



## CHAPTER 12

# Sharing Bandwidth Fairly During Congestion

---

When no QoS policies exist, the router serves traffic with “best effort” service. The router makes no distinction between high and low priority traffic and makes no allowances for the needs of different applications. This is not a problem until congestion occurs and response time slows—larger packets take longer to transmit and smaller packets are delayed behind larger packets.

To provide consistent response time to heavy and light traffic without adding excessive bandwidth, the Cisco 10000 series router provides a queuing technique called Class-Based Weighted Fair Queuing (CBWFQ). This technique reduces response time for real-time traffic and fairly shares the remaining bandwidth between high bandwidth flows.

This chapter describes CBWFQ and includes the following topics:

- [Class-Based Weighted Fair Queuing, page 12-1](#)
- [Class-Based Weighted Fair Queuing for Virtual Access Interfaces, page 12-4](#)
- [System Limits for CBWFQ, page 12-6](#)
- [Interfaces Supporting Class-Based Weighted Fair Queuing, page 12-7](#)
- [Configuring Fair Bandwidth Sharing During Congestion, page 12-7](#)
- [Configuration Examples for Sharing Bandwidth Fairly, page 12-18](#)
- [Verifying and Monitoring Class-Based Weighted Fair Queuing, page 12-20](#)
- [Related Documentation, page 12-20](#)

## Class-Based Weighted Fair Queuing

Class-Based Weighted Fair Queuing (CBWFQ) is an automated scheduling method that provides fair bandwidth allocation to all network traffic. CBWFQ applies priority, or weights, to identified traffic to classify traffic into conversations and determine how much bandwidth each conversation is allowed relative to other conversations. CBWFQ provides consistent response time to heavy and light traffic without adding excessive bandwidth. CBWFQ reduces response time for real-time traffic and fairly shares the remaining bandwidth between high bandwidth flows.

CBWFQ provides support for user-defined traffic classes. The router classifies traffic based on match criteria you define for each class (for example, protocols, access control lists (ACLs), and input interfaces). Packets satisfying the match criteria for a class constitute the traffic for that class. The router reserves a first-in first-out (FIFO) queue for each class and directs traffic belonging to a class to the queue for that class.

After defining a class according to its match criteria, you can assign it characteristics by assigning it bandwidth, weight, and maximum packet limit. The bandwidth you assign to a class is the guaranteed bandwidth delivered to the class during congestion. You can also specify the maximum number of packets allowed to accumulate in the queue for a class, referred to as the queue limit for the class. Packets belonging to a class are subject to the bandwidth and queue limits that characterize the class.

After a queue reaches its configured queue limit, enqueueing of additional packets to the class causes packet drop to occur. The router drops packets using one of the following methods, depending on how you configured the traffic class:

- Tail drop—The default congestion avoidance mechanism for Layer 3 queues. Tail drop activates when a queue becomes full. After being activated, no packets make it to the queue. Tail drop treats all traffic equally and does not differentiate between classes of service.
- Weighted Random Early Detection (WRED)—A mechanism for avoiding congestion of Layer 3 queues. WRED combines the capabilities of the random early detection (RED) mechanism with IP precedence, differential services code point (DSCP), and discard-class to provide preferential handling of higher priority packets. WRED attempts to anticipate and avoid congestion. WRED implements a proactive queuing strategy that manages congestion before a queue reaches its queue depth. By selectively dropping packets, WRED prevents packets from enqueueing to the Layer 3 queue.

**Note**

---

If you use WRED packet drop instead of tail drop for one or more traffic classes in a policy map, the interface to which you attach that policy map cannot have WRED configured.

---

If you configure the class-default class using the **bandwidth** command, the router places all unclassified traffic into a single FIFO queue and allocates bandwidth according to the configured bandwidth. If you do not configure the class-default class, then by default the router gives best-effort treatment to the traffic that does not match any of the configured classes. After the router classifies a packet, all of the standard mechanisms that you can use to differentiate service among the classes apply.

For CBWFQ, the weight specified for the class becomes the weight of each packet that meets the match criteria of the class. The router classifies packets that arrive at the output interface according to the match criteria filters you define. The router then assigns each one the appropriate weight. The router derives the weight for a packet belonging to a specific class from the bandwidth you assign to the class when you configure it. In this sense the weight for a class is user-configurable.

After the router assigns the weight for a packet, the router enqueues the packet in the appropriate class queue. CBWFQ uses the weights assigned to the queued packets to ensure that the router services the class queue fairly.

You can configure CBWFQ on a physical interface only if the interface is in the default queuing mode. Serial interfaces at E1 (2.048 Mbps) and below use WFQ by default; other interfaces use FIFO by default. Enabling CBWFQ on a physical interface overrides the default interface queuing method. Enabling CBWFQ on an ATM permanent virtual circuit (PVC) does not override the default queuing method.

Configuring CBWFQ involves the following processes:

- Classifying traffic—This process uses class maps to define the classification criteria the router uses to differentiate one traffic class from another.
- Associating class characteristics with each traffic class—This process uses policy maps to define the class characteristics (policy actions) the router applies to packets belonging to one of the traffic classes.
- Attaching policies to interfaces—This process uses the **service-policy** command to associate an existing policy map (service policy) with an interface. The router applies the policy actions defined in the service policy to the traffic on the interface that belongs to the traffic classes defined in the service policy.

## Feature History for Class-Based Weighted Fair Queuing

Cisco IOS Release	Description	Required PRE
Release 12.0(19)SL	The class-based weighted fair queuing (CBWFQ) feature was introduced on the router.	PRE1
Release 12.2(16)BX	This feature was introduced on the PRE2.	PRE2
Release 12.2(28)SB	This feature was integrated in Cisco IOS Release 12.2(28)SB for the PRE2.	PRE2
Release 12.2(31)SB2	This feature was introduced on the PRE3.	PRE3

## Class-Default Class

The *class-default* class is used to classify traffic that does not fall into one of the defined classes in a policy map. After the router classifies a packet, the router applies all the standard mechanisms that are used to differentiate service among the classes. The class-default class is predefined when you create the policy map, but you must configure it. If you do not configure the default class, then by default the traffic that does not match any of the configured classes in a policy map is FIFO-classified and given best-effort treatment.

## CBWFQ and Bandwidth Allocation

CBWFQ allows you to specify the exact amount of bandwidth to allocate for a specific class of traffic. Distributing bandwidth on a link using the **bandwidth** command ensures that bandwidth is shared fairly among competing traffic. The router uses class queues to allocate bandwidth, first servicing priority queue traffic followed by either bandwidth guarantee or bandwidth remaining queue traffic. By default, a minimum bandwidth guaranteed queue has buffers for up to 50 milliseconds of 256-byte packets at line rate, but not less than 32 packets. The router does not ensure latency characteristics for bandwidth queues.

After the router allocates bandwidth to priority and bandwidth guaranteed class queues, the router divides unused (excess) bandwidth among the packets remaining in the class queues.

For more information about distributing bandwidth across class queues, including how bandwidth is calculated, see [Chapter 5, “Distributing Bandwidth Between Queues.”](#)

## CBWFQ and RSVP

You can use Resource Reservation Protocol (RSVP) in conjunction with CBWFQ. When you configure both RSVP and CBWFQ for an interface, RSVP and CBWFQ act independently, exhibiting the same behavior that they would if each were running alone. RSVP continues to work as it does when CBWFQ is not present, even in regard to bandwidth availability assessment and allocation.

## Restrictions and Limitations for CBWFQ

- If you configure a policy map class to use weighted random early detection (WRED), the interface to which you attach the service policy cannot have WRED configured.
- The router does not support traffic shaping and policing with CBWFQ.
- The router supports CBWFQ on variable bit rate (VBR) ATM connections.

# Class-Based Weighted Fair Queuing for Virtual Access Interfaces

Class-based weighted fair queuing (CBWFQ) for virtual access interfaces (VAIs) allows a VAI to inherit the service policy of the VC that the VAI uses (see the [“Service Policy Inheritance”](#) section on page 12-5). Using CBWFQ, you can:

- Configure user-defined traffic classes
- Use access control lists (ACLs), protocols, or input interface names to define how to classify traffic
- Specify the exact amount of bandwidth to be allocated for a specific class of traffic

To use CBWFQ, you define traffic classes based on match criteria. Packets satisfying the match criteria for a class constitute the traffic for that class. CBWFQ reserves a FIFO for each class and directs traffic belonging to a class to the queue for that class.

After you define the match criteria for a traffic class, you can assign the class characteristics. To characterize a class, you create a policy map and assign each class such parameters as bandwidth and queue limit. The bandwidth is the guaranteed bandwidth delivered to the class during congestion. The bandwidth assigned to a class is used to derive a weight for the class. Each packet that meets the match criteria of the class is assigned the weight of the class and is then enqueued in the appropriate class queue. CBWFQ uses the weights assigned to the queued packets to ensure that the class queue is serviced fairly.

You can also specify the queue limit for a class to indicate the maximum number of packets allowed to accumulate in the queue for the class. Packets belonging to a class are subject to the queue limits that characterize the class.

The following apply to CBWFQ functionality:

- The definition and application of policies in a policy map are in a single direction, either input or output.
- Rules apply both to input and output policies.
- The first policy defined is the one applied for a given VAI.
- Policy behavior of native VC traffic remains unchanged.
- The order of definition and removal of policies is not stateful.

For more information about CBWFQ, see the [“Class-Based Weighted Fair Queuing”](#) section on page 12-1.

## Feature History for CBWFQ for VAIs

Cisco IOS Release	Description	Required PRE
Release 12.0(25)SX	This feature was introduced on the router.	PRE1
Release 12.2(16)BX	This feature was introduced on the PRE2.	PRE2
Release 12.2(28)SB	This feature was integrated in Cisco IOS Release 12.2(28)SB for the PRE2.	PRE2

## Service Policy Inheritance

CBWFQ allows you to configure service policies on VC interfaces or subinterfaces. When a remote user initiates a session, the router uses a predefined configuration template to dynamically create and configure a virtual access interface (VAI). The VAI uses the attributes of the template to create the session, which results in a VAI that is uniquely configured for a specific user.

For CBWFQ, when you apply a service policy to a VC, the VAIs that use that VC inherit the service policy of the VC. Any VAI that uses that VC is subject to the queuing, policing, and marking actions defined in the VC service policy.

You can only apply a service policy with queuing-related actions to a VC. Do not apply service policies with CBWFQ actions to a VAI using a virtual template. The Cisco 10000 router supports queuing only when you apply the service policy to a VC.



### Note

You can apply a service policy without queuing-related actions to either a VC or a VAI, but not to both at the same time.

## Restrictions and Limitations for CBWFQ for VAIs

- Virtual template interfaces and VAIs do not apply to routed bridge encapsulation (RBE) over ATM.
- Both the virtual template and virtual connection must exist before a remote user initiates a session to the router.
- Cisco IOS Release 12.2(25)SX does not support the configuration of broadband aggregation (BBA) groups using RADIUS. You must configure BBA groups manually.
- You can only apply a QoS policy with queuing-related actions to a VC. Do not apply service policies with class-based weighted fair queuing (CBWFQ) actions to a VAI using a virtual template. The router supports queuing only when you apply the QoS policy to a VC.
- You can apply a QoS policy without queuing-related actions to either a VC or a VAI, but not to both at the same time.
- You cannot use RADIUS to configure a QoS policy on the VC.
- If you configure a QoS policy on a VC, the **show policy interface vai** command does not display information to indicate that the VAI is subject to the VC service policy. However, if you apply a policy directly to a VAI, the **show policy interface vai** command displays information about the policy on the VAI.
- The port must be in **atm pxf queuing** mode.
- You cannot configure a service policy on a VC and on a session at the same time.

## System Limits for CBWFQ

For information about the system limits for class maps and policy maps configured for CBWFQ, see the [“System Limits for Class Maps”](#) section on page 2-2 and the [“System Limits for Policy Maps”](#) section on page 3-2.

# Interfaces Supporting Class-Based Weighted Fair Queuing

The following describes interface support for class-based weighted fair queuing (CBWFQ):

## Interfaces Supporting CBWFQ (Outbound Only)

- Physical
- Multilink PPP and Multilink Frame Relay
- ATM shaped (peak cell rate is specified) unspecified bit rate (UBR) PVCs and point-to-point subinterfaces
- ATM constant bit rate (CBR) PVCs and point-to-point subinterfaces
- ATM variable bit rate (VBR) PVCs and point-to-point subinterfaces
- Label-controlled ATM (LC-ATM) subinterfaces \*
- Frame Relay PVCs, point-to-point subinterfaces, and map classes \*
- Ethernet VLANs \*

\* Requires a specific type of hierarchical policy. For more information, see [Chapter 13, “Defining QoS for Multiple Policy Levels.”](#)

**Note**

---

The router only supports CBWFQ on outbound interfaces.

---

## Interfaces Not Supporting CBWFQ

- ATM unshaped (no peak cell rate specified) UBR PVCs and point-to-point subinterfaces
- IP tunnel
- Virtual-access (See the [“VAI QoS Inheritance”](#) section on page 4-24.)

**Note**

---

The router does not support the CBWFQ on inbound interfaces.

---

# Configuring Fair Bandwidth Sharing During Congestion

To configure fair bandwidth sharing using QoS service policies, perform the following required configured tasks:

- [Defining Traffic Classes Using Class Maps, page 12-8](#)
- [Configuring Policy Actions for Traffic Classes Using Policy Maps, page 12-9](#)
- [Attaching Service Policies, page 12-15](#)

The following are optional configuration tasks:

- [Modifying the Bandwidth for an Existing Policy Map Class, page 12-16](#)
- [Modifying the Queue Limit for an Existing Policy Map Class, page 12-17](#)
- [Deleting Classes, page 12-17](#)
- [Deleting Policy Maps, page 12-18](#)

## Defining Traffic Classes Using Class Maps

To define a traffic class using a class map, enter the following commands beginning in global configuration mode:

	Command	Purpose
Step 1	Router(config)# <b>class-map</b> <i>class-map-name</i>	Creates or modifies a class map. Enters class-map configuration mode.  <i>class-map-name</i> is the name of the class map.
Step 2	Router(config-cmap)# <b>match access-group</b> { <i>access-group-number</i>   <b>name</b> <i>access-group-name</i> }	Specifies the name of the access control list (ACL) against whose contents packets are checked to determine if they belong to the class. CBWFQ supports numbered and named ACLs.  <i>access-group-number</i> is the number of the access control list (ACL) against whose contents you want packets to be checked. <b>name</b> <i>access-group-name</i> is the name of the ACL.
	or	
	Router(config-cmap)# <b>match</b> <b>input-interface</b> <i>interface-name</i>	Specifies the name of the input interface used as a match criterion against which packets are checked to determine if they belong to the class.  <i>interface-name</i> is the type and number of the interface (for example, ATM 1/0/0).
	or	
	Router(config-cmap)# <b>match mpls</b> <b>experimental</b> <i>number</i>	Specifies the value of the experimental (EXP) field to be used as a match criterion against which packets are checked to determine if they belong to the class.  <i>number</i> is the EXP value to be used as a match criterion. Valid values are from 0 to 7, delimited using spaces (for example, 3 4 7).  For more information about match criteria you can define, see the <a href="#">“Defining Match Criteria Using the match Commands”</a> section on page 2-5.

### Configuration Example for Defining Traffic Classes Using Class Maps

**Example 12-1** creates two ACLs (101 and 102) and two class maps (*class1* and *class2*). To determine if packets belong to the class, *class1* matches packets using ACL 101 and *class2* uses ACL 102 match criteria.

#### Example 12-1 Defining Traffic Classes Using a Class Map

```
Router(config)# access-list 101 permit udp host 10.10.10.10 host 10.10.10.20 range 16384
20000
Router(config)# access-list 102 permit udp host 10.10.10.10 host 10.10.10.20 range 53000
56000
Router(config)# class-map class1
Router(config-cmap)# match access-group 101
Router(config-cmap)# class-map class2
Router(config-cmap)# match access-group 102
Router(config-cmap)# exit
```

## Configuring Policy Actions for Traffic Classes Using Policy Maps

To configure policy actions for a traffic class using a policy map, perform the following configuration tasks:

- [Configuring a Default Traffic Class Policy](#), page 12-9
- [Configuring a Class Policy and Dropping Packets Using Tail Drop](#), page 12-11
- [Configuring a Class Policy and Dropping Packets Using WRED](#), page 12-12

For more information about QoS policies, see [Chapter 3, “Configuring QoS Policy Actions and Rules.”](#)

### Configuring a Default Traffic Class Policy

The router uses the policy actions defined in the default class named *class-default* for traffic that does not match the criteria for a specific class or when a QoS policy does not exist for a traffic class.

To configure a default traffic class policy, enter the following commands beginning in global configuration mode:

	Command	Purpose
Step 1	Router(config)# <b>policy-map</b> <i>policy-map-name</i>	Creates or modifies a policy map. Enters policy-map configuration mode.  <i>policy-map-name</i> is the name of the policy map.
Step 2	Router(config-pmap)# <b>class class-default</b>	Specifies the default class so that you can configure or modify its policy. Enters policy-map class configuration mode.
Step 3	Router(config-pmap-c)# <b>bandwidth</b> { <i>bandwidth-kbps</i>   <b>percent</b> <i>percentage</i>   <b>remaining percent</b> <i>percentage</i> }	Specifies the amount of bandwidth (in kbps or as a percentage of available bandwidth) to be assigned to the class. The amount of bandwidth configured should be large enough to also accommodate Layer 2 overhead.  <i>bandwidth-kbps</i> specifies or modifies the minimum bandwidth allocated for a class belonging to a policy map. Valid values are from 8 to 2,488,320, which represents from 1 to 99 percent of the link bandwidth.  <b>Note</b> The range of valid values for <i>bandwidth-kbps</i> might be smaller than the values indicated above. Use the question mark (?) in context-sensitive help to display the range of valid values.  <b>percent</b> <i>percentage</i> specifies or modifies the minimum percentage of the link bandwidth allocated for a class belonging to a policy map. Valid values are from 1 to 99.  <b>remaining percent</b> <i>percentage</i> specifies or modifies the minimum percentage of unused link bandwidth allocated for a class belonging to a policy map. Valid values are from 1 to 99.

Command	Purpose
<b>Step 4</b> Router(config-pmap-c) # <b>queue-limit</b> <i>number-of-packets</i>	(Optional) Specifies the maximum number of packets that the queue can accumulate for this class.  <i>number-of-packets</i> is a number from 1 to 64. The default number of queue entries is based on the bandwidth rate.  <b>Note</b> When you specify the <b>queue-limit</b> command, the router uses tail drop to drop packets.
<b>Step 5</b> Router(config-pmap-c) # <b>random-detect</b>	(Optional) Enables WRED.  <b>Note</b> When you specify the <b>random-detect</b> command, the policy map uses WRED to drop packets, not tail drop. If you configure a class to use WRED, you must ensure that WRED is not configured on the interface to which you intend to attach the service policy.
<b>Step 6</b> Router(config-pmap-c) # <b>random-detect</b> <b>exponential-weighting-constant</b> <i>exponent</i>  or  Router(config-pmap-c) # <b>random-detect</b> <b>precedence</b> <i>precedence min-threshold</i> <i>max-threshold mark-prob-denominator</i>	Configures the exponential weight factor used in calculating the average queue length.  <i>exponent</i> is a number from 1 to 16 used in the average queue size calculation.  Configures WRED parameters for packets with a specific IP precedence. Repeat this command for each precedence.  <i>precedence</i> is the IP precedence number. Valid values are from 0 to 7.  <i>min-threshold</i> is the minimum average queue length, expressed in number of packets. Valid values are from 1 to 4096. When the average queue length reaches the minimum threshold, WRED randomly drops some packets with the specified IP precedence.  <i>max-threshold</i> is the maximum average queue length, expressed in number of packets. Valid values are from the value of the min-threshold to 4096. When the average queue length exceeds the maximum threshold, WRED drops all packets with the specified IP precedence.  <i>mark-prob-denominator</i> is the denominator for the fraction of packets dropped when the average queue depth is at the maximum threshold. For example, if the denominator is 512, 1 out of every 512 packets is dropped when the average queue is at the maximum threshold. Valid values are from 1 to 65536. The default is 10 (1 out of every 10 packets is dropped at the maximum threshold).

## Configuring a Class Policy and Dropping Packets Using Tail Drop

To configure a class policy and drop packets using tail drop, enter the following commands beginning in global configuration mode:



**Note**

Repeat Steps 2 through 4 to configure additional classes in the policy map.

	Command	Purpose
<b>Step 1</b>	Router(config)# <b>policy-map</b> <i>policy-map-name</i>	Creates or modifies a policy map. Enters policy-map configuration mode.  <i>policy-map-name</i> is the name of the policy map.
<b>Step 2</b>	Router(config-pmap)# <b>class</b> <i>class-map-name</i>	Assigns a traffic class to a policy map. Enters policy-map class configuration mode.  <i>class-map-name</i> is the name of a previously configured class map.
<b>Step 3</b>	Router(config-pmap-c)# <b>bandwidth</b> { <i>bandwidth-kbps</i>   <b>percent</b> <i>percentage</i>   <b>remaining percent</b> <i>percentage</i> }	Specifies the amount of bandwidth (in kbps or as a percentage of available bandwidth) to be assigned to the class. The amount of bandwidth configured should be large enough to also accommodate Layer 2 overhead.  <i>bandwidth-kbps</i> specifies or modifies the minimum bandwidth allocated for a class belonging to a policy map. Valid values are from 8 to 2,488,320, which represents from 1 to 99 percent of the link bandwidth.  <b>Note</b> The range of valid values for <i>bandwidth-kbps</i> might be smaller than the values indicated above. Use the question mark (?) in context-sensitive help to display the range of valid values.  <b>percent</b> <i>percentage</i> specifies or modifies the minimum percentage of the link bandwidth allocated for a class belonging to a policy map. Valid values are from 1 to 99.  <b>remaining percent</b> <i>percentage</i> specifies or modifies the minimum percentage of unused link bandwidth allocated for a class belonging to a policy map. Valid values are from 1 to 99.
<b>Step 4</b>	Router(config-pmap-c)# <b>queue-limit</b> <i>number-of-packets</i>	Specifies the maximum number of packets that the queue can accumulate for this class.  <i>number-of-packets</i> is a number from 1 to 64. The default number of queue entries is based on the bandwidth rate.  <b>Note</b> When you specify the <b>queue-limit</b> command, the router uses tail drop to drop packets.

## Configuration Example for Configuring a Class Policy and Dropping Packets Using Tail Drop

Example 12-2 creates a policy map named *policy1* that contains two classes (*class1* and *class2*) whose match criteria were previously defined (see Example 12-1 on page 12-8). The class1 configuration requests a specific bandwidth allocation and specifies the maximum number of packets that can be queued for the class. Because the class1 configuration specifies the **queue-limit** command, the router uses tail drop to drop packets. The class2 configuration specifies only the bandwidth allocation request; therefore, the policy map assumes a default queue limit based on the configured bandwidth rate. The policy1 service policy is applied to PVC 1/32 for outbound packets.

### Example 12-2 Configuring a Class Policy and Dropping Packets Using Tail Drop

```
Router(config)# policy-map policy1
Router(config-pmap)# class class1
Router(config-pmap-c)# bandwidth 3000
Router(config-pmap-c)# queue-limit 32
Router(config-pmap-c)# class class2
Router(config-pmap-c)# bandwidth 2000
Router(config-pmap-c)# exit
Router(config-pmap)# exit
Router(config)# interface atm 1/0/0
Router(config-if)# atm pxf queuing
Router(config-if)# interface atm 1/0/0.1 point-to-point
Router(config-subif)# pvc 1/32
Router(config-subif-atm-vc)# service-policy output policy1
```

## Configuring a Class Policy and Dropping Packets Using WRED

To configure a class policy and to drop packets using WRED, enter the following commands beginning in global configuration mode:



### Note

Repeat Steps 2 through 5 to assign additional traffic classes to the policy map and to configure a class policy for the traffic classes.

	Command	Purpose
Step 1	Router(config)# <b>policy-map</b> <i>policy-map-name</i>	Creates or modifies a policy map. Enters policy-map configuration mode.  <i>policy-map-name</i> is the name of the policy map.
Step 2	Router(config-pmap)# <b>class</b> <i>class-map-name</i>	Assigns a traffic class to a policy map. Enters policy-map class configuration mode.  <i>class-map-name</i> is the name of a previously configured class map.

	Command	Purpose
<b>Step 3</b>	<pre>Router(config-pmap-c)# <b>bandwidth</b> {<i>bandwidth-kbps</i>   <b>percent</b> <i>percentage</i>   <b>remaining percent</b> <i>percentage</i>}</pre>	<p>Specifies the amount of bandwidth (in kbps or as a percentage of available bandwidth) to be assigned to the class. The amount of bandwidth configured should be large enough to also accommodate Layer 2 overhead.</p> <p><i>bandwidth-kbps</i> specifies or modifies the minimum bandwidth allocated for a class belonging to a policy map. Valid values are from 8 to 2,488,320, which represents from 1 to 99 percent of the link bandwidth.</p> <p><b>Note</b> The range of valid values for <i>bandwidth-kbps</i> might be smaller than the values indicated above. Use the question mark (?) in context-sensitive help to display the range of valid values.</p> <p><b>percent percentage</b> specifies or modifies the minimum percentage of the link bandwidth allocated for a class belonging to a policy map. Valid values are from 1 to 99.</p> <p><b>remaining percent percentage</b> specifies or modifies the minimum percentage of unused link bandwidth allocated for a class belonging to a policy map. Valid values are from 1 to 99.</p>
<b>Step 4</b>	<pre>Router(config-pmap-c)# <b>random-detect</b></pre>	<p>Enables weighted random early detection (WRED).</p> <p><b>Note</b> When you specify the <b>random-detect</b> command, the policy map uses WRED to drop packets, not tail drop. If you configure a class to use WRED, you must ensure that WRED is not configured on the interface to which you intend to attach the service policy.</p>

Command	Purpose
<b>Step 5</b> Router(config-pmap-c)# <b>random-detect exponential-weighting-constant</b> <i>exponent</i>  or  Router(config-pmap-c)# <b>random-detect precedence</b> <i>precedence min-threshold max-threshold mark-prob-denominator</i>	Configures the exponential weight factor used in calculating the average queue length.  <i>exponent</i> is a number from 1 to 16 used in the average queue size calculation.  Configures WRED parameters for packets with a specific IP precedence. Repeat this command for each precedence.  <i>precedence</i> is the IP precedence number. Valid values are from 0 to 7.  <i>min-threshold</i> is the minimum average queue length, expressed in number of packets. Valid values are from 1 to 4096. When the average queue length reaches the minimum threshold, WRED randomly drops some packets with the specified IP precedence.  <i>max-threshold</i> is the maximum average queue length, expressed in number of packets. Valid values are from the value of the min-threshold to 4096. When the average queue length exceeds the maximum threshold, WRED drops all packets with the specified IP precedence.  <i>mark-prob-denominator</i> is the denominator for the fraction of packets dropped when the average queue depth is at the maximum threshold. For example, if the denominator is 512, 1 out of every 512 packets is dropped when the average queue is at the maximum threshold. Valid values are from 1 to 65536. The default is 10 (1 out of every 10 packets is dropped at the maximum threshold).

### Configuration Example for Configuring a Class Policy and Dropping Packets Using WRED

[Example 12-3](#) creates the class map named *class1* and defines the match criteria used to determine if packets belong to the class. The policy map named *policy1* contains the class characteristics for *class1*. Because the *class1* configuration specifies the **random-detect** command, the policy map uses WRED packet drop to drop packets. The service policy for *policy1* is applied to the PVC range 1/32 to 1/81 for outbound packets.

#### Example 12-3 Configuring a Class Policy and Dropping Packets Using WRED

```

Router(config)# class-map class1
Router(config-cmap)# match input-interface Ethernet0/1
Router(config-cmap)# exit
Router(config)# policy-map policy1
Router(config-pmap)# class class1
Router(config-pmap-c)# bandwidth 1000
Router(config-pmap-c)# random-detect
Router(config-pmap-c)# exit
Router(config-pmap)# exit
Router(config)# interface atm 1/0/0
Router(config-if)# atm pxf queuing
Router(config-if)# interface atm 1/0/0.1 point-to-point
Router(config-subif)# range pvc 1/32 1/81
Router(config-subif-range-vc)# service-policy output policy1
!
```

## Attaching Service Policies

To attach a service policy, perform any of the following configuration tasks:

- [Attaching a Service Policy to an Interface, Subinterface, or PVC](#), page 12-15
- [Attaching a Service Policy to a Virtual Access Interface](#), page 12-16

For more information, see [Chapter 4, “Attaching Service Policies.”](#)

### Attaching a Service Policy to an Interface, Subinterface, or PVC

To attach a service policy to an interface, subinterface, or PVC, enter the following command in the appropriate configuration mode:

Command	Purpose
<pre>Router(config-if)# <b>service-policy</b> [input   output] policy-map-name</pre>	<p>Attaches the policy map you specify to an interface.</p> <p><b>input</b> indicates to apply the service policy to inbound packets.</p> <p><b>output</b> indicates to apply the service policy to outbound packets.</p> <p><i>policy-map-name</i> is the name of the policy map whose QoS policies you want to apply to the interface, subinterface, or PVC.</p> <p><b>Note</b> When you attach a service policy to an interface and the policy map configuration enables CBWFQ, all classes configured as part of the policy map are installed in the fair queuing system.</p>

#### Configuration Example for Attaching a Service Policy to an Interface

As shown in [Example 12-4](#), you can attach the same policy map to multiple interfaces. [Example 12-4](#) attaches the policy map named *policy1* to the Ethernet 1/0/0 and serial 1/0/1 interfaces for outbound packets. Each interface can have only one policy attached for inbound packets and one policy attached for outbound packets.

##### **Example 12-4 Attaching a Service Policy to an Interface**

```
Router(config)# interface ethernet1/0/0
Router(config-if)# service-policy output policy1
Router(config-if)# interface serial1/0/1
Router(config-if)# service-policy output policy1
Router(config-if)# exit
```

## Attaching a Service Policy to a Virtual Access Interface

To attach a service policy to a virtual access interface (VAI), see the [“Attaching Virtual Access Interface QoS Service Policies”](#) section on page 4-23.



### Note

To configure class-based WFQ (CBWFQ) for VAIs, you must specify the **bandwidth** command for the traffic classes configured in the policy map and attach the policy map to either a VC or to a virtual template interface (requires Cisco IOS Release 12.2(16)BX and later releases).

## Modifying the Bandwidth for an Existing Policy Map Class

To change the amount of bandwidth allocated for an existing class, enter the following commands beginning in global configuration mode:

	Command	Purpose
Step 1	Router(config)# <b>policy-map</b> <i>policy-map-name</i>	Specifies the name of the policy map containing the class you want to modify. Enters policy-map configuration mode.  <i>policy-map-name</i> is the name of the policy map.
Step 2	Router(config-pmap)# <b>class</b> <i>class-map-name</i>	Specifies the name of the class whose bandwidth you want to modify. Enters policy-map class configuration mode.  <i>class-map-name</i> is the name of the class map.
Step 3	Router(config-pmap-c)# <b>bandwidth</b> { <i>bandwidth-kbps</i>   <b>percent</b> <i>percentage</i>   <b>remaining percent</b> <i>percentage</i> }	Specifies the changed amount of bandwidth (in kbps or as a percentage of available bandwidth) to be assigned to the class. The amount of bandwidth configured should be large enough to also accommodate Layer 2 overhead.  <i>bandwidth-kbps</i> specifies the minimum bandwidth allocated for a traffic class. Valid values are from 8 to 2,488,320, which represents from 1 to 99 percent of the link bandwidth.  <b>Note</b> The range of valid values for <i>bandwidth-kbps</i> might be smaller than the values indicated above. Use the question mark (?) in context-sensitive help to display the range of valid values.  <b>percent</b> <i>percentage</i> specifies the minimum percentage of the link bandwidth allocated for a class belonging to a policy map. Valid values are from 1 to 99.  <b>remaining percent</b> <i>percentage</i> specifies the minimum percentage of unused link bandwidth allocated for a class belonging to a policy map. Valid values are from 1 to 99

## Modifying the Queue Limit for an Existing Policy Map Class

To change the maximum number of packets that can accrue in a queue reserved for an existing class, enter the following commands beginning in global configuration mode:

	Command	Purpose
Step 1	Router(config)# <b>policy-map</b> <i>policy-map-name</i>	Creates or modifies a policy map. Enters policy-map configuration mode.  <i>policy-map-name</i> is the name of the policy map containing the class you want to modify.
Step 2	Router(config-pmap)# <b>class</b> <i>class-map-name</i>	Creates or modifies a traffic class in a policy map. Enters policy-map class configuration mode.  <i>class-map-name</i> is the name of the class whose queue limit you want to modify. This is the name of a previously configured class map.
Step 3	Router(config-pmap-c)# <b>queue-limit</b> <i>number-of-packets</i>	Specifies the maximum number of packets that the queue can accumulate for this class.  <i>number-of-packets</i> is a number from 1 to 64. The default number of queue entries is based on the bandwidth rate.  <b>Note</b> When you specify the <b>queue-limit</b> command, the router uses tail drop to drop packets.

## Deleting Classes

To delete one or more classes from a policy map, enter the following commands beginning in global configuration mode:

	Command	Purpose
Step 1	Router(config)# <b>policy-map</b> <i>policy-map-name</i>	Creates or modifies a policy map. Enters policy-map configuration mode.  <i>policy-map-name</i> is the name of the policy map containing the classes you want to delete.
Step 2	Router(config-pmap)# <b>no class</b> <i>class-map-name</i>	Deletes the class you specify.  <i>class-map-name</i> is the name of the traffic class you want to delete. This is the name of a previously configured class map.

## Deleting Policy Maps

To delete a policy map, enter the following command in global configuration mode:

Command	Purpose
Router(config)# <b>no policy-map</b> <i>policy-map-name</i>	Deletes the policy map you specify. <i>policy-map-name</i> is the name of the policy map.

## Configuration Examples for Sharing Bandwidth Fairly

This section provides the following configuration examples:

- [Configuration Example for Configuring CBWFQ and Attaching a Policy to an ATM Subinterface, page 12-19](#)
- [Configuration Example for Configuring CBWFQ and Attaching a Policy to an RBE Subinterface, page 12-19](#)

### Configuration Example for Configuring CBWFQ and Attaching a Policy to an Ethernet Interface

[Example 12-5](#) shows how to configure CBWFQ in a policy map and attach the policy to an Ethernet interface. The example configures a classification policy named *voip* and a policy map named *policy1*, which defines the class characteristics for the *voip*, *video*, and *class-default* classes. The service policy is attached to the Ethernet 1/0/1 interface in the outbound direction.

#### **Example 12-5** Configuring CBWFQ and Attaching a Policy to an Ethernet Interface

```
Router(config)# class-map match-any voip
Router(config-cmap)# match ip precedence 5
Router(config-cmap)# class-map match-any video
Router(config-cmap)# match ip precedence 4
Router(config-cmap)# exit
Router(config)# policy-map policy1
Router(config-pmap)# class voip
Router(config-pmap-c)# priority
Router(config-pmap-c)# police 560000
Router(config-pmap-c)# class video
Router(config-pmap-c)# bandwidth 4560
Router(config-pmap-c)# class class-default
Router(config-pmap-c) bandwidth 2560
Router(config-pmap-c)# exit
Router(config-pmap)# exit
Router(config)# interface ethernet1/0/1
Router(config-if)# service-policy output policy1
Router(config-if)# exit
```

## Configuration Example for Configuring CBWFQ and Attaching a Policy to an ATM Subinterface

[Example 12-6](#) shows how to configure CBWFQ in a policy map and attach the policy to an ATM subinterface. This example creates a class map named *voice* to classify traffic and assigns class characteristics in the policy map named *policy1*. The service policy is applied to the ATM subinterface 7/0/0.1, which uses PPPoA encapsulation.

### Example 12-6 Configuring CBWFQ and Attaching a Policy to a PPPoA Subinterface

```
Router(config)# class-map voice
Router(config-cmap)# match any
Router(config-cmap)# exit
Router(config)# policy-map policy1
Router(config-pmap)# class voice
Router(config-pmap-c)# police 120000 16000 32000 conform-action transmit exceed-action
set-prec-transmit 4
Router(config-pmap-c)# exit
Router(config-pmap)# exit
Router(config)# interface atm7/0/0.1 point-to-point
Router(config-subif)# pvc 10/32
Router(config-subif-atm-pvc)# encapsulation aal5mux ppp virtual-template 1
Router(config-subif-atm-pvc)# service-policy output policy1
Router(config-subif)# interface virtual-template 1
Router(config-if)# ip unnumbered loopback1
Router(config-if)# peer default ip address pool pool1
Router(config-if)# ppp authentication chap
```

## Configuration Example for Configuring CBWFQ and Attaching a Policy to an RBE Subinterface

[Example 12-7](#) shows how to configure CBWFQ in a policy map and attach the policy to an routed bridge encapsulation (RBE) subinterface. The example creates a class map named *voice* and a policy map named *map1*. The service policy is applied to the ATM subinterface 7/0/1.1, which uses ATM RBE.

### Example 12-7 Configuring CBWFQ and Attaching a Policy to an RBE Subinterface

```
Router(config)# class-map voice
Router(config-cmap)# match all
Router(config-cmap)# exit
Router(config)# policy-map map1
Router(config-pmap)# class voice
Router(config-pmap-c)# police 120000 16000 32000 conform-action transmit exceed-action set
precedence-transmit 4
Router(config-pmap-c)# exit
Router(config-pmap)# exit
Router(config-if)# interface atm7/0/1.1 point-to-point
Router(config-subif)# ip unnumbered loopback1
Router(config-subif)# atm route-bridge ip
Router(config-subif)# service-policy output map1
Router(config-subif)# range pvc 101/32 101/2031
Router(config-subif-atm-pvc)# encapsulation aal5snap
```

# Verifying and Monitoring Class-Based Weighted Fair Queuing

To verify the configuration of service policy maps and the classes associated with them, enter the following commands in privileged EXEC mode:

Command	Purpose
Router# <b>show policy-map</b>	Displays the configuration of all policy maps configured on the router.
Router# <b>show policy-map</b> <i>policy-map-name</i>	Displays the configuration of all classes contained in the policy map you specify.  <i>policy-map-name</i> is the name of the policy map.
Router# <b>show policy-map</b> <i>policy-map-name</i> <b>class</b> <i>class-map-name</i>	Displays the configuration of the class you specify. The policy map you specify includes this class.  <i>policy-map-name</i> is the name of the policy map that contains the traffic class for which you want to display the configuration.  <i>class-map-name</i> is the name of the class map that defines the traffic class.
Router# <b>show policy-map interface</b> <i>interface</i>	Displays the configuration of all classes configured for all policy maps attached to the interface you specify.  <i>interface</i> is the type and number of the interface.  <b>Note</b> After you enter the <b>show policy-map interface</b> command, the counters that display update only if congestion is present on the interface.
Router# <b>show queue</b> <i>interface</i>	Displays queuing configuration information and statistics for the interface you specify.  <i>interface</i> is the type and number of the interface.

## Related Documentation

This section provides hyperlinks to additional Cisco documentation for the features discussed in this chapter. To display the documentation, click the document title or a section of the document highlighted in blue. When appropriate, paths to applicable sections are listed below the documentation title.

Feature	Related Documentation
Weighted fair queuing (WFQ)	<a href="#">Cisco IOS Quality of Service Solutions Configuration Guide, Release 12.3</a> Part 2: Congestion Management > Congestion Management Overview > Weighted Fair Queuing
Class-based weighted fair queuing (CBWFQ)	<a href="#">Cisco IOS Quality of Service Solutions Configuration Guide, Release 12.3</a> Part 2: Congestion Management > Congestion Management Overview > Class-Based Weighted Fair Queuing

Feature	Related Documentation
Class maps	<i>Cisco IOS Quality of Service Solutions Configuration Guide, Release 12.2</i>  Part 8: Modular Quality of Service Command Line Interface > Configuring the Modular Quality of Service Command Line Interface > Modular QoS CLI Configuration Task List > Creating a Traffic Class
Policy maps	<i>Cisco IOS Quality of Service Solutions Configuration Guide, Release 12.2</i>  Part 8: Modular Quality of Service Command Line Interface > Configuring the Modular Quality of Service Command Line Interface > Modular QoS CLI Configuration Task List > Creating a Traffic Policy

