QoS: Hierarchical Queueing Framework Configuration Guide, Cisco IOS Release 15M&T

First Published: January 28, 2013

Last Modified: January 28, 2013

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

THE SPECIFICATIONS AND INFORMATION REGARDING THE PRODUCTS IN THIS MANUAL ARE SUBJECT TO CHANGE WITHOUT NOTICE. ALL STATEMENTS, INFORMATION, AND RECOMMENDATIONS IN THIS MANUAL ARE BELIEVED TO BE ACCURATE BUT ARE PRESENTED WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED. USERS MUST TAKE FULL RESPONSIBILITY FOR THEIR APPLICATION OF ANY PRODUCTS.

THE SOFTWARE LICENSE AND LIMITED WARRANTY FOR THE ACCOMPANYING PRODUCT ARE SET FORTH IN THE INFORMATION PACKET THAT SHIPPED WITH THE PRODUCT AND ARE INCORPORATED HEREIN BY THIS REFERENCE. IF YOU ARE UNABLE TO LOCATE THE SOFTWARE LICENSE OR LIMITED WARRANTY, CONTACT YOUR CISCO REPRESENTATIVE FOR A COPY.

The Cisco implementation of TCP header compression is an adaptation of a program developed by the University of California, Berkeley (UCB) as part of UCB's public domain version of the UNIX operating system. All rights reserved. Copyright © 1981, Regents of the University of California.

NOTWITHSTANDING ANY OTHER WARRANTY HEREIN, ALL DOCUMENT FILES AND SOFTWARE OF THESE SUPPLIERS ARE PROVIDED “AS IS" WITH ALL FAULTS. CISCO AND THE ABOVE-NAMED SUPPLIERS DISCLAIM ALL WARRANTIES, EXPRESSED OR IMPLIED, INCLUDING, WITHOUT LIMITATION, THOSE OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT OR ARISING FROM A COURSE OF DEALING, USAGE, OR TRADE PRACTICE.

IN NO EVENT SHALL CISCO OR ITS SUPPLIERS BE LIABLE FOR ANY INDIRECT, SPECIAL, CONSEQUENTIAL, OR INCIDENTAL DAMAGES, INCLUDING, WITHOUT LIMITATION, LOST PROFITS OR LOSS OR DAMAGE TO DATA ARISING OUT OF THE USE OR INABILITY TO USE THIS MANUAL, EVEN IF CISCO OR ITS SUPPLIERS HAVE BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

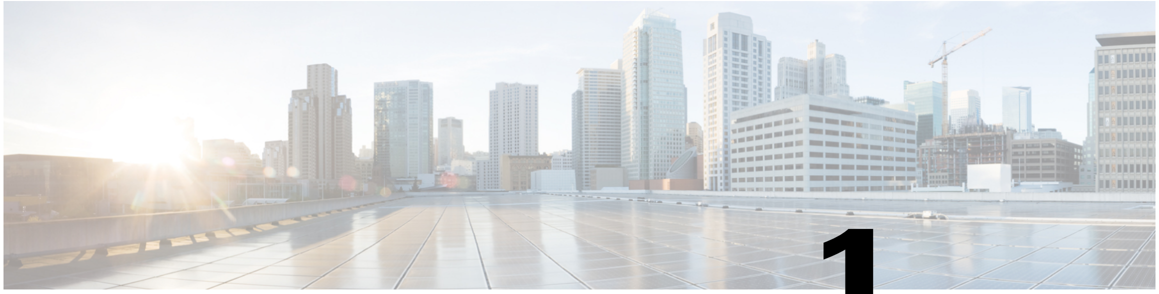
Any Internet Protocol (IP) addresses and phone numbers used in this document are not intended to be actual addresses and phone numbers. Any examples, command display output, network topology diagrams, and other figures included in the document are shown for illustrative purposes only. Any use of actual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

Cisco and the Cisco logo are trademarks or registered trademarks of Cisco and/or its affiliates in the U.S. and other countries. To view a list of Cisco trademarks, go to this URL: http://www.cisco.com/go/trademarks. Third-party trademarks mentioned 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. (1110R)

CONTENTS

CHAPTER 1

CHAPTER 2



CHAPTER 1

QoS Hierarchical Queueing Framework

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 for QoS queueing and shaping mechanisms by using the modular QoS command-line interface (MQC) to provide a granular and flexible overall QoS architecture. 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.

Finding Feature Information

Your software release may not support all the features documented in this module. For the latest caveats and feature information, see Bug Search Tool and 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.

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

Prerequisites for QoS Hierarchical Queueing Framework

Configure MQC in your network.

Restrictions for QoS Hierarchical Queueing Framework

- Service policies with queueing features cannot simultaneously coexist on child and parent interfaces, such as tunnel and physical interfaces or subinterface and physical interfaces.
- If a queueing policy is applied on a tunnel interface, and if a queueing policy needs to be applied on the physical interface on which the tunnel is built, the pmap on tunnel needs to be removed before the pmap on the physical interface can be attached.
- Queuing based QoS policies are not supported on LAG (port-channel) interfaces.

Information About QoS Hierarchical Queueing Framework

Background of QoS Hierarchical Queueing Framework

MQC allows you to configure QoS using a generic CLI that is 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 provides queueing 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.

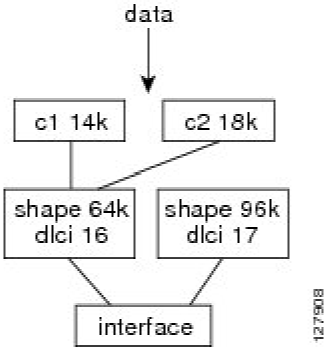
When data passes through an interface using HQF, the data is classified so that it traverses the branches of the tree. Data arrives at the top of the tree and is classified on one of the leaves. Data then traverses down the hierarchy (tree) until it is transmitted out the interface at the root (trunk).

For example, the following configuration builds the hierarchy shown in the figure below:

```
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

The QoS Hierarchical Queueing Framework 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 software releases, making implementation of QoS easier and transparent regardless of the Cisco 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 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.
- Up to three levels of queueing can be configured on the egress. Queueing features are not supported in the ingress direction.
- Priority can be configured on a nonleaf node, but nonqueueing (policing or marking) classes can be configured only on its child.

New Functionality in QoS Hierarchical Queueing Framework

The QoS Hierarchical Queueing Framework 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

The QoS Hierarchical Queueing Framework 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-polic**y 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 you can reserve the other 1 percent for the class-default class by using the **percent** keyword from the **bandwidth** (policy-map class) command. If you use the *kbps* argument, you can assign a maximum of the entire interface bandwidth minus 1 kilobits per second (kbps) to user-defined classes and reserve the remaining 1 kbps 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 on 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 of bandwidth that is guaranteed for the class-default. Then you can apply a service policy to the interface.

Per-Flow Queue Limit in Fair Queueing

In HQF, when you enable fair queueing, the per-flow queue limit is calculated in one of the following ways:

- 1/4 * n (where n = queue limit)
- Two packets (in the case of packet-based queue limits)
- One MTU size (in the case of byte-based queue limits)

These values are static even if the number of flows increase, so consider the overall buffer pool when configuring the queue limit in order to avoid exhausting the buffer pool.

Note The queue limit per class in packets and bytes can be configured without fair queue. Therefore, the minimum value of queue limit per class and queue limit per flow is not connected.

It is recommended to use the default value or 200 ms worth of packets/bytes for the “queue limit per class”.

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 queueing features applied at the tunnel or a virtual interface and a service policy with queueing features applied at the physical interface. In Cisco IOS Release 12.4(20)T, you can apply a service policy with queueing 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 keep only 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 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 queueing policy under a parent shaping class, the reference bandwidth for the child queueing 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 restrictions in the "Policy Map Bandwidth" section applies.

Handling Traffic Congestion on an Interface Configured with a Policy Map

In Cisco IOS Release 12.4(20)T, if an interface configured with a policy map is congested, the implicitly defined queue allows the traffic as defined in the bandwidth statement of each traffic class. The queueing is activated whenever there is traffic congestion on an interface.

How to Configure QoS Hierarchical Queueing Framework

Configuring a Service Policy

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:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**policy-map** [**type access-control**] *policy-map-name* **Example:** `Device(config)# policy-map shape`	Specifies the name of the policy map to be created. Enters policy-map configuration mode. • The optional **type access-control** keywords determine the exact pattern to look for in the protocol stack of interest. • Enter the policy-map name.

	Command or Action	Purpose

Step 4	**class** [*class-name* | **class-default**]
	Example:
	`Device(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:** `Device(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.
Step 6	**interface** *type number* **Example:** `Device(config-pmap-c)# interface serial4/3`	Configures the interface type specified and enters interface configuration mode. • Enter the interface type and number.
Step 7	**encapsulation frame-relay** [**cisco** | **ietf**] **Example:** `Device(config-if)# encapsulation frame-relay`	Enables Frame Relay encapsulation on an interface. • (Optional) Enter **cisco** or **ietf** to specify the encapsulation method.
Step 8	**service-policy** [**type access-control**] {**input** | **output**} *policy-map-name* **Example:** `Device(config-if)# service-policy output shape`	Specifies the name of the policy map to be attached to the interface. **Note** You can configure policy maps on ingress or egress devices and attach them in the input or output direction of an interface. The direction (input or output) and the device (ingress or egress) to which the policy map should be attached vary according to your network configuration. • 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	**end**
	Example:
	`Device(config-if)# end`
		(Optional) Exits interface configuration mode.

Attaching an MQC Policy to a Map Class

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	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**map-class frame-relay** *map-class-name* **Example:** `Device(config)# map-class frame-relay shape`	Specifies the name of a Frame Relay map class that is to be created or modified and enters map-class configuration mode. • Enter the map-class name.
Step 4	**service-policy** [**type access-control**] {**input** | **output**} *policy-map-name* **Example:** `Device(config-map-class)# service-policy output shape`	Specifies the name of the policy map to be attached to the interface. **Note** You can configure policy maps on ingress or egress devices and attach them in the input or output direction of an interface. The direction (input or output) and the device (ingress or egress) to which the policy map should be attached varies according to your network configuration. • 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.

	Command or Action	Purpose
Step 5	**interface** *type number* **Example:** `Device(config-map-class)# interface serial4/3`	Configures the interface type specified and enters interface configuration mode. • Enter the interface type and number.
Step 6	**frame-relay class** *name* **Example:** `Device(config-if)# frame-relay class shape`	Associates a map class with an interface or subinterface. • Enter the name of the map class.
Step 7	**frame-relay interface-dlci** *dlci* [**cisco** | **ietf**] [**voice-cir** *cir*] [**ppp** *virtual-template-name*] **Example:** `Device(config-if)# frame-relay interface-dlci 16`	Assigns a data-link connection identifier (DLCI) to a specified Frame Relay subinterface on a device and enters Frame Relay DLCI interface configuration mode. • Enter the DLCI number. • (Optional) Enter **cisco** or **ietf** for the encapsulation type. • (Optional; supported on the Cisco MC3810 only) Enter **voice-cir** and *cir* 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	**end**
	Example:
	`Device(config-fr-dlci)# end`
		(Optional) Exits Frame Relay DLCI interface configuration mode.

Verifying the HQF Configuration

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:** `Device> enable`	(Optional) Enables privileged EXEC mode. • 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**] *interface-name* [**vc**[*vpi*/] *vci*] [**dlci** *dlci*] [**input** | **output**] **Example:** `Device# show policy-map interface serial4/3`	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.
Step 3	**exit** **Example:** `Device# exit`	(Optional) Exits privileged EXEC mode.

Configuration Examples for QoS Hierarchical Queueing Framework

Example: Configuring QoS Hierarchical Queueing Framework

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.

```
Device# configure terminal

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

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

Example: Verifying the HQF Configuration

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

```
Device# 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 for QoS Hierarchical Queueing Framework

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

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 QoS Hierarchical Queueing Framework

The following table provides release information about the feature or features described in this module. This 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.

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

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	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 for QoS queueing and shaping mechanisms by using the modular QoS command-line interface (MQC) to provide a granular and flexible overall QoS architecture. 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. 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**.

CHAPTER 2

MQC Hierarchical Queuing with 3 Level Scheduler

The MQC Hierarchical Queuing with 3 Level Scheduler feature provides a flexible packet scheduling and queuing system in which you can specify how excess bandwidth is to be allocated among the subscriber (logical) queues.

History for the MQC Hierarchical Queuing with 3 Level Scheduler Feature

Release	Modification
12.2(31)SB2	This feature was introduced and implemented on the Cisco 10000 series router for the PRE3.

Finding Support Information for Platforms and Cisco IOS Software Images

Use Cisco Feature Navigator to find information about platform support and Cisco IOS software image support. Access Cisco Feature Navigator at http://www.cisco.com/go/fn . You must have an account on Cisco.com. If you do not have an account or have forgotten your username or password, click **Cancel** at the login dialog box and follow the instructions that appear.

Prerequisites for the Three-Level Scheduler

Traffic classes must be configured on the router using the class-map command.

Restrictions for the Three-Level Scheduler

- The priority queue in a child policy must be policed to 90 percent of the parent's shaped bandwidth.
- The three-level scheduler does not support bandwidth propagation. Therefore, you cannot configure a bandwidth guarantee for any queue other than a priority queue.
- To allow oversubscription provisioning, the admission control check is not performed.
- The three-level scheduler does not allocate an implicit bandwidth guarantee for the parent class-default class. Instead, the scheduler uses the ratio of the classes to allocate bandwidth.
- When hierarchical policies are enabled on multiple VLANs and each VLAN hierarchical policy has priority services configured in a child policy, the three-level scheduler first services the priority traffic from all VLANs and then proportionally shares the remaining bandwidth of the interface among all of the VLANs.

Note The two-level scheduler allocates an equal share of interface bandwidth to each VLAN. After the two-level scheduler serves priority services, best-effort traffic from a VLAN uses the remaining bandwidth. If priority traffic is not configured, instead of proportionally allocating the remaining bandwidth available to each VLAN, the two-level scheduler allocates the entire interface bandwidth to the VLAN's best-effort traffic.

- The sum of all priority traffic running on a given port must be less than or equal to 90 percent of the port bandwidth.

Information About the MQC Hierarchical Queuing with 3 Level Scheduler

The MQC Hierarchical Queuing with 3 Level Scheduler feature provides a flexible packet scheduling and queuing system in which you can specify how excess bandwidth is to be allocated among the subscriber queues and logical interfaces. Rather than allocating an implicit minimum bandwidth guarantee to each queue, the three-level scheduler uses the bandwidth-remaining ratio parameter to allocate unused bandwidth to each logical queue. The three-level scheduler services queues based on the following user-configurable parameters:

- Maximum rate—The specified shape rate of the parent queue.
- Bandwidth-remaining ratio—The value used to determine the portion of unused, non-guaranteed bandwidth allocated to a logical queue relative to other queues competing for the unused bandwidth.

Note At the class level, the router converts the values specified in the bandwidth bps and bandwidth remaining percent commands to a bandwidth-remaining ratio value. The router does not allow you to configure the bandwidth bps and bandwidth remaining percent commands on the physical and logical layers.

The three-level scheduler on the PRE3 supports priority propagation by propagating the priority guarantees you configure for subscriber services down to the logical interface level. Therefore, the priority traffic is serviced first at the logical and class level. After servicing the priority traffic bandwidth, the three-level scheduler allocates unused bandwidth to the logical queues based on the configured bandwidth-remaining ratio. In this default case, the three-level scheduler allocates an equal share of the unused bandwidth to each logical queue.

The three-level scheduler supports shaping and scheduling only on the egress interface. The bandwidth command must be configured as a percentage of the available bandwidth or as an absolute bandwidth. You cannot concurrently configure the bandwidth and bandwidth remaining commands on the same class queue or the same policy map.

For more information about the bandwidth-remaining ratio, see the Distribution of Remaining Bandwidth Using Ratio feature module.

Modular QoS Command-Line Interface

The Modular Quality of Service Command-Line Interface (MQC) is designed to simplify the configuration of Quality of Service (QoS) on Cisco routers and switches by defining a common command syntax and resulting set of QoS behaviors across platforms. This model replaces the previous model of defining unique syntaxes for each QoS feature and for each platform.

The MQC contains the following three steps:

- Define a traffic class using the class-map command.
- Create a traffic policy by associating the traffic class with one or more QoS features using the policy-map command.
- Attach the traffic policy to the interface, subinterface, or virtual circuit (VC) using the service-policy command.

For more information about MQC, see the Modular Quality of Service Command-Line Interface document.

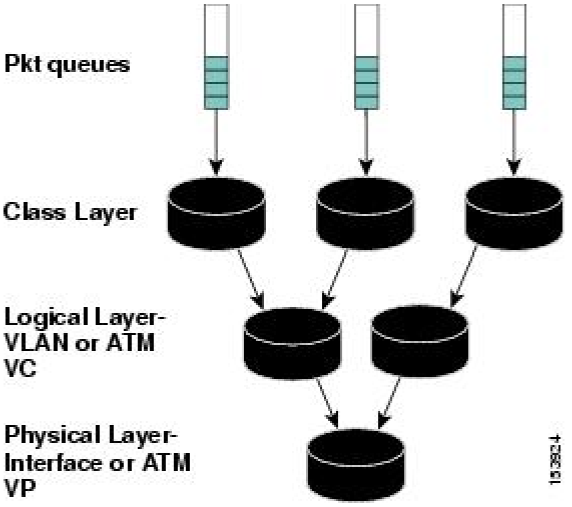
Scheduling Hierarchy

As shown in the figure below, the three-level scheduler uses the following scheduling hierarchy to allocate bandwidth for subscriber traffic:

- Class layer—The three-level scheduler uses virtual-time calendars to schedule class queues and logical interfaces.
- Logical layer (VLAN or ATM VC)—Virtual-time calendars perform weighted round robin based on the weight of the logical interface and the number of bytes dequeued.

- Physical layer (interface or ATM virtual path)—Token buckets ensure that the maximum rate for the class and the logical interface are not exceeded.

Figure 2: Scheduling Hierarchy



The figure above provides an example of how the scheduling hierarchy can apply to Ethernet and ATM topologies. For Ethernet, you cannot oversubscribe the Queue-in-Queue (qinq) into the interface. For ATM, you cannot oversubscribe the virtual path (VP) into the interface.

Table 2: Applying the Scheduling Hierarchy to Ethernet and ATM

Scheduling Hierarchy	Ethernet	ATM
Class layer (virtual time)	MQC-defined queues	MQC-defined queues
Logical layer (virtual time)	VLAN (inner tag)	Virtual channel (VC)
	Session	
Physical (real time)	Queue-in-Queue (outer tag)	Virtual path (VP)
	VLAN (inner tag), if session is the logical layer identifier	

By using VP and VC scheduling with existing Cisco 10000 ATM line cards, the scheduler supports priority propagation: cell-based VP shaping in the segmentation and reassembly (SAR) mechanism with frame-based VC scheduling in the performance routing engine 3 (PRE3).

Priority Service and Latency

The three-level scheduler supports multiple levels of priority service that you can use for such purposes as control traffic, delay-sensitive traffic (for example, voice), minimum guarantees, and excess bandwidth allocation. Each level of priority supports multiple queues, which allows for multiple types of delay-sensitive traffic (for example, voice and video).

The three-level scheduler can service the same queue from multiple levels of priority service. For example, the three-level scheduler uses priority level 1 for voice, priority level 2 for video, and the excess bandwidth for data.

For a priority class with policing configured, the three-level scheduler always polices the priority traffic to the rate specified in the police command (1000 kbps as shown in the following example configuration), regardless of whether or not the underlying interface is congested.

```
Router(config-pmap-c)# priority
Router(config-pmap-c)# police 1000
```

Note The three-level scheduler does not support the priority kbps command.

Latency Requirements

Delay-sensitive traffic incurs a maximum of 10 milliseconds (ms) of latency on edge router interfaces and a maximum of 1 ms of latency on core router interfaces. For interface speeds at T1/E1 and below, the three-level scheduler services 2 maximum transmission units (MTUs) of nonpriority traffic before servicing a priority packet. Requirements for high-speed interfaces are not as strict as 2 MTUs, but are always bound by 10 ms on edge interfaces and 1 ms on core interfaces.

The three-level scheduler also supports the minimal latency requirement (2 MTUs of nonpriority traffic in front of priority traffic) at the physical link rate. However, in some cases, it is impossible for the three-level scheduler to service all competing packets with a latency of 2 MTUs. For example, if many priority packets compete at the same time for bandwidth, the last one serviced may incur latency that is greater than 2 MTUs.

The table below lists the maximum latency requirements for various interface speeds.

Table 3: Maximum Latency Requirements

Interface Speed	Maximum Latency
Greater than 2 Mbps	2 MTU + 6 ms
2 Mbps to 1 Gbps	2 MTU
1 Gbps or greater	1 ms

Priority Propagation with Imposed Burstiness

A single physical interface can have large numbers of logical interfaces and each of these logical interfaces can have both priority and nonpriority traffic competing for the physical link. To minimize latency, the priority traffic of one logical interface has priority over the nonpriority traffic of other logical interfaces, thereby imposing burstiness on the minimum rate traffic of other logical interfaces. The latency that the priority traffic incurs results from the rate constraining the delivered rate of the priority traffic. In many cases, this constraining rate is not the rate of the priority class's parent policy.

For example, suppose a 10 Gigabit Ethernet (GE) interface has 100 VLANs that are shaped to various rates. Each VLAN has a priority class and additional classes configured. Through priority propagation, the scheduler delivers latency to the priority traffic based on the 10 GE rate and not the VLAN rate.

Note The VLAN rate is at most 1 to 2 MTUs of nonpriority traffic in front of priority traffic, which would bound the latency incurred by priority traffic (due to non-priority traffic) at 1 to 2 MTUs served at the 10 GE rate.

The priority traffic of one logical interface cannot only impose burstiness on other traffic, but also starve other traffic. The only way to prevent the starvation of other traffic is by configuring a policer on the priority queue by limiting the percent of priority traffic to less than 90 percent of the parent bandwidth and the port bandwidth.

Configuration Granularity

The table below describes the configuration granularity for the three-level scheduler.

Table 4: Three-Level Scheduler Configuration Granularity

Interface Bandwidth	Granularity
Less than or equal to 2 Mbps	.4%
Greater than 2 Mbps and less than 1 Gbps	.2%
Greater than or equal to 1 Gbps	.1%

How to Configure Bandwidth-Remaining Ratios

To configure bandwidth-remaining ratios on subinterface-level and class-level queues, see the Distribution of Remaining Bandwidth Using Ratio, Release 12.2(31)SB2 feature module.

Configuration Examples for the Three-Level Scheduler

This section provides the following configuration examples:

Bandwidth Allocation—Policy Attached to an Interface Example

The following example configuration consists of one policy map named Child with the following traffic classes defined: prec0, prec2, and class-default. The policy is attached to the ATM interface 1/0/0, which has a configured rate of 1000 kbps.

```
policy-map Child
class prec0
bandwidth 300
class prec2
```

```
bandwidth 100
class class-default
bandwidth 50
!
interface atm 1/0/0
bandwidth 1000
service-policy output Child
```

Assuming that the traffic flow through each class is enough to require maximum possible bandwidth, the three-level scheduler allocates bandwidth as described in the table below.

Table 5: Queuing Presentation—Policy Attached to an Interface

Traffic Class	Bandwidth Ratio	Total Bandwidth Allocated
prec0	6	666 kbps
prec2	2	222 kbps
class-default	1	111 kbps

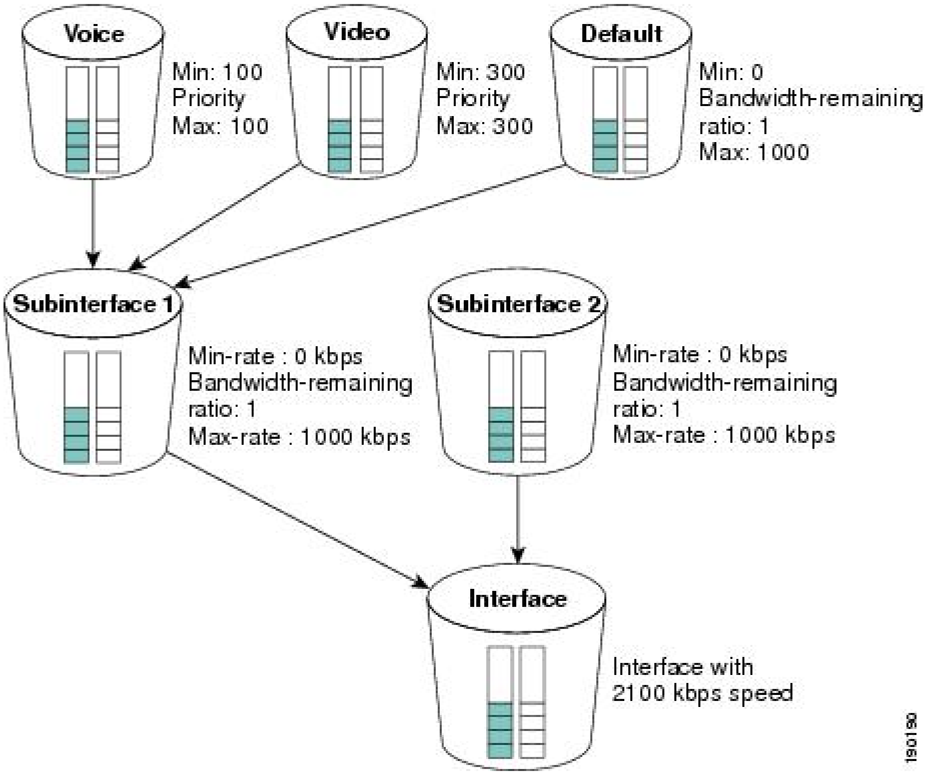
Bandwidth Allocation—Parent Policy Attached to Two Subinterfaces Example

The following example configuration contains a hierarchical policy consisting of two policy maps: Child and Parent. The Child policy has two traffic classes (voice and video) with each configured as a priority class with policing enabled. The Parent policy has its class-default class shaped to 1000 kbps. The Parent policy is attached to the ATM subinterface 1/0/1.1 and to subinterface 1/0/1.2. ATM interface 1/0/1 has a configured rate of 2100 kbps.

```
policy-map Child
class voice
priority level 1
police 100
!
class video
priority level 2
police 300
!
policy-map Parent
class class-default
shape average 1000
service-policy Child
!
interface atm 1/0/1
atm pvp 1 1400
!
interface atm 1/0/1.1
bandwidth remaining ratio 1
service-policy output Parent
!
interface atm 1/0/1.2
bandwidth remaining ratio 1
service-policy output Parent
!
```

The figure below shows an example of the queuing presentation based on the above configuration. The service rates for all Child classes under each subinterface might differ from the rates shown in the figure below, depending on the presence or absence of priority propagation and how the class's bandwidth usage is accounted against the Parent queue.

Figure 3: Queuing Presentation—Parent Enabled on Two Subinterfaces



Each subinterface receives an equal share of bandwidth. Based on the bandwidth-remaining ratio of 1, each subinterface-level queue receives a rate of 700 kbps (subinterfaces 1 and 2 queues, and default queue at subinterface-level).

- For subinterface 1, assume that only the voice traffic is active. From the 700-kbps bandwidth allocated to subinterface 1, the voice traffic receives a bandwidth rate of 100 kbps and the default traffic receives a rate of 600 kbps.
- For subinterface 2, assume that only the video traffic is active. From the 700-kbps bandwidth allocated to subinterface 2, the video traffic receives a bandwidth rate of 300 kbps and the default traffic receives a rate of 400 kbps.

Tuning the Bandwidth-Remaining Ratio Example

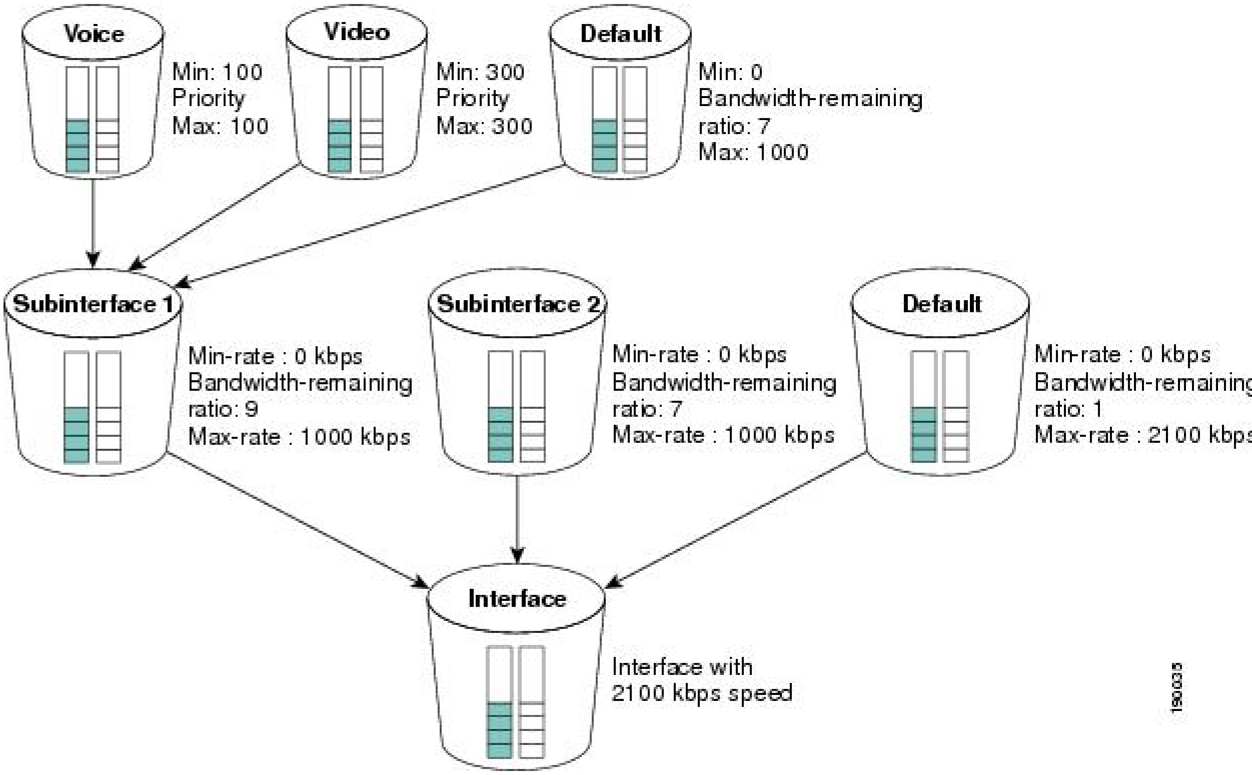
The following example configuration shows how to tune the bandwidth-remaining ratio using the bandwidth remaining ratio command. In the example, the class-default class of Parent1 has a bandwidth-remaining ratio of 9 and the class-default class of Parent2 has a bandwidth-remaining ratio of 7.

```
policy-map Child
class prec0
priority level 1
police 100
```

```
!
class prec2
priority level 2
police 300
!
policy-map Parent1
class class-default
shape average 10000
bandwidth remaining ratio 9
!
policy-map Parent2
class class-default
shape average 1000
bandwidth remaining ratio 7
```

The figure below shows an example of the queuing presentation based on the above configuration and assuming that the Parent1 policy is enabled on subinterface 1 and the Parent2 policy is enabled on subinterface 2, and that the interface speed is 2100 kbps.

Figure 4: Queuing Presentation—Tuning the Bandwidth-Remaining Ratio



Based on the preceding configuration, the three-level scheduler distributes bandwidth in the following way (assuming that the voice traffic is active on subinterface 1 only and the video traffic is active on subinterface 2 only):

- A total of 400 kbps of bandwidth is used from the interface: 100 kbps-bandwidth guarantee for voice traffic on subinterface 1 and 300-kbps bandwidth guarantee for video traffic on subinterface 2.
- The remaining 1700-kbps bandwidth is distributed across the subinterface-level queues based on their bandwidth-remaining ratios:
    - Subinterface 1 with bandwidth-remaining ratio 9 receives 956 kbps

- Subinterface 2 with bandwidth-remaining ratio 7 receives 743 kbps

Additional References

The following sections provide references related to the MQC Hierarchical Queuing with 3 Level Scheduler feature.

Related Documents

Related Topic	Document Title
Bandwidth	*Cisco 10000 Series Router Quality of Service Configuration Guide* “Distributing Bandwidth Between Queues”
Bandwidth-remaining ratio	“Distribution of Remaining Bandwidth Using Ratio” feature module
Hierarchical policies	*Cisco 10000 Series Router Quality of Service Configuration Guide* “Defining QoS for Multiple Policy Levels”
Policy maps	*Cisco 10000 Series Router Quality of Service Configuration Guide* “Configuring QoS Policy Actions and Rules”
Shaping traffic	*Cisco IOS Quality of Service Solutions Configuration Guide*, Release 12.2 Part 4: Policing and Shaping > Configuring Class-Based Shaping Part 4: Policing and Shaping > Policing and Shaping Overview > Traffic Shaping > Class-Based Shaping
Traffic policing and shaping	Comparing Traffic Policing and Traffic Shaping for Bandwidth Limiting

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 Technical Support & Documentation website contains thousands of pages of searchable technical content, including links to products, technologies, solutions, technical tips, and tools. Registered Cisco.com users can log in from this page to access even more content.	http://www.cisco.com/techsupport