class-map class1`	Creates a class to be used with a class map, and enters class-map configuration mode. The class map is used for matching packets to the specified class. • **match-all—**(Optional) Specifies that all match criteria in the class map must be matched, using a logical AND of all matching statements defined under the class. This is the default. • **match-any—**(Optional) Specifies that one or more match criteria must match, using a logical OR of all matching statements defined under the class. • *class-name* —User-defined name of the class.
Step 4	**match** *type* **Example:** `Router(config-cmap)# match access-group 101`	Specifies the matching criterion to be applied to the traffic, where *type* represents one of the forms of the **match** command.
Step 5	end **Example:** `Router(config-cmap)# end`	Exits the class-map configuration mode and returns to privileged EXEC mode.

What to Do Next

Table below lists the match options supported on the class-map command.

Table 6: Match Options Supported on the MQC QoS

Command	Purpose
match access-group {*number* | *name*}	Specifies that the packet must be permitted by the specified access control list (ACL). • *number*—ACL identifier applied to an interface. Valid values are from 1 to 2699. • *name*—Packet with the indicated name must be permitted by the access list. The name can be a maximum of 40 alphanumeric characters.
match-all	Specifies that the packet must match all of the matching criteria defined for a class map.
match-any	Specifies that the packet must match at least one of the matching criteria defined for a class map.
match cos *cos-value* [*cos-value* [*cos-value* [*cos-value*]]]	Specifies that the packet must match on the basis of a Layer 2 CoS/Inter-Switch Link (ISL) marking. • *cos-value*— IEEE 802.1Q/ISL CoS value. The cos-value can range from 0 to 7; up to four CoS values, separated by a space, can be specified in one match cos statement.
match input-interface *name*	Specifies that the packet input interface must match the interface name. **Note** Matching is supported for cable bundles but not for physical cable interfaces.

Command	Purpose
match ip dscp {*ip-dscp-value* | **afxy** | **csx** | **ef** | **default**}	Specifies that the packet IP differentiated service code point (DSCP) value must match one or more of the specified attributes. • *ip-dscp-value*—*DSCP* value to match. Valid values are from 0 to 63. You can specify up to 8 code point values, using a space to separate consecutive values. Instead of specifying a numeric ip-dscp-value, you can specify one of the following reserved keywords: • **afxy**—Indicates assured forwarding points. The first number (x) indicates the AF class. Valid values are from 1 to 4. The second number (y) indicates the level of drop preference within each class. Valid values are from 1 (low drop) to 3 (high drop). • **csx**—Indicates class selector code points that are backward-compatible with IP precedence. Valid values for x are from 1 to 7. The CS code points (CS1 through CS7) are identical to IP precedence values from 1 to 7. • **ef**—Indicates expedited forwarding. • **default**—Indicates best effort or DSCP 0.
match ip precedence {*ip-precedence-value* | *precedence-name*}	Specifies that the packet IP precedence value must match one or more precedence values or the name of the precedence. • *ip-precedence-value* —IP precedence value to match. Valid values are from 0 to 7. You can specify up to 8 precedence values, using a space to separate consecutive values. • *precedence-name*—Name of the IP precedence value.

Command	Purpose
match ip rtp {*lowest-udp-port range* }	Specifies that the packet with even-numbered UDP port value must be within the specified range of port numbers. Only even-numbered ports are matched because they carry the real-time data streams. Odd-numbered ports are not matched because they only carry control information. • *lowest-udp-port*—Number specified from 0 to 65535 and is the lowest number in the range. • *range*—Number specified from 0 to 65535 and is the highest number in the range.
match mpls experimental topmost *value*	Matches the experimental (EXP) value in the topmost label. • *value*—Value to which you want to set the MPLS EXP bits in the topmost label header. Valid values are from 0 to 7.
match not *criteria*	Specifies that the packet must not match this particular matching criterion value. • *criteria*—Match criterion value that should be an unsuccessful match criteria. All other values of the specified match criterion are considered successful match criteria.
match qos-group *number*	Specifies that the packet QoS group number value must match the specified QoS group number. • *number*—Group number specified from 0 to 99.

Configuring Traffic Policies

After creating traffic classes, you can configure traffic policies to configure marking features to apply certain actions to the selected traffic in those classes.

The **policy-map** command is used to create a traffic policy. The purpose of a traffic policy is to configure the QoS features that should be associated with the traffic that has been classified in a user-specified traffic class.

Note A packet can match only one traffic class within a traffic policy. If a packet matches more than one traffic class in the traffic policy, the first traffic class defined in the policy will be used.

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**policy-map***policy-map-name* **Example:** `Router(config)# policy-map policy9`	Creates or modifies a traffic policy and enters policy map configuration mode, where: • *policy-map-name*—Name of the traffic policy to configure. Names can be a maximum of 40 alphanumeric characters.
Step 4	**class** {*class-name* | **class-default**} **Example:** `Router(config-pmap)# class class1`	Specifies the name of the traffic class to which this policy applies and enters policy-map class configuration mode: • *class-name*—Policy applied to a user-defined class name previously configured. • **class-default**—Specifies that the policy applies to the default traffic class.
Step 5	end **Example:** `Router(config-pmap)# end`	Exits the policy-map class configuration mode and returns to privileged EXEC mode.

Defining QoS Actions in a Policy Map

Action commands can be added from within class mode on a policy map. Action commands fall into three general categories as given below:

Set Actions

Set commands allow traffic to be marked such that other network devices along the forwarding path can quickly determine the proper class of service to apply to a traffic flow. Set commands can be applied to both input and output policy maps.

Table below lists the set options supported on the Cisco uBR10012 Universal Broadband Router.

Table 7: Set Options Supported on the MQC QoS

Command	Purpose
set cos {*cos-value* | *from-field* [**table** *table-map-name*]}	Sets the Layer 2 CoS value of an outgoing packet. • *cos-value*—IEEE 802.1Q CoS value. The valid range is from 0 to 7. • *from-field*—Packet-marking category used to set packet CoS value. If a table map is used for mapping and converting packet-marking values, this establishes the "map from" packet-marking category. Packet-marking category keywords are precedence and dscp. • **table**—(Optional) Sets the values specified in a table that is used to set the CoS value. • *table-map-name*—(Optional) Name of the table map used to specify the CoS value. Maximum of 64 alphanumeric characters.
set ip dscp {*ip-dscp-value* | **afxy** | **csx** | **ef** | **default**}	Marks a packet with the differentiated services code point (DSCP) you specify. Valid values are from 0 to 63. Instead of specifying a numeric ip-dscp-value, you can specify one of the following reserved keywords: • **afxy**—Indicates assured forwarding points. The first number (x) indicates the AF class. Valid values are from 1 to 4. The second number (y) indicates the level of drop preference within each class. Valid values are from 1 (low drop) to 3 (high drop). • **csx**—Indicates class selector code points that are backward-compatible with IP precedence. Valid values for x are from 1 to 7. The CS code points (CS1 through CS7) are identical to IP precedence values from 1 to 7. • **ef**—Indicates expedited forwarding. • **default**—Indicates best effort or DSCP 0.
set ip precedence {*precedence-value* }	Marks a packet with the IP precedence level you specify. Valid values are from 0 to 7.

Command	Purpose
set mpls experimental topmost {*mpls-exp-value* | **qos-group** [**table** *table-map-name*]}	Set the MPLS EXP field value in the topmost label on either an input or an output interface. • *mpls-exp-value*—Value used to set the MPLS EXP bits defined by the policy map. The valid values range from 0 to 7. • **qos-group**—Specifies that the qos-group packet-marking category is used to set the MPLS EXP imposition value. If you are using a table map for mapping and converting packet-marking values, this establishes the "map from" packet-marking category. • **table**—(Optional) Used in conjunction with the **qos-group** keyword. Indicates that the values set in a specified table map will be used to set the MPLS EXP value. • *table-map-name* —(Optional) Name of the table map used to specify the MPLS EXP value. Used in conjunction with the table keyword. The name can be a maximum of 64 alphanumeric characters.
set qos group *group-id*	Marks a packet with the QoS group identifier you specify. The valid values range from 0 to 99.

Police Actions

Traffic policing is a traffic regulation mechanism that is used to limit the rate of traffic streams. Policing allows you to control the maximum rate of traffic sent or received on an interface. Policing propagates bursts of traffic and is applied to the inbound or outbound traffic on an interface. When the traffic rate exceeds the configured maximum rate, policing drops or remarks the excess traffic. Although policing does not buffer excess traffic, in the output direction, a configured queuing mechanism applies to conforming packets that might need to be queued while waiting to be serialized at the physical interface.

Traffic policing uses a token bucket algorithm to manage the maximum rate of traffic. This algorithm is used to define the maximum rate of traffic allowed on an interface at a given moment in time. The algorithm puts tokens into the bucket at a certain rate. Each token is permission for the source to send a specific number of bits into the network. With policing, the token bucket determines whether a packet exceeds or conforms to the applied rate. In either case, policing implements the action you configure such as setting the IP precedence or differentiated services code point (DSCP).

To configure traffic policing based on bits per second, use the **police** command in policy-map class configuration mode.

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configureterminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**policy-map** [**name**] **Example:** `Router(config)# policy-map policy9`	Specifies the traffic policy and enters policy-map configuration mode.
Step 4	**class** [**name**] **Example:** `Router(config-pmap)# class class1`	Specifies the name of the traffic class to which this policy applies and enters policy-map class configuration mode.
Step 5	police [bps] [burst-normal] [burst-excess] conform [conform-action] exceed [exceed-action] **Example:** `Router(config-pmap-c)# police 10000000 15000 20000 conform transmit exceed drop`	Configures traffic policing based on bits per second. • *bps*—(Optional) Average rate in bits per second (bps). Valid values are from 8,000 to 2,488,320,000 bps. If you only specify police bps, the router transmits the traffic that conforms to the bps value and drops the traffic that exceeds the bps value. • *burst-normal*—(Optional) Normal or committed burst size used by the first token bucket for policing. The burst-normal option specifies the committed burst in bytes. Valid values are from 1 to 512,000,000. The default is 9,216 bytes. • *burst-excess*—(Optional) Excess burst size used by the second token bucket for policing. The burst-excess option specifies the excess burst in bytes. Valid values are from 0 to 1,024,000,000 bytes. The default is 0. You must specify burst-normal before you specify burst-excess. **Note** When the burst-excess value equals 0, we recommend that you set the egress burst-normal value to be greater than or equal to the ingress burst-normal value plus 1. Otherwise, packet loss can occur. For example: burst-excess = 0; egress burst-normal >= ingress burst-normal + 1. • *conform-action*—Action to take on packets that conform to the rate limit. The default action is transmit. You must specify burst-excess before you specify conform. • *exceed-action* —Action to take on packets that exceed the rate limit. The default action is drop. You must specify conform before you specify exceed.

	Command or Action	Purpose
Step 6	**end** **Example:** `Router(config-pmap-c)# end`	Exits the policy-map class configuration mode and returns to privileged EXEC mode.

Queuing Actions

When queuing actions are applied to a given class within a policy map, they either cause queues to be created for that particular class of traffic or control how the queues are managed. Queuing commands are valid only in the output direction.

The Cisco uBR10012 Universal Broadband Router supports the MQC policy maps for class queue creation on WAN interfaces.

The following two types of queues are supported through MQC:

- Priority queues—Used mainly for voice traffic. They are policed at their individual committed information rate (CIR) to limit their bandwidth to the subscribed level. Only one priority queue is allowed per logical interface.
- Class queues—Implemented as best effort queues. They are based on a specified bandwidth in Kbps and shaped using the "bandwidth" policy map action. Generally, the specified bandwidth is not guaranteed.

Weighted random early detection (WRED) is a mechanism for controlling congestion of queues. WRED combines the capabilities of the random early detection (RED) mechanism with IP precedence, DSCP, and discard class to provide preferential handling of higher priority packets. For additional information on WRED, refer to the Cisco IOS Quality of Service Solutions Configuration Guide, Release 12.2 .

Note Cisco IOS Release 12.2(33)SCB does not support random-detect for type of service (ToS) peer-to-peer (P2P) policy maps.

Table below lists the **queuing actions** supported on the Cisco uBR10012 Universal Broadband Router.

Table 8: Queuing Actions Supported on the MQC QoS

Command	Purpose
priority	Assigns priority to the class you specified and reserves a priority queue for class-based weighted fair queuing (CBWFQ) traffic. The **priority** command does not have any arguments. You must use the **police** command to specify a guaranteed bandwidth.

Command	Purpose
queue-limit number-of-packets	Specifies or modifies the maximum number of packets that a particular class queue can hold.
random-detect dscp-based	Configures WRED to drop packets based on a DSCP value.
random-detect precedence-based	Configures WRED to drop packets based on an IP precedence value.
random-detect dscp dscp-values sub-class-val1 [...[sub-class-val8]]minimum-thresh *min-thresh-value* **maximum-thresh** *max-thresh-value* **mark-prob** *mark-prob-value*	Configures the minimum and maximum packet thresholds for the differentiated services code point (DSCP) value. • dscp-values—DSCP value. The DSCP value can be a number from 0 to 63. • **min-thresh-value**—Minimum threshold in number of packets. The value range of this argument is from 1 to 4096. • **max-thresh-value**—Maximum threshold in number of packets. The value range of this argument is from the value of the **min-thresh-value** argument to 4096. • **max-prob-value**—Specifies the fraction of packets dropped when the average queue depth is at the maximum threshold.
random-detect precedence values *sub-class-val1* [...[*sub-class-val8*]] **minimum-thresh** *min-thresh-value* **maximum-thresh** *max-thresh-value* **mark-prob** *mark-prob-value*	Configures WRED and distributed WRED (DWRED) parameters for a particular IP Precedence. Valid values are from 0 to 7. Typically, 0 represents low priority traffic that can be aggressively managed (dropped) and 7 represents high priority traffic. • *min-thresh-value*—Minimum threshold in number of packets. The value range of this argument is from 1 to 4096. • *max-thresh-value*—Maximum threshold in number of packets. The value range of this argument is from the value of the min-thresh-value argument to 4096. • *mark-prob-value* —Fraction of packets dropped when the average queue depth is at the maximum threshold.

Command	Purpose
shape [**average**]*cir*	Shapes traffic to the rate you specify, or shapes traffic based on the percentage of available bandwidth you specify. • **average—**Specifies the committed burst (Bc) that specifies the maximum number of bits sent out in each interval. • *cir***—**Committed information rate (CIR), in bits per second (bps).
bandwidth {*bandwidth-kbps* | **percent** *percentage* | **remaining percent** *percentage*}	Specifies or modifies the minimum bandwidth allocated for a traffic class in a policy map. • *bandwidth-kbps***—**Minimum bandwidth allocated for a class belonging to a policy map. Accepted input values are from 8 to 10,000,000,000 although the maximum value entered should not be larger than the link bandwidth of the slowest interface to which the policy will be applied. • **percent** *percentage* **—**Specifies or modifies the minimum percentage of the link bandwidth allocated for a class belonging to a policy map. Valid values are from 1 to 100. • **remaining percent** *percentage***—**Specifies or modifies the minimum percentage of unused link bandwidth allocated for a class belonging to a policy map. Valid values are from 1 to 100. **Note** Configure the amount of bandwidth large enough to also accommodate Layer 2 overhead.

Attaching Service Policies

The **service-policy** command is used to attach the traffic policy, as specified with the **policy-map** command, to an interface. Because the elements of the traffic policy can be applied to packets entering and leaving the interface, it is essential to specify whether the traffic policy characteristics should be applied to incoming or outgoing packets.

To attach a policy map that the router can use to apply QoS policies to inbound and outbound packets, use the **service-policy** command in interface or map class configuration mode.

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configureterminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**interface** *interface-name* **Example:** `Router(config)# interface GigabitEthernet 3/0/0`	Specifies the interface and enters interface configuration mode.
Step 4	Router(config-if)# service-**policy** {*input* | *output*} *policy-map-name* **Example:** `Router(config-if)# service-policy output policy1`	Specifies a policy map that the router can use to apply QoS policies to inbound or outbound packets. • *input* —Applies the QoS policy to inbound packets. • *output*—Applies the QoS policy to outbound packets. • *policy-map-name*—Name of the policy map (created using the policy-map command) you want to attach. The policy-map-name can be a maximum of 40 alphanumeric characters.

Configuring Output Rate

To restrict the WAN interface bandwidth output rate to a smaller value than that of the physical link bandwidth, use the output-rate command in interface configuration mode.

Note The output-rate command is valid only for Gigabit Ethernet interfaces.

Note Starting with Cisco IOS Release 12.2(33)SCG, the **output-rate** command is not supported and the value 10,000 is used for the output line rate on a Cisco uBR10012 router.

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**interface interface-name** **Example:** `Router(config)# interface GigabitEthernet 3/0/0`	Specifies the interface and enters interface configuration mode.
Step 4	**output-rate** *rate* **Example:** `Router(config-if)# output-rate 100`	Specifies a custom-defined output rate to a WAN interface instead of the default line rate. • *rate* —Output rate defined for the WAN interface, in kilobits per second. Valid values range from 1 to 1,000,000.
Step 5	exit **Example:** `Router(config-if)# exit`	Exits the interface configuration mode and returns to privileged EXEC mode.

Configuring Input MQC Support on the Cable Bundle Interfaces

To configure input MQC on a cable bundle interface to differentiate upstream traffic and set corresponding “qos-group” features, follow the steps given below.

Restriction

- QoS actions like policing, shaping, WRED, and queuing are not supported.
- Input MQC cannot be configured on cable physical interfaces.

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**class-map match-all** *class-name* **Example:** `Router(config)# class-map match-all class-ip-prec-6`	Creates a class to be used with a class map, and enters class-map configuration mode. The class map is used for matching packets to the specified class. • **match-all—**Specifies that all match criteria in the class map must be matched, using a logical AND of all matching statements defined under the class. This is the default option. • *class-name***—**User-defined name of the class.
Step 4	**match ip precedence** *ip-precedence-value* **Example:** `Router(config-cmap)# match ip precedence 6`	Specifies the IP precedence values as match criteria. • *ip-precedence-value* **—**IP precedence value. The valid values range from 0 to 7.
Step 5	exit **Example:** `Router(config-cmap)# exit`	Exits the class-map configuration mode and returns to global configuration mode.
Step 6	Repeat Step 3, on page 23 and Step 4, on page 23 to define the second class map.	
Step 7	**policy-map** *policy-map-name* **Example:** `Router(config)# policy-map policy-input`	Creates a policy map that can be attached to one or more interfaces to specify a service policy and enters policy map configuration mode. • *policy-map-name* **—**Name of the policy map. The name can be up to 40 alphanumeric characters.
Step 8	**class** *class-name* **Example:** `Router(config-pmap-c)# class`	Specifies the name of the class for which to create a policy and enters the policy-map class configuration mode. • *class-name***—**Name of the class to configure.

	Command or Action	Purpose
	`class-ip-prec-6`	
Step 9	**set qos-group** *group-id* **Example:** `Router(config-pmap-c)# set qos-group 6`	Sets a group ID that can be used later releases to classify packets. • *group-id*—Group identifier number. The valid range is from 0 to 99.
Step 10	exit **Example:** `Router(config-pmap-c)# exit`	Exits the policy-map class configuration mode and returns to global configuration mode.
Step 11	**interface bundle** *bundle-number* **Example:** `Router(config)# interface bundle 1`	Configures a cable bundle interface and enters interface configuration mode. • *bundle-number* —Cable bundle interface identification number.
Step 12	**service-policy input** *policy-map-name* **Example:** `Router(config-if)# service-policy input policy-input`	Attaches a policy map to an input interface that is used as the service policy for the interface • *input* —Attaches the specified policy map to the input interface. • *policy-map-name* —Name of the service policy map (created using the **policy-map** command) to be attached. The name can be up to 40 alphanumeric characters
Step 13	end **Example:** `Router(config-pmap-c)# end`	Exits the interface configuration mode and returns to privileged EXEC mode.

Configuration Examples for MQC QoS

This section provides the following configuration examples:

Example: Configuring the Traffic Class

The following example shows how to create two traffic classes and specify their match criteria. For the first traffic class called class1, access control list (ACL) 101 is used as the match criteria. For the second traffic

class called class2, ACL 102 is used as the match criteria. Packets are checked against the contents of these ACLs to determine if they belong to the class.

```
Router(config)# class-map class1

Router(config-cmap)# match access-group 101
Router(config-cmap)# exit

Router(config)# class-map class2

Router(config-cmap)# match access-group 102

Router(config-cmap)# exit
```

Example: Configuring the Traffic Policy

The following example shows how to define a traffic policy called policy1 that contains policy specifications for class1.

```
Router(config)# policy-map policy1
Router(config-pmap)# class class1
Router(config-pmap-c)# bandwidth 3000

Router(config-pmap-c)# queue-limit 30

Router(config-pmap)# exit
```

Example: Attaching the Service Policy

The following example shows how to attach an existing traffic policy to an interface. After you define a traffic policy with the **policy-map** command, you can attach it to one or more interfaces by using the **service-policy** command in interface configuration mode. Although you can assign the same traffic policy to multiple interfaces, each interface can have only one traffic policy attached at the input and only one traffic policy attached at the output.

```
Router(config)# interface GigabitEthernet 3/0/0
Router(config-if)# service-policy output policy1

Router(config-if)# exit
```

Example: Verifying QoS Policy

The following example shows how to verify a policy map configuration by entering any of the following commands in privileged EXEC mode.

```
Router# show policy-map policy-map-name class class-name
Class foobar
      bandwidth percent 20
       packet-based wred, exponential weight 9
      random-detect aggregate
      random-detect precedence values 2 minimum-thresh 1024 maximum-thresh 20481
```

Example: Configuring Input MQC Support on the Cable Bundle Interfaces

The following example shows how to differentiate upstream traffic on the cable bundle interfaces and set corresponding “qos-group” features.

```
!Define two different class maps.
class-map match-all class-ip-prec-6
  match ip precedence 6
class-map match-all class-ip-prec-7
  match ip precedence 7
!Define a policy map.
policy-map policy-input
  class class-ip-prec-6
   set qos-group 6
  class class-ip-prec-7
   set qos-group 7
!Attach the policy map to the cable bundle interface in the input direction.
configure terminal
 interface bundle 1
  service-policy input policy-input
```

How to Configure 802.1p CoS and MPLS EXP on the Cisco CMTS Routers

This section describes the following required procedures:

Configuring 802.1p CoS Matching

Use the following procedure to specify a class-map and match a packet based on the CoS marking.

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configureterminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**class-map** *class-map-name*— **Example:** `Router(config)# class-map cos1`	Specifies the class name used for the class in the policy map. • *class-map-name*— Name of the class for the class map.

	Command or Action	Purpose
Step 4	**match cos***cos-value* **Example:** `Router(config-cmap)# match cos 0`	Enters the class-map configuration mode and specifies the class of service that needs to match the class map. • *cos-value*— Packet CoS bit value. The valid values range from 0 to 7. You can specify up to four CoS values in one **match cos** statement.
Step 5	end **Example:** `Router(config-cmap)# end`	Exits the class-map configuration mode and returns to privileged EXEC mode.

Configuring 802.1p CoS Marking

Use the following procedure to specify a policy-map and associate a map class with it to set the CoS value for an outgoing packet.

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configureterminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**policy-map policy-map-name** **Example:** `Router(config)# policy-map cos0`	Specifies a policy map name. • *policy-map-name*—Policy map name.
Step 4	**class name** **Example:** `Router(config-pmap)# class cos1`	Enters the policy-map configuration mode and specifies the map class to which the packets has to be matched. • *name* —Map class name.

	Command or Action	Purpose
Step 5	set cos *cos-value* **Example:** `Router(config-pmap-c)# set cos 2`	Enters the policy-map class configuration mode and specifies a CoS value to associate with the packet. • *cos-value*—Class of service value. The valid values range from 0 to 7.
Step 6	**end** **Example:** `Router(config-pmap-c)# end`	Exits the policy-map class configuration mode and returns to privileged EXEC mode.

Configuring MPLS EXP Matching

Use the following procedure to specify a class-map and match a packet based on the EXP marking.

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configureterminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**class-map** *class-map-name* **Example:** `Router(config)# class-map exp7`	Specifies the class name used for the class in the policy map. • *class-map-name*—Name of the class for the class map.
Step 4	**match mpls experimental topmost** *number* **Example:** `Router(config-cmap)# match mpls experimental topmost 2`	Enters the class-map configuration mode and specifies the MPLS EXP field in the topmost label header. • *number*—MPLS EXP field number. The valid values range from 0 to 7.

	Command or Action	Purpose
Step 5	end **Example:** `Router(config-cmap)# end`	Exits the class-map configuration mode and returns to privileged EXEC mode.

Configuring MPLS EXP Marking

Use the following procedure to specify a policy-map and associate a map class set to the EXP value for an outgoing packet.

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configureterminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**policy-map** *policy-map-name* **Example:** `Router(config)# policy-map exp5`	Specifies a policy map name. • *policy-map-name*—Policy map name.
Step 4	**class name** **Example:** `Router(config-pmap)# class exp7`	Enters the policy-map configuration mode and specifies the map class to which the packets has to be matched. • *name*— Map class name.
Step 5	t **set mpls experimental topmos***number* **Example:** `Router(config-pmap-c)# set mpls experimental topmost 2`	Enters the policy-map class configuration mode and sets the MPLS EXP field in the topmost label header. • *number* —MPLS EXP field number. The valid values range from 0 to 7.

	Command or Action	Purpose
Step 6	**end** **Example:** `Router(config-pmap-c)# end`	Exits the policy-map class configuration mode and returns to privileged EXEC mode.

Configuration Examples for 802.1p CoS and MPLS EXP Matching and Marking

This section provides the following configuration examples:

Example: Configuring 802.1p CoS Matching

The following example shows how to match traffic classes within the 802.1p domain using packet CoS values.

```
Router> enable
Router# configure terminal
Router(config)# class-map cos1
Router(config-cmap)# match cos 0
Router(config-cmap)# end
```

Example: Configuring 802.1p CoS Marking

The following example shows how to define traffic classes within the 802.1p domain using packet CoS values.

```
Router> enable
Router# configure terminal
Router(config)# policy-map cos2
Router(config-pmap)# class cos1
Router(config-pmap)# set cos 2
Router(config-pmap)# end
```

Example: Configuring MPLS EXP Matching

The following example shows how to match traffic classes within the MPLS domain using packet EXP values.

```
Router> enable
Router# configure terminal
Router(config)# class-map exp1
Router(config-cmap)# match mpls experimental topmost 2
Router(config-cmap)# end
```

Example: Configuring MPLS EXP Marking

The following example shows how to define traffic classes within the MPLS domain using packet EXP values.

```
Router> enable
Router# configure terminal
Router(config)# policy-map exp2
Router(config-pmap)# class exp1
Router(config-pmap)# set mpls experimental topmost 2
Router(config-pmap)# end
```

Verifying 802.1p CoS and MPLS EXP Matching and Marking

This section explains how to verify 802.1p CoS and MPLS EXP configurations on the Cisco CMTS router:

Example: Verifying Matching and Marking

The following example shows how to verify the matching and marking configuration by entering the following command in privileged EXEC mode:

```
Router# ping vrf vrfa 1.3.99.98
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 1.3.99.98, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 8/10/20 ms
```

Example: Verifying QoS and Service Flow

The following example shows how to verify the QoS and service flow information for a particular cable modem (CM) by entering the following command in privileged EXEC mode:

```
Router# show cable modem 1.3.99.98 qos
Load for five secs: 1%/0%; one minute: 1%; five minutes: 1%
Time source is hardware calendar, *23:01:41.873 pst Thu Mar 3 2011
Sfid  Dir Curr  Sid   Sched  Prio MaxSusRate  MaxBrst     MinRsvRate  Throughput
          State       Type
49    US  act   31    BE     7    300000000   1000000     0           1968000
50    DS  act   N/A   BE     0    300000000   3044        0           1969253
```

Example: Verifying Input and Output Service Policies

The following example shows how to verify the statistics and the configurations of the input and output service policies that are attached to an interface by entering the following command in privileged EXEC mode:

```
Router# show policy-map interface gigabitEthernet 1/2/0
Load for five secs: 1%/0%; one minute: 1%; five minutes: 1%
Time source is hardware calendar, *23:02:40.857 pst Thu Mar 3 2011
 GigabitEthernet1/2/0
  Service-policy input: policy-in
    Class-map: class-exp-0 (match-all)
      6647740 packets, 9304674796 bytes
      30 second offered rate 3234000 bps, drop rate 0 bps
      Match: mpls experimental topmost 0
```

```
      QoS Set
        precedence 3
          Packets marked 6647740
    Class-map: class-default (match-any)
      1386487 packets, 1903797872 bytes
      30 second offered rate 658000 bps, drop rate 0 bps
      Match: any
  Service-policy output: policy-out
    Class-map: class-pre-1 (match-all)
      2041355 packets, 2857897000 bytes
      30 second offered rate 986000 bps, drop rate 0 bps
      Match: ip precedence 1
      QoS Set
        mpls experimental topmost 1
          Packets marked 2041355
    Class-map: class-default (match-any)
      6129975 packets, 8575183331 bytes
      30 second offered rate 2960000 bps, drop rate 0 bps
      Match: any
```

Example: Verifying PXF QoS

The following example shows how to verify the Parallel eXpress Forwarding (PXF) CPU statistics and QoS for a particular interface by entering the following command in privileged EXEC mode:

```
Router# show pxf cpu statistics qos gigabitEthernet 1/2/0
Load for five secs: 1%/0%; one minute: 1%; five minutes: 1%
Time source is hardware calendar, *23:04:11.893 pst Thu Mar 3 2011
   Classmap           Match         Pkts           Bytes
    Index             Number        Matched        Matched
-------------- --------------- --------------- -------------
 policy-in (Input) service-policy :
class-exp-0     (0)     0               6674330         751960948
class-default   (1)     0               1392134         1911548970
 policy-out (Output) service-policy :
class-pre-1     (0)     0               2049682         2869554800
class-default   (1)     0               6154976         20224030
```

Example: Verifying MPLS Virtual Circuits

The following example shows how to verify the state of all virtual circuits associated with an MPLS pseudowire by entering the following command in privileged EXEC mode:

```
Router# show cable l2-vpn xconnect mpls-vc-map state
Load for five secs: 2%/0%; one minute: 2%; five minutes: 1%
Time source is hardware calendar, *23:05:30.061 pst Thu Mar 3 2011
MAC Address    Peer IP Address  VCID  Type  Prio  State  Customer Name/VPNID  State
54d4.6f88.7362  10.8.8.8        97    Prim*        UP                          UP
```

Example: Verifying MPLS Mapping

The following example shows how to verify the details about the mapping between an MPLS pseudowire and its virtual circuits by entering the following command in privileged EXEC mode:

```
Router# show cable l2-vpn xconnect mpls-vc-map 54d4.6f88.7362 verbose
Load for five secs: 2%/0%; one minute: 1%; five minutes: 2%
Time source is hardware calendar, *16:46:29.139 pst Thu Mar 3 2011
MAC Address                    : 54d4.6f88.7362
Customer Name                  :
Prim Sid                       : 18
```

```
Cable Interface                    : Cable5/0/1
MPLS-EXP                           : 0
PW TYPE                            : Ethernet
Backup enable delay                : 0 seconds
Backup disable delay               : 0 seconds
Primary peer
Peer IP Address (Active)           : 10.8.8.8
XConnect VCID                      : 97
Circuit  ID                        : Bu254:97
Local State                        : UP
Remote State                       : UP
Total US pkts                      : 418756
Total US bytes                     : 586227332
Total US pkts   discards           : 0
Total US bytes  discards           : 0
Total DS pkts                      : 417347
Total DS bytes                     : 591773612
Total DS pkts   discards           : 0
Total DS bytes  discards           : 0
```

Additional References

The following sections provide references related to the MQC QoS feature.

Related Documents

Related Topic	Document Title
CMTS cable commands	Cisco IOS CMTS Cable Command Reference
Modular Quality of Service Command-Line Interface	Cisco IOS Quality of Service Solutions Configuration Guide, Release 12.2
IP Differentiated Services Code Point Marking	Cisco IOS Quality of Service Solutions Configuration Guide, Release 12.2
Weighted Random Early Detection	Cisco IOS Quality of Service Solutions Configuration Guide, Release 12.2
DiffServ Tunneling Modes for MPLS Networks	MPLS Design TechNotes

Standards

Standard	Title
No new or modified standards are supported by this feature, and support for existing standards has not been modified by this feature.	—

MIBs

MIB	MIBs Link
No new or modified MIBs are supported by this feature, and support for existing MIBs has not been modified by this feature.	To locate and download MIBs for selected platforms, Cisco software releases, and feature sets, use Cisco MIB Locator found at the following URL: http://www.cisco.com/go/mibs

RFCs

RFC	Title
No new or modified RFCs are supported by this feature, and support for existing RFCs has not been modified by this feature.	—

Technical Assistance

Description	Link
The Cisco Support and Documentation website provides online resources to download documentation, software, and tools. Use these resources to install and configure the software and to troubleshoot and resolve technical issues with Cisco products and technologies. Access to most tools on the Cisco Support and Documentation website requires a Cisco.com user ID and password.	http://www.cisco.com/cisco/web/support/index.html

Feature Information for MQC QoS on the Cisco CMTS Routers

Use Cisco Feature Navigator to find information about platform support and software image support. Cisco Feature Navigator enables you to determine which software images support a specific software release, feature set, or platform. To access Cisco Feature Navigator, go to http://tools.cisco.com/ITDIT/CFN/. An account on http://www.cisco.com/ is not required.

Note The below table lists only the software release that introduced support for a given feature in a given software release train. Unless noted otherwise, subsequent releases of that software release train also support that feature.

Table 9: Feature Information for MQC QoS on the Cisco CMTS Routers

Feature Name	Releases	Feature Information
MQC QoS on the Cisco CMTS Routers	12.2(33)SCB	MQC simplifies the configuration of QoS on the Cisco CMTS routers by defining a common command syntax and resulting set of QoS behaviors across platforms.
MQC QoS on the Cisco CMTS Routers	12.2(33)SCC	The output-rate command was introduced to limit the upstream bandwidth output rate to a smaller number than that of the physical link bandwidth.
802.1Q QoS Support on GiGE WAN	12.2(33)SCF	This feature introduces QoS service on the Gigabit Ethernet WAN interface for 802.1q packets, enabling the user to set priority bits for traffic prioritization. The following commands were introduced or modified: • **class** • **class-map** • **policy-map** • **match cos** • **set cos**
MPLS short-pipe mode	12.2(33)SCF	This feature introduces QoS service on the WAN interface for MPLS EXP bits, enabling the user to set priority bits for traffic prioritization. The following commands were introduced or modified: • **match mpls experimental topmost** • **set mpls experimental topmost** • **show policy-map interface** • **show cable l2-vpn xconnect**

Feature Name	Releases	Feature Information
Input MQC Support on the Cable Interfaces	12.2(33)SCG	This feature enables you to differentiate upstream traffic on cable bundle interface and set MPLS EXP bits without changing the ToS and DSCP value of IP packets. The following sections provide information about this feature: • Input MQC Support on the Cable Bundle Interfaces , on page 6 • Configuring Input MQC Support on the Cable Bundle Interfaces, on page 22 • Example: Configuring Input MQC Support on the Cable Bundle Interfaces, on page 26