



QoS—Hierarchical Queueing Framework (HQF)

First Published: March 16, 2006
Last Updated: November 4, 2009

The QoS—Hierarchical Queueing Framework (HQF) feature enables you to manage quality of service (QoS) at three different levels—the physical interface level, the logical interface level, and the class level of scheduling for applying QoS queueing and shaping mechanisms by using the modular QoS command-line interface (CLI) (MQC) to provide a granular and flexible overall QoS architecture.

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—Hierarchical Queueing Framework \(HQF\)](#)” 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

- [Prerequisites for QoS—Hierarchical Queueing Framework \(HQF\)](#), page 2
- [Restrictions for QoS—Hierarchical Queueing Framework \(HQF\)](#), page 2
- [Information About QoS—Hierarchical Queueing Framework \(HQF\)](#), page 2
- [How to Configure QoS—Hierarchical Queueing Framework \(HQF\)](#), page 8
- [Configuration Examples for QoS—Hierarchical Queueing Framework \(HQF\)](#), page 13
- [Additional References](#), page 15
- [Command Reference](#), page 16



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

© 2006, 2008 Cisco Systems, Inc. All rights reserved.

- [Feature Information for QoS—Hierarchical Queueing Framework \(HQF\)](#), page 17
- [Glossary](#), page 18

Prerequisites for QoS—Hierarchical Queueing Framework (HQF)

- Configure MQC in your network.

Restrictions for QoS—Hierarchical Queueing Framework (HQF)

Service policies with queueing features cannot coexist on child and parent interfaces, such as tunnel and physical interfaces, or subinterface and physical interfaces, simultaneously.

Information About QoS—Hierarchical Queueing Framework (HQF)

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

- [Background of QoS—Hierarchical Queueing Framework \(HQF\)](#), page 2
- [Functions of QoS—Hierarchical Queueing Framework \(HQF\)](#), page 3
- [Benefits of QoS—Hierarchical Queueing Framework \(HQF\)](#), page 4
- [New Functionality in QoS—Hierarchical Queueing Framework \(HQF\)](#), page 4
- [Behavioral Changes in QoS—Hierarchical Queueing Framework \(HQF\)](#), page 5

Background of QoS—Hierarchical Queueing Framework (HQF)

MQC provides the means for you to configure QoS using a generic CLI applicable to all types of interfaces and protocols. MQC builds configurations that depend on HQF for queueing and shaping.

For example, to support Frame Relay, extensions to the HQF mechanism were required so that fragmentation could be provided within the queueing framework. These extensions enable priority queueing (PQ) configurations to be set up to support latency-sensitive traffic.

Functions of QoS—Hierarchical Queueing Framework (HQF)

HQF provides queuing and shaping capabilities. HQF is a logical engine used to support QoS features. The HQF hierarchy is a tree structure that is built using policy maps.

When data passes through an interface using HQF, the data is classified so that it traverses the branches of the tree. Data arrives at the top of the tree and is classified on one of the leaves. Data then traverses down the hierarchy (tree) until it is transmitted out the interface at the root (trunk).

For example, the following configuration builds the hierarchy shown in [Figure 1](#):

```

policy-map class
  class c1
    bandwidth 14
  class c2
    bandwidth 18

policy-map map1
  class class-default
    shape average 64000
    service-policy class

policy-map map2
  class class-default
    shape average 96000

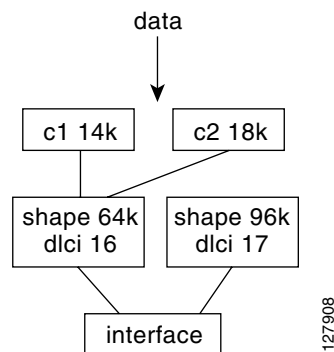
map-class frame-relay fr1
  service-policy output map1

map-class frame fr2
  service-policy output map2

interface serial4/1
  encapsulation frame-relay
  frame-relay interface-dlci 16
  class fr1
  frame-relay interface-dlci 17
  class fr2

```

Figure 1 HQF Tree Structure



Benefits of QoS—Hierarchical Queueing Framework (HQF)

The QoS—Hierarchical Queueing Framework (HQF) feature provides the following benefits:

- Faster deployment of QoS queueing and shaping in large-scale networks.
- Consistent queueing behavior applied with common MQC CLI across all main Cisco IOS software releases, making implementation of QoS easier and transparent regardless of the Cisco IOS software release being used.
- Common functionality for both distributed and non-distributed implementations, providing consistency of QoS feature behavior across all software-forwarding hardware, thus making implementation of QoS easier and transparent regardless of the platform being used.
- Behavioral consistency across hardware, resulting in accelerated delivery of feature enhancements and new QoS features in different Cisco IOS software releases.
- Multiple levels of packet scheduling.
- Support for integrated class-based shaping and queueing.
- The ability to apply fair queueing and drop policies on a per-class basis.

New Functionality in QoS—Hierarchical Queueing Framework (HQF)

The QoS—Hierarchical Queueing Framework (HQF) feature introduces the following functionality:

Hierarchical Policy with Queueing Features at Every Level

You can apply class-based queueing to any traffic class in the parent or child level of a hierarchical policy and obtain service levels for different sessions or subscribers.

In the example shown below, the traffic belonging to class parent-c2 has more scheduling time than class parent-c1:

```
policy-map child
  class child-c1
    bandwidth 400
  class child-c2
    bandwidth 400

policy-map parent
  class parent-c1 <-----
    bandwidth 1000
    service-policy child
  class parent-c2 <-----
    bandwidth 2000
    service-policy child
```

Shaping in an ATM PVC Policy

You can apply class-based shaping within an ATM PVC as shown in the following example:

```
policy-map p1
  class c1
    shape average 1000000
  class c2
    shape average 1000000

interface atm1/0.1
  pvc 1/100
    service-policy output p1
```

```
policy-map p1
  class c1
    shape average 1000000
  class c2
    shape average 1000000

interface atm1/0.1
  pvc 1/100
    service-policy output p1
```

Child Policy in a Priority Class

You can apply a child policy to a class with priority enabled as shown in the following example. The child policy can contain police or set features, but not queueing features.

```
policy-map p1
  class c1
    priority 256
    service-policy child
```

Behavioral Changes in QoS—Hierarchical Queueing Framework (HQF)

The QoS—Hierarchical Queueing Framework (HQF) feature introduces the following behavioral changes in some QoS features:

Flow-Based Fair-Queueing Support in Class-Default

The fair-queueing behavior for the class-default class is flow-based. This is a change from the weighted fair queueing (WFQ) behavior in previous releases. With flow-based fair queueing, the flow queues in the class-default class are scheduled equally instead of by weight based on the IP Precedence bits.

Default Queueing Implementation for Class-Default

When you do not explicitly configure the class-default class in a policy map, its default queueing behavior is FIFO. You can configure the **bandwidth**, **fair-queue**, or **service-policy** commands in the class-default class to achieve different queueing behaviors.

Class-Default and Bandwidth

The bandwidth assigned to the class-default class is the unused interface bandwidth not consumed by user-defined classes. By default, the class-default class receives a minimum of 1% of the interface bandwidth.

Default Queueing Implementation for Shape Class

When you configure the **shape** command in a class, the default queueing behavior for the shape queue is FIFO instead of weighted fair queueing (WFQ). You can configure the **bandwidth**, **fair-queue**, or **service-policy** commands in shape class to achieve different queueing behaviors.

Policy Map and Interface Bandwidth

In HQF, a policy map can reserve up to 100 percent of the interface bandwidth. If you do not assign an explicit bandwidth guarantee to the class-default class, you can assign a maximum of 99 percent of the interface bandwidth to user-defined classes and reserve the other 1 percent for the class-default class.



Note If you are migrating to Cisco IOS Release 12.4(20)T and the configured policy map allocates 100 percent of the bandwidth to the user-defined classes, an error message appears in the console after booting the HQF image. The message indicates that the allocated bandwidth exceeds the allowable amount, and the service policy is rejected. In HQF, you must reconfigure the policy to account for the minimum 1 percent bandwidth guaranteed for the class-default. Then you can apply a service policy to the interface.

Per-Flow Queue Limit in Fair Queue

In HQF, when you enable fair queuing, the default per-flow queue limit is 1/4 of the class queue limit. If you do not enable the queue limit in a class, the default per-flow queue limit is 16 packets (1/4 of 64).

Over-Subscription Support for Multiple Policies on Logical Interfaces

When you attach a shaping policy to multiple logical interfaces including a subinterface, and the sum of the shape rate exceeds the physical interface bandwidth, congestion at the physical interface results in back pressure to each logical interface policy. This back pressure causes each policy to reduce the output rate down to its fair share of the interface bandwidth.

Here is an example: 10 subinterface policies each shaped to 2 Mbps, physical interface has 10 Mbps bandwidth (2:1 oversubscription), when all 10 subinterfaces are sending at 2 Mbps, each subinterface gets a throughput of 1 Mbps (10 Mbps/10 subinterfaces).

Shaping on a GRE Tunnel

In HQF, you can apply the shaping to a generic routing encapsulation (GRE) tunnel by using a hierarchical service policy after encapsulation. This means that the shape rate is based on packets with tunnel encapsulation and L2 encapsulation.

When configuring the shape feature in the parent policy applied to the tunnel interface, you can use the class-default class only. You cannot configure a user-defined class in the parent policy.

A typical hierarchical policy applied to a GRE tunnel interface is shown below:

```
Interface tunnel0
Service-policy output parent

policy-map parent
  class class-default
    shape average 10000000
    service-policy child

policy-map child
  class voice
    priority 512
  class video
    bandwidth 6000
  class data
    bandwidth 3000
```

**Note**

Some QoS deployments include a service policy with queuing features applied at the tunnel or a virtual interface and a service policy with queuing features applied at the physical interface. In Cisco IOS Release 12.4(20)T, you can apply a service policy with queuing features only at one of these interfaces. When migrating to Cisco IOS Release 12.4(20)T, a router configuration containing service policies at both interfaces will only keep the one applied to the physical interface.

FRF.12 and FRF.9

With HQF implementation, when you enable Frame Relay fragmentation (FRF.12) on an FR PVC or FR main interface, priority class packets are no longer subject to fragmentation. Priority packets, regardless of the packet size, always interleave among data fragments.

When you enable Frame Relay payload compression (FRF.9) on an FR PVC or main interface, priority class packets are no longer compressed. When you enable both FRF.12 and FRF.9, priority class packets are neither fragmented nor compressed.

User-Defined Classes Added to Policy Maps Attached to Logical Interfaces

A policy map may be configured with multiple user-defined classes and may contain a default class, called class-default. Optionally, a policy map may contain just the class-default, as illustrated below:

```
policy-map parent
  class class-default
  service-policy child
```

Typically, at this point, you would attach the policy map to the interface. After the policy map has been attached to the interface, the HQF would allow you to add a user-defined class to the policy map.

However, HQF behavior has now changed so that this kind of modification is no longer permitted on a logical interface. If you want to add a user-defined class to a policy map (and that policy map has already been attached to a logical interface), you must first remove the policy map from the logical interface. Then add the user-defined class to the policy map and reattach the policy map to the logical interface.

**Note**

This behavior change applies only to logical interfaces. It does not apply to physical interfaces.

Nested Policy and Reference Bandwidth for Child-Policy

In HQF when you configure a nested policy with a child queuing policy under a parent shaping class, the reference bandwidth for the child queuing policy is taken from the following: minimum (parent shaper rate, parent class's implicit / explicit bandwidth guarantee). When you do not define bandwidth for the parent class, the interface bandwidth divides equally among all parent classes as the implicit bandwidth guarantee.

The example below shows a nested policy applied on a serial interface of 1536 Kbps. The 1536 Kbps is equally shared, as the implicit bandwidth, among parent classes parent-c1 and class-default. For the parent class, the shaping rate of 1200 Kbps is the maximum, while the implicit guarantee of 768 Kbps is the minimum.

```
interface serial 0/0
  service-policy parent
  policy-map child
    class child-c1
      bandwidth percent 10
  policy-map parent
    class parent-c1
      shape average 1200000
```

```
service-policy child
```

For the child policy child-c1 to take the parent shaping rate as the reference bandwidth, configure parent class parent-c1 with an explicit guarantee greater than the shaping rate. For example,

```
policy-map parent
class parent-c1
    bandwidth 1300
    shape average 1200000
service-policy child
```

When configuring explicit bandwidth for parent classes with oversubscription, the restriction here [Policy Map and Interface Bandwidth, page 6](#) applies.

How to Configure QoS—Hierarchical Queueing Framework (HQF)

This section contains the following procedures:

- [Configuring a Service Policy, page 8](#) (required)
- [Attaching an MQC Policy to a Map Class, page 10](#) (required)
- [Verifying the HQF Configuration, page 12](#) (optional)

Configuring a Service Policy

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

SUMMARY STEPS

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

DETAILED STEPS

	Command or Action	Purpose
Step 1	enable Example: Router> enable	Enables privileged EXEC mode. <ul style="list-style-type: none"> Enter your password if prompted.
Step 2	configure terminal Example: Router# configure terminal	Enters global configuration mode.
Step 3	policy-map [type access-control] <i>policy-map-name</i> Example: Router(config)# policy-map shape	Specifies the name of the policy map to be created. Enters policy-map configuration mode. <ul style="list-style-type: none"> The optional type access-control keywords determine the exact pattern to look for in the protocol stack of interest. Enter the policy-map name.
Step 4	class [<i>class-name</i> 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. <ul style="list-style-type: none"> Enter the <i>class-name</i> argument or the class-default keyword.
Step 5	shape [average peak] <i>cir</i> [<i>bc</i>] [<i>be</i>] Example: Router(config-pmap-c)# shape average 256000	Shapes traffic to the indicated bit rate according to the algorithm specified. <ul style="list-style-type: none"> 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	interface <i>type number</i> Example: Router(config-pmap-c)# interface serial4/3	Configures the interface type specified and enters interface configuration mode. <ul style="list-style-type: none"> Enter the interface type and number.
Step 7	encapsulation frame-relay [cisco ietf] Example: Router(config-if)# encapsulation frame-relay	Enables Frame Relay encapsulation on an interface. <ul style="list-style-type: none"> (Optional) Enter cisco or ietf to specify the encapsulation method.

	Command or Action	Purpose
Step 8	<p>service-policy [type access-control] {input output} <i>policy-map-name</i></p> <p>Example: Router(config-if)# service-policy output shape</p>	<p>Specifies the name of the policy map to be attached to the interface.</p> <p>Note You can configure policy maps on ingress or egress routers and attach them in the input or output direction of an interface. The direction (input or output) and the router (ingress or egress) to which the policy map should be attached vary according to your network configuration.</p> <ul style="list-style-type: none"> • The optional type access-control keywords determine the exact pattern to look for in the protocol stack of interest. • Enter the input or output keyword followed by the policy-map name.
Step 9	<p>end</p> <p>Example: Router(config-if)# end</p>	<p>(Optional) Exits interface configuration mode.</p>

Attaching an MQC Policy to a Map Class

Perform the following task to attach an MQC policy to a map class. This action also enables HQF.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **map-class frame-relay** *map-class-name*
4. **service-policy** [**type access-control**] {**input** | **output**} *policy-map-name*
5. **interface** *type number*
6. **frame-relay class** *name*
7. **frame-relay interface-dlci** *dlci* [**cisco** | **ietf**] [**voice-cir** *cir*] [**ppp** *virtual-template-name*]
8. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	<p>enable</p> <p>Example: Router> enable</p>	<p>Enables privileged EXEC mode.</p> <ul style="list-style-type: none"> Enter your password if prompted.
Step 2	<p>configure terminal</p> <p>Example: Router# configure terminal</p>	<p>Enters global configuration mode.</p>
Step 3	<p>map-class frame-relay <i>map-class-name</i></p> <p>Example: Router(config)# map-class frame-relay shape</p>	<p>Specifies the name of a Frame Relay map class that is to be created or modified and enters map-class configuration mode.</p> <ul style="list-style-type: none"> Enter the map-class name.
Step 4	<p>service-policy [type access-control] {input output} <i>policy-map-name</i></p> <p>Example: Router(config-map-class)# service-policy output shape</p>	<p>Specifies the name of the policy map to be attached to the interface.</p> <p>Note You can configure policy maps on ingress or egress routers and attach them in the input or output direction of an interface. The direction (input or output) and the router (ingress or egress) to which the policy map should be attached varies according to your network configuration.</p> <ul style="list-style-type: none"> The optional type access-control keywords determine the exact pattern to look for in the protocol stack of interest. Enter the input or output keyword followed by the policy-map name.
Step 5	<p>interface <i>type number</i></p> <p>Example: Router(config-map-class)# interface serial4/3</p>	<p>Configures the interface type specified and enters interface configuration mode.</p> <ul style="list-style-type: none"> Enter the interface type and number.
Step 6	<p>frame-relay class <i>name</i></p> <p>Example: Router(config-if)# frame-relay class shape</p>	<p>Associates a map class with an interface or subinterface.</p> <ul style="list-style-type: none"> Enter the name of the map class.

	Command or Action	Purpose
Step 7	<pre>frame-relay interface-dlci dlci [cisco ietf] [voice-cir cir] [ppp virtual-template-name]</pre> <p>Example: Router(config-if)# frame-relay interface-dlci 16</p>	<p>Assigns a data-link connection identifier (DLCI) to a specified Frame Relay subinterface on a router and enters Frame Relay DLCI interface configuration mode.</p> <ul style="list-style-type: none"> • Enter the DLCI number. • (Optional) Enter cisco or ietf for the encapsulation type. • (Optional; supported on the Cisco MC3810 only) Enter voice-cir and <i>cir</i> to specify the upper limit on the voice bandwidth that may be reserved for this DLCI. The default is the committed information rate (CIR) configured for the Frame Relay map class. • (Optional) Enter ppp to enable the circuit to use PPP in Frame Relay encapsulation. • (Optional) Enter the virtual template name to specify to which virtual template interface to apply the PPP connection.
Step 8	<pre>end</pre> <p>Example: Router(config-fr-dlci)# end</p>	<p>(Optional) Exits Frame Relay DLCI interface configuration mode.</p>

Verifying the HQF Configuration

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



Note

You can use the following **show** command in user EXEC or privileged EXEC mode.

SUMMARY STEPS

1. **enable**
2. **show policy-map interface** [**type access-control**] *interface-name* [**vc** [*vpi*]/] *vci*] [**dlci** *dlci*] [**input** | **output**]
3. **exit**

DETAILED STEPS

	Command or Action	Purpose
Step 1	enable Example: Router> enable	(Optional) Enables privileged EXEC mode. <ul style="list-style-type: none"> Enter your password if prompted. Note Skip this step if you are using the show command in user EXEC mode.
Step 2	show policy-map interface [type access-control] <i>interface-name</i> [vc [<i>vpi/</i>] <i>vci</i>] [dlci <i>dlci</i>] [input output]	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. <ul style="list-style-type: none"> Enter the interface name.
Step 3	exit Example: Router# exit	(Optional) Exits privileged EXEC mode.

Configuration Examples for QoS—Hierarchical Queueing Framework (HQF)

This section provides configuration examples for the QoS—Hierarchical Queueing Framework (HQF) feature.

- [Configuring QoS—Hierarchical Queueing Framework \(HQF\): Example, page 13](#)
- [Verifying the HQF Configuration: Example, page 14](#)

Configuring QoS—Hierarchical Queueing Framework (HQF): Example

There are two main tasks for configuring this feature:

- Configuring a policy map
- Attaching the policy map to a map class

In the following example, a policy map called **shape** is configured on serial interface 4/3 and attached in the output direction. Its parameters include a class **class-default**, a traffic shaping average of 256000 bps, and Frame Relay encapsulation.

```
Router# configure terminal
Enter configuration commands, one per line. End with CNTL/Z.
Router(config)# policy-map shape
Router(config-pmap)# class class-default
Router(config-pmap-c)# shape average 256000
Router(config-pmap-c)# interface serial4/3
Router(config-if)# encapsulation frame-relay
Router(config-if)# service-policy output shape
Router(config-if)# end
```

In the following example, the policy map called shape is attached to serial interface 4/3 in the output direction and is associated with a map class called shape. There is also a PVC being associated with DLCI 16.

```
Router# configure terminal
Enter configuration commands, one per line. End with CNTL/Z.
Router(config)# map-class frame-relay shape
Router(config-map-class)# service-policy output shape
Router(config-map-class)# interface serial4/3
Router(config-if)# frame-relay class shape
Router(config-if)# frame interface-dlci 16
Router(config-fr-dlci)# end
```

Verifying the HQF Configuration: Example

In the following example, shaping is active with HQF installed on serial interface 4/3. All traffic is classified to the class-default queue.

```
Router# show policy-map interface serial4/3

Serial4/3

Service-policy output: shape

Class-map: class-default (match-any)
  2203 packets, 404709 bytes
  30 second offered rate 74000 bps, drop rate 14000 bps
  Match: any
  Queueing
    queue limit 64 packets
    (queue depth/total drops/no-buffer drops) 64/354/0
    (pkts output/bytes output) 1836/337280
    shape (average) cir 128000, bc 1000, be 1000
    target shape rate 128000
      lower bound cir 0, adapt to fecn 0

Service-policy : LLQ

  queue stats for all priority classes:

    queue limit 64 packets
    (queue depth/total drops/no-buffer drops) 0/0/0
    (pkts output/bytes output) 0/0

Class-map: c1 (match-all)
  0 packets, 0 bytes
  30 second offered rate 0 bps, drop rate 0 bps
  Match: ip precedence 1
  Priority: 32 kbps, burst bytes 1500, b/w exceed drops: 0

Class-map: class-default (match-any)
  2190 packets, 404540 bytes
  30 second offered rate 74000 bps, drop rate 14000 bps
  Match: any

    queue limit 64 packets
    (queue depth/total drops/no-buffer drops) 63/417/0
    (pkts output/bytes output) 2094/386300
```

Additional References

The following sections provide references related to the QoS—Hierarchical Queuing Framework (HQF) feature.

Related Documents

Related Topic	Document Title
Frame Relay commands: complete command syntax, command modes, command history, defaults, usage guidelines, and examples	Cisco IOS Wide-Area Networking Command Reference
QoS commands: complete command syntax, command modes, command history, defaults, usage guidelines, and examples	Cisco IOS Quality of Service Solutions Command Reference
MQC	“Applying QoS Features Using the MQC” 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
<p>The Cisco Support website provides extensive online resources, including documentation and tools for troubleshooting and resolving technical issues with Cisco products and technologies.</p> <p>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.</p> <p>Access to most tools on the Cisco Support website requires a Cisco.com user ID and password.</p>	<p>http://www.cisco.com/techsupport</p>

Command Reference

The following commands are introduced or modified in the feature or features documented in this module. For information about these commands, see the *Cisco IOS Quality of Service Solutions Command Reference* at http://www.cisco.com/en/US/docs/ios/qos/command/reference/qos_book.html. For information about all Cisco IOS commands, use the Command Lookup Tool at <http://tools.cisco.com/Support/CLILookup> or the *Cisco IOS Master Command List, All Releases*, at http://www.cisco.com/en/US/docs/ios/mcl/allreleasemcl/all_book.html.

- **bandwidth (policy-map class)**
- **fair-queue (WFQ)**
- **max-reserved-bandwidth**
- **police (two rates)**
- **queue-limit**
- **random-detect**
- **random-detect atm-clp-based**
- **random-detect cos-based**
- **random-detect prec-based**
- **random-detect precedence**
- **shape-max buffers**
- **show policy-map**
- **show policy-map interface**
- **show queue**
- **show queueing**

Feature Information for QoS—Hierarchical Queueing Framework (HQF)

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, Catalyst OS, and Cisco IOS XE 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—Hierarchical Queueing Framework (HQF)

Feature Name	Releases	Feature Information
QoS—Hierarchical Queueing Framework (HQF)	12.2(28)SB 12.4(20)T	<p>The QoS—Hierarchical Queueing Framework (HQF) feature enables you to manage quality of service (QoS) at three different levels—the physical interface level, the logical interface level, and the class level of scheduling for applying QoS queueing and shaping mechanisms by using the modular QoS command-line interface (CLI) (MQC) to provide a granular and flexible overall QoS architecture.</p> <p>In Release 12.2(28)SB, this feature was introduced as QoS: Frame Relay QoS Hierarchical Queueing Framework Support on the Cisco 7200 Series Router.</p> <p>The following commands were introduced or modified: bandwidth (policy-map class), fair-queue (WFQ), max-reserved-bandwidth, police (two rates), queue-limit, random-detect, random-detect atm-clp-based, random-detect cos-based, random-detect prec-based, random-detect precedence, shape-max buffers, show policy-map, show policy-map interface, show queue, show queueing.</p>

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.

CCDE, CCENT, CCSI, Cisco Eos, Cisco HealthPresence, Cisco IronPort, the Cisco logo, Cisco Nurse Connect, Cisco Pulse, Cisco SensorBase, Cisco StackPower, Cisco StadiumVision, Cisco TelePresence, Cisco Unified Computing System, Cisco WebEx, DCE, Flip Channels, Flip for Good, Flip Mino, Flipshare (Design), Flip Ultra, Flip Video, Flip Video (Design), Instant Broadband, and Welcome to the Human Network are trademarks; Changing the Way We Work, Live, Play, and Learn, Cisco Capital, Cisco Capital (Design), Cisco:Financed (Stylized), Cisco Store, Flip Gift Card, and One Million Acts of Green are service marks; and Access Registrar, Aironet, AllTouch, AsyncOS, Bringing the Meeting To You, Catalyst, CCDA, CCDP, CCIE, CCIP, CCNA, CCNP, CCSP, CCVP, Cisco, the Cisco Certified Internetwork Expert logo, Cisco IOS, Cisco Lumin, Cisco Nexus, Cisco Press, Cisco Systems, Cisco Systems Capital, the Cisco Systems logo, Cisco Unity, Collaboration Without Limitation, Continuum, EtherFast, EtherSwitch, Event Center, Explorer, Follow Me Browsing, GainMaker, iLYNX, IOS, iPhone, IronPort, the IronPort logo, Laser Link, LightStream, Linksys, MeetingPlace, MeetingPlace Chime Sound, MGX, Networkers, Networking Academy, PCNow, PIX, PowerKEY, PowerPanels, PowerTV, PowerTV (Design), PowerVu, Prisma, ProConnect, ROSA, SenderBase, SMARTnet, Spectrum Expert, StackWise, WebEx, and the WebEx logo are registered trademarks of Cisco Systems, Inc. and/or its affiliates in the United States and certain other countries.

All other trademarks mentioned in this document or website are the property of their respective owners. The use of the word partner does not imply a partnership relationship between Cisco and any other company. (0910R)

Any Internet Protocol (IP) addresses used in this document are not intended to be actual addresses. Any examples, command display output, and figures included in the document are shown for illustrative purposes only. Any use of actual IP addresses in illustrative content is unintentional and coincidental.

© 2006, 2008 Cisco Systems, Inc. All rights reserved.