QoS—HQF Multiple Policy Support

First Published: October 10, 2008
Last Updated: October 10, 2008

The QoS—HQF Multiple Policy Support feature enables you to configure queueing service policies at the tunnel (logical) interface level and at the physical (virtual) interface level simultaneously by using the modular quality of service (QoS) command-line interface (CLI) (MQC).

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 for QoS—HQF Multiple Policy Support" section on page 17.

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

Contents

Americas Headquarters:
Cisco Systems, Inc., 170 West Tasman Drive, San Jose, CA 95134-1706 USA

Prerequisites for QoS—HQF Multiple Policy Support

Configure MQC in your network.

Restrictions for QoS—HQF Multiple Policy Support

The QoS—HQF Multiple Policy Support feature has the following restrictions:

- You must apply service policies in the output direction only; input service policies are not supported.

  Note Service policies contain the queueing and shaping mechanisms including policy maps that you configure for your router.

- Only policy maps with no specific types associated with them are supported.
- The sequence of policy-map attachment on the tunnel and physical interface is critical. You must attach the policy map on the physical interface before you attach the policy map on the tunnel. If there is a queueing policy map on the tunnel and you attempt to attach a queueing policy map on the physical interface, a warning message is generated, asking you to remove the policy map on the tunnel first, before configuring it on the physical interface.
- Because tunnel traffic is mapped to the class-default of the physical interface policy map, you should not have user-defined classes at the same level that use all the link bandwidth. Based on the estimated bandwidth taken by tunnel traffic, you should configure a policy map on the physical interface.
- If there is no queueing policy map on the tunnel interface, the queueing policy map on the physical interface is in effect; and if there is no queueing policy map on the physical interface, the queueing policy map on the interface is in effect as was the case in prior releases.
- If you configure a policy map with non-queueing features on the tunnel interface, along with a non-queueing policy map on the physical interface, the policy maps are independent of each other and function accordingly as was the case in prior releases.
- The maximum hierarchical levels allowed with a queueing policy map on a tunnel and a queueing policy map on the physical interface are four.
- Multiple policy support is exclusively for configuring a queueing policy map on a tunnel and the physical interface on which the tunnel is built. Tunnel environments that are supported are generic routing encapsulation (GRE), GRE/IPSec, dynamic virtual template interface (DVTI), static virtual template interface (SVTI), and easy virtual private network (EZVPN). Dynamic multipoint VPN (DMVPN) environments are not supported.

  Note There can be anti-replay issues if you have a GRE/IPSec tunnel in your network. This is because of the hierarchical queueing framework (HQF) design and is not addressed in this release. You should increase your anti-replay window.

Information About QoS—HQF Multiple Policy Support

To use the QoS—HQF Multiple Policy Support feature, you should understand the following concepts:

Overview of QoS—HQF Multiple Policy Support

Prior to Cisco IOS Release 12.4(22)T, HQF, which was introduced in Cisco IOS Release 12.4(20)T, supported a policy map configured either on a main interface or on any child target interface. HQF did not support the coexistence of policy maps with queueing action on both the main interface and other child targets such as tunnels or subinterfaces.

Before the introduction of HQF in Cisco IOS Release 12.4(20)T, you could attach a policy map with queueing action on both a tunnel interface and the physical interface that carries the tunnel. These two policies were treated independently.

The above configuration combination allows QoS treatment of tunneled and non-tunneled traffic. However, with HQF, the configuration combination was not supported. Since HQF supported QoS on a tunnel or on a physical interface, but not both, only one type of traffic, either tunneled or non-tunneled, received QoS treatment.

Benefits of QoS—HQF Multiple Policy Support

The QoS—HQF Multiple Policy Support feature provides the following benefits:

- Low latency propagation from the tunnel to the main interface for voice traffic. If you configure voice traffic and give it priority at a tunnel interface, the priority voice traffic receives low latency treatment at the main interface as a result of the HQF layer hierarchy merge.
- Extension of the benefits introduced in the QoS—Hierarchical Queueing Framework (HQF) feature.

How to Configure QoS—HQF Multiple Policy Support

This section contains the following procedures:

Configuring a Service Policy on a Main Interface

Perform the following task to configure a service policy and attach it to the main interface in the output direction. This action also installs HQF on the main interface.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **policy-map** *policy-map-name*
4. **class** [*class-name* | **class-default**]
5. **shape** [**average** | **peak**] *cir* [*bc*] [*be*]
6. **interface** *type number*
7. **service-policy** {**input** | **output**} *policy-map-name*
8. **end**

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 shape`	Specifies the name of the policy map to be created. Enters policy-map configuration mode. • Enter the *policy-map* name.
Step 4	**class** [*class-name* | **class-default**] **Example:** `Router(config-pmap)# class class-default`	Specifies the class so that you can configure or modify its policy. Enters policy-map class configuration mode. • Enter the *class-name* argument or the **class-default** keyword.
Step 5	**shape** [**average** | **peak**] *cir* [*bc*] [*be*] **Example:** `Router(config-pmap-c)# shape average 256000`	Shapes traffic to the indicated bit rate according to the algorithm specified. • Enter **average** or **peak** rate shaping. • Enter the committed information rate (CIR) in bits per second (bps). • (Optional) Enter the committed burst (bc) size or the excess burst (be) size in bits.

	Command or Action	Purpose
Step 6	**interface** *type number* **Example:** `Router(config-pmap-c)# interface gigabitethernet0/2`	Configures the interface type specified and enters interface configuration mode. • Enter the interface type and number.
Step 7	**service-policy** {**input** | **output**} *policy-map-name* **Example:** `Router(config-if)# service-policy output shape`	Specifies the name of the service policy to be attached to the interface. • Enter the **input** or **output** keyword followed by the policy-map name. **Note** Service policies are supported in the output direction only in Cisco IOS Release 12.4(22)T.
Step 8	**end** **Example:** `Router(config-if)# end`	(Optional) Exits interface configuration mode and returns to privileged EXEC mode.

Configuring Class Maps to Use in Service Policies

Perform the following task to configure class maps to use in your service policies.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **class-map** [**match-all** | **match-any**] *class-map-name*
4. **match** [**ip**] **precedence** {*precedence-criteria1* | *precedence-criteria2* | *precedence-criteria3* | *precedence-criteria4*}
5. **end**

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.

	Command or Action	Purpose
Step 3	**class-map** [**match-all** | **match-any**] *class-map-name* **Example:** `Router(config)# class-map voice`	Creates a class map to use for matching packets to a specified class. Enters class-map configuration mode. • Enter the class-map name.
Step 4	**match** [**ip**] **precedence** {*precedence-criteria1* | *precedence-criteria2* | *precedence-criteria3* | *precedence-criteria4*} **Example:** `Router(config-cmap)# match ip precedence 3`	Identifies IP precedence criteria to use as the match criterion. • Enter the precedence criterion.
Step 5	**end** **Example:** `Router(config-cmap)# end`	(Optional) Exits class-map configuration mode and returns to privileged EXEC mode.

Configuring a Child Service Policy

Perform the following task to configure a child service policy.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **policy-map** *policy-map-name*
4. **class** [*class-name* | **class-default**]
5. **bandwidth** {*bandwidth-kbps* | **remaining percent** *percentage* | **percent** *percentage*}
6. **exit**
7. **class** [*class-name* | **class-default**]
8. **priority** {*bandwidth-kbps* | **percent** *percentage*} [**burst**]
9. **end**

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.

	Command or Action	Purpose
Step 3	**policy-map** *policy-map-name* **Example:** `Router(config)# policy-map tunnel_queues`	Specifies the name of the policy map to be created. Enters policy-map configuration mode. • Enter the policy-map name.
Step 4	**class** [*class-name* | **class-default**] **Example:** `Router(config-pmap)# class voice`	Specifies the class so that you can configure or modify its policy. Enters policy-map class configuration mode. • Enter a class name or the **class-default** keyword.
Step 5	**bandwidth** {*bandwidth-kbps* | **remaining percent** *percentage* | **percent** *percentage*} **Example:** `Router(config-pmap-c)# bandwidth percent 50`	Specifies or modifies the bandwidth allocated for a class belonging to a policy map. • Enter the bandwidth or the percentage.
Step 6	**exit** **Example:** `Router(config-pmap-c)# exit`	Exits policy-map class configuration mode.
Step 7	**class** [*class-name* | **class-default**] **Example:** `Router(config-pmap)# class video`	Specifies the class so that you can configure or modify its policy. Enters policy-map class configuration mode. • Enter a class name or the **class-default** keyword. **Note** Skip this step if you do not need to create another class.
Step 8	**priority** {*bandwidth-kbps* | **percent** *percentage*} [*burst*] **Example:** `Router(config-pmap-c)# priority percent 49`	Gives priority to a class of traffic belonging to a policy map. • Enter the guaranteed allowed bandwidth or the percent available.
Step 9	**end** **Example:** `Router(config-pmap-c)# end`	(Optional) Exits policy-map class configuration mode and returns to privileged EXEC mode.

Configuring a Parent Service Policy Using a Child Service Policy

Perform the following task to configure a parent service policy using the child service policy that you just configured.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **policy-map** *policy-map-name*
4. **class** [*class-name* | **class-default**]
5. **shape** [**average** | **peak**] *cir* [*bc*] [*be*]

6. **service-policy** *policy-map-name*
7. **end**

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 tunnel_queues`	Specifies the name of the policy map to be created. Enters policy-map configuration mode. • Enter the policy-map name.
Step 4	**class** [*class-name* | **class-default**] **Example:** `Router(config-pmap)# class voice`	Specifies the class so that you can configure or modify its policy. Enters policy-map class configuration mode. • Enter a class name or the **class-default** keyword.
Step 5	**shape** [**average** | **peak**] *cir* [*bc*] [*be*] **Example:** `Router(config-pmap-c)# shape average 3000000`	Shapes traffic to the indicated bit rate according to the algorithm specified. • Enter **average** or **peak** rate shaping. • Enter the committed information rate (CIR) in bits per second (bps). • (Optional) Enter the committed burst (bc) size or the excess burst (be) size in bits.
Step 6	**service-policy** *policy-map-name* **Example:** `Router(config-pmap-c)# service-policy tunnel_queues`	Specifies the name of the service policy to be attached to the main interface.
Step 7	**end** **Example:** `Router(config-pmap-c)# end`	Exits policy-map class configuration mode and returns to privileged EXEC mode.

Attaching a Service Policy to a Tunnel Interface

Perform the following task to attach a service policy to a tunnel interface in the output direction.

Prerequisites

You must configure a GRE, GRE/IPSec, DVTI, SVTI, or EZVPN tunnel in your network. For detailed information, see the *Implementing Tunnels* feature module, the *Generic Routing Encapsulation (GRE) Tunnel Keepalive* feature module, and the *IPSec Virtual Tunnel Interface* feature module.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **interface tunnel** *number*
4. **service-policy** {**input** | **output**} *policy-map-name*
5. **end**

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 tunnel** *number* **Example:** `Router(config)# interface tunnel1`	Specifies a tunnel interface and enters interface configuration mode.
Step 4	**service-policy** [**type access-control**]{**input** | **output**} *policy-map-name* **Example:** `Router(config-if)# service-policy output tunnel_traffic`	Specifies the name of the service policy to be attached to the main interface. • Enter the **input** or **output** keyword followed by the policy-map name. **Note** Service policies are supported in the output direction only in Cisco IOS Release 12.4(22)T.
Step 5	**end** **Example:** `Router(config-if)# end`	(Optional) Exits interface configuration mode and returns to privileged EXEC mode.

Verifying the Multiple Policy Support Configuration

Perform the following task to verify that HQF has been installed and enabled on an interface.

SUMMARY STEPS

1. **enable**

Note Use the following **show** command in user EXEC or privileged EXEC mode after you attach a service policy on a target (tunnel or physical interface).

2. **show policy-map interface** *interface-name*

Note Use the following **show** commands in user EXEC or privileged EXEC mode after you configure your policy maps and class maps.

3. **show class-map** [*class-map-name*]
4. **show policy-map** [*policy-map*]
5. **show running-config** [**policy-map** *policy-map-name* | **class-map** *class-map-name*]
6. **exit**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	(Optional) Enables privileged EXEC mode. • Enter your password if prompted. **Note** Skip this step if you are using the **show** commands in user EXEC mode.
Step 2	**show policy-map interface** *interface-name* **Example:** `Router# show policy-map interface gigabitethernet0/2`	Displays the packet statistics of all classes that are configured for all service policies either on the specified interface or subinterface or on a specific PVC on the interface. • Enter the interface name. **Note** Use this command after you attach a service policy on a target (tunnel or physical interface).
Step 3	**show class-map** [*class-map-name*] **Example:** `Router# show class-map voice`	Displays information about any class maps currently configured. • Enter the class-map-name. **Note** Use this command after you configure your policy maps and class maps.

	Command or Action	Purpose
Step 4	**show policy-map** [*policy-map-name*] **Example:** `Router# show policy-map voice`	Displays information about any policy maps currently configured. • Enter the policy-map name. **Note** Use this command after you configure your policy maps and class maps.
Step 5	**show running-config** [**policy-map** *policy-map-name* | **class-map** *class-map-name*] **Example:** `Router# show running-config policy-map tunnel_traffic`	Displays the configuration information currently running on the router. • Enter the policy-map name or the class-map name. **Note** Use this command after you configure your policy maps and class maps.
Step 6	**exit** **Example:** `Router# exit`	(Optional) Exits privileged EXEC mode.

Configuration Examples for QoS—HQF Multiple Policy Support

This section provides configuration examples for the QoS—HQF Multiple Policy Support feature.

- Configuring QoS—HQF Multiple Policy Support: Example, page 11
- Verifying QoS—HQF Multiple Policy Support: Examples, page 12

Configuring QoS—HQF Multiple Policy Support: Example

In the following example, there is video, data, and non-critical data traffic going over a tunnel. There are also mission-critical data and internet traffic bypassing the tunnel. The requirements are that the tunnel traffic be protected, thereby giving a bandwidth guarantee to data, a latency guarantee to video, and some bandwidth guarantee to the non-tunnel mission-critical data traffic. At the same time, the tunnel traffic has to be shaped to 3 Megabits per second (Mbps). This calls for queueing on the tunnel and the physical interface. The total link bandwidth available on the physical interface is 10 Mbps.

```
Router# configure terminal
Enter configuration commands, one per line. End with CNTL/Z.
Router(config)# policy-map tunnel_queues
Router(config-pmap)# class video
Router(config-pmap-c)# priority percent 40 <---- Video traffic gets 40% of 3 Mbps, which
is 1.2 Mbps of bandwidth, and a latency guarantee when there is congestion.
Router(config-pmap-c)# exit
Router(config-pmap)# class critical-data
Router(config-pmap-c)# bandwidth percent 40 <---- Critical-data class gets 40% of 3 Mbps,
which is 1.2 Mbps, and a bandwidth guarantee when there is congestion.
Router(config-pmap-c)# exit
Router(config-pmap)# exit
Router(config)# policy-map tunnel_queues
Router(config-pmap)# class class-default
Router(config-pmap-c)# shape average 3000000
Router(config-pmap-c)# service-policy tunnel_queues
Router(config-pmap-c)# exit
Router(config-pmap)# exit
```

```
Router(config)# policy-map physical_traffic
Router(config-pmap)# class mission-critical
Router(config-pmap-c)# bandwidth 3000 <---- Mission-critical class gets 1 Mbps of the
bandwidth guarantee when there is congestion.
Router(config-pmap-c)# class class-default <---- Tunnel traffic is mapped to the
class-default class.
Router(config-pmap-c)# shape average 7000000 <---- The class-default class is shaped to
7 Mbps.
Router(config-pmap-c)# exit
Router(config-pmap)# exit
Router(config)# interface Ethernet2/1
... Interface-specific configuration goes here.
Router(config-if)# service-policy output physical_traffic
Router(config-if)# exit
Router(config)# interface tunnel1
... Tunnel-specific configuration goes here.
Router(config-if)# service-policy output tunnel_traffic
Router(config-if)# end
```

Verifying QoS—HQF Multiple Policy Support: Examples

In the following examples, multiple policies have been configured on the main interface and on the tunnel interface.

```
Router# show policy-map tunnel_traffic

  Policy Map tunnel_traffic
    Class class-default
      Average Rate Traffic Shaping
      cir 3000000 (bps)
      service-policy tunnel_queues

Router# show policy-map tunnel_queues

  Policy Map tunnel_queues
    Class critical-data
      bandwidth 40 (%)
    Class voice
      priority 40 (%)

Router# show policy-map physical_traffic

  Policy Map physical_traffic
    Class mission-critical
      bandwidth 3000 (kbps)
    Class class-default
      Average Rate Traffic Shaping
      cir 7000000 (bps)

Router# show running-config interface Ethernet1/2

Building configuration...

Current configuration : 169 bytes
!
interface Ethernet1/2
 ip address 10.0.0.2 255.255.255.0
 no ip redirects
 no ip proxy-arp
 load-interval 30
 duplex half
```

```
 service-policy output physical_traffic
end

Router# show running-config interface virtual-template 1

Building configuration...

Current configuration : 209 bytes
!
interface Virtual-Template1 type tunnel
 ip unnumbered Loopback0
 no ip redirects
 load-interval 30
 tunnel mode ipsec ipv4
 tunnel protection ipsec profile ipspro
 service-policy output tunnel_traffic
end

Router# show policy-map interface

 Ethernet1/2

  Service-policy output: physical_traffic

    Class-map: mission-critical (match-all)
      102694 packets, 39639884 bytes
      30 second offered rate 3683000 bps, drop rate 0 bps
      Match: ip precedence 3
      Queueing
      queue limit 64 packets
      (queue depth/total drops/no-buffer drops) 0/8/0
      (pkts output/bytes output) 102695/39640270
      bandwidth 3000 kbps

    Class-map: class-default (match-any)
      215680 packets, 91790836 bytes
      30 second offered rate 8518000 bps, drop rate 2755000 bps
      Match: any
      Queueing
      queue limit 64 packets
      (queue depth/total drops/no-buffer drops) 132/38970/0
      (pkts output/bytes output) 70436/29283508
      shape (average) cir 7000000, bc 28000, be 28000
      target shape rate 7000000
 Virtual-Template1

  Service-policy output: tunnel_traffic

    Service policy content is displayed for cloned interfaces only such as vaccess and
sessions
 Virtual-Access3

  Service-policy output: tunnel_traffic

    Class-map: class-default (match-any)
      96124 packets, 35758128 bytes
      30 second offered rate 4764000 bps, drop rate 2279000 bps
      Match: any
      Queueing
      queue limit 64 packets
      (queue depth/total drops/no-buffer drops) 130/38970/0
      (pkts output/bytes output) 40226/17618988
      shape (average) cir 3000000, bc 12000, be 12000
      target shape rate 3000000
```

```
Service-policy : tunnel_queues

  queue stats for all priority classes:
    Queueing
    queue limit 64 packets
    (queue depth/total drops/no-buffer drops) 0/16928/0
    (pkts output/bytes output) 16115/7058370

  Class-map: critical-data (match-any)
    33043 packets, 12291996 bytes
    30 second offered rate 1641000 bps, drop rate 996000 bps
    Match: ip precedence 1
      33043 packets, 12291996 bytes
      30 second rate 1641000 bps
    Queueing
    queue limit 64 packets
    (queue depth/total drops/no-buffer drops) 64/16991/0
    (pkts output/bytes output) 16052/7030776
    bandwidth 40% (1200 kbps)

  Class-map: voice (match-all)
    33043 packets, 12291996 bytes
    30 second offered rate 1641000 bps, drop rate 995000 bps
    Match: ip precedence 2
    Priority: 40% (1200 kbps), burst bytes 30000, b/w exceed drops: 16928

  Class-map: class-default (match-any)
    30038 packets, 11174136 bytes
    30 second offered rate 1493000 bps, drop rate 1286000 bps
    Match: any

    queue limit 64 packets
    (queue depth/total drops/no-buffer drops) 62/21979/0
    (pkts output/bytes output) 8059/3529842
```

Additional References

The following sections provide references related to the QoS—HQF Multiple Policy Support feature.

Related Documents

Related Topic	Document Title
HQF	*QoS—Hierarchical Queueing Framework (HQF)* feature module
MQC	*Applying QoS Features Using the MQC* feature module
QoS commands: complete command syntax, command modes, command history, defaults, usage guidelines, and examples	*Cisco IOS Quality of Service Solutions Command Reference*
Tunnel configuration information	• *Implementing Tunnels* feature module • *Generic Routing Encapsulation (GRE) Tunnel Keepalive* feature module • *IPSec Virtual Tunnel Interface* feature module

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 IOS 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 website provides extensive online resources, including documentation and tools for troubleshooting and resolving technical issues with Cisco products and technologies. To receive security and technical information about your products, you can subscribe to various services, such as the Product Alert Tool (accessed from Field Notices), the Cisco Technical Services Newsletter, and Really Simple Syndication (RSS) Feeds. Access to most tools on the Cisco Support website requires a Cisco.com user ID and password.	http://www.cisco.com/techsupport

Command Reference

This feature uses no new or modified commands.

Feature Information for QoS—HQF Multiple Policy Support

Table 1 lists the release history for this feature.

Not all commands may be available in your Cisco IOS software release. For release information about a specific command, see the command reference documentation.

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

Note Table 1 lists only the Cisco IOS software release that introduced support for a given feature in a given Cisco IOS software release train. Unless noted otherwise, subsequent releases of that Cisco IOS software release train also support that feature.

Table 1 Feature Information for QoS—HQF Multiple Policy Support

Feature Name	Releases	Feature Information
QoS—HQF Multiple Policy Support	12.4(22)T	The QoS—HQF Multiple Policy Support feature enables you to configure queueing service policies at the tunnel (logical) interface level and at the physical (virtual) interface level simultaneously by using the MQC. No commands were introduced or modified by this feature.

Glossary

latency—The delay on a router between the time a device receives a packet and the time that packet is forwarded out the destination port.

MQC—modular quality of service (QoS) command-line interface (CLI). A way to specify a traffic class independently of QoS policies.

policy map—Any defined rule that determines the use of resources within the network. A QoS policy map identifies the traffic class to which it applies and the instructions for one or more actions to take on that traffic.

QoS—quality of service. A measure of performance for a transmission system that reflects its transmission quality and service availability. Quality of service focuses on achieving appropriate network performance for networked applications; it is superior to best effort performance.
