



# Class-Based Weighted Fair Queueing and Weighted Random Early Detection

---

The Class-Based Weighted Fair Queueing (CBWFQ) and Weighted Random Early Detection (WRED) features bring CBWFQ and WRED to both Versatile Interface Processor (VIP) and non-VIP platforms.

**Note:**

---

These features were previously called by the following names:

- *Class-Based Weighted Fair Queueing and Weighted Random Early Detection for the VIP Using the Modular QoS CLI*
- *Distributed Class-Based Weighted Fair Queueing and Distributed Weighted Random Early Detection*

These features may be referred to by those names in some Cisco documentation.

---

## Feature Specifications for the Class-Based Weighted Fair Queueing and Weighted Random Early Detection

Release	Modification
12.1(5)T	This feature was introduced.
12.0(26)S	<p>This feature was integrated into Cisco IOS Release 12.0(26)S for the Cisco 7200 and 7500 series routers.</p> <p>The name of the feature changed from <i>Distributed Class-Based Weighted Fair Queueing and Distributed Weighted Random Early Detection</i> to <i>Class-Based Weighted Fair Queueing and Weighted Random Early Detection</i>.</p> <p>The <b>fair-queue (policy-map class)</b> command was enhanced to enable fair queuing in every class (not just the default class) for VIP and non-VIP platforms.</p> <p>The hierarchical queuing feature was added for the Cisco 7200 series routers, which allows you to enable queuing in child policies without enabling shaping in parent policies.</p>

---

### 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.

## Contents

- [Feature Overview, page 2](#)
- [Benefits, page 4](#)
- [Prerequisites, page 6](#)
- [Configuration Tasks, page 7](#)
- [Additional References, page 13](#)
- [Additional References, page 13](#)

## Feature Overview

CBWFQ extends the standard WFQ functionality to provide support for user-defined traffic classes on both VIP and non-VIP platforms. These user-defined traffic classes are configured in the Modular Quality of Service Command-Line Interface (Modular QoS CLI). For information on configuring QoS with the Modular QoS CLI, see the *Modular Quality of Service Command-Line Interface* document on CCO and the Documentation CD-ROM.

The maximum number of packets allowed to accumulate in a traffic class queue is called the queue limit and is specified with the **queue-limit** command when you create a service policy with the **policy-map** command. Packets belonging to a traffic class are subject to the guaranteed bandwidth allocation and the queue limits that characterize the traffic class.

After a queue has reached its configured queue limit, enqueueing of additional packets to the traffic class causes tail drop or WRED drop to take effect, depending on how the service policy is configured. (Tail drop is a means of avoiding congestion that treats all traffic equally and does not differentiate between classes of service. Queues fill during periods of congestion. When the output queue is full and tail drop is in effect, packets are dropped until the congestion is eliminated and the queue is no longer full).

Tail drop is used for CBWFQ traffic classes unless you explicitly configure a service policy to use WRED to drop packets as a means of avoiding congestion. Note that if you use WRED packet drop instead of tail drop for one or more traffic classes making up a service policy, you must ensure that WRED is not configured for the interface to which you attach that service policy.

WRED drops packets selectively based on IP precedence. Packets with a higher IP precedence are less likely to be dropped than packets with a lower precedence. Thus, higher-priority traffic is delivered with a higher probability than lower priority traffic in the default scenario. However, packets with a lower IP precedence are less likely to be dropped than packets with a higher IP precedence in certain WRED configurations. You can also configure WRED to ignore IP precedence when making drop decisions, so that non-weighted RED behavior is achieved.

WRED is useful on any output interface where you expect to have congestion. However, WRED is usually used in the core routers of a network, rather than the edge. Edge routers assign IP precedences to packets as they enter the network. WRED uses these precedences to determine how it treats different types of traffic.

## Resource Reservation Protocol Interaction with CBWFQ

When Resource Reservation Protocol (RSVP) and CBWFQ are configured, RSVP and CBWFQ act independently of one another. RSVP and CBWFQ allocate bandwidth among their traffic classes and flows according to unallocated bandwidth available at the underlying point of congestion.

## WRED Functional Description

When a packet arrives, the following events occur:

- The average queue size is calculated. See the [“Average Queue Size”](#) section for details.
- If the average is less than the minimum queue threshold, the arriving packet is queued.
- If the average is between the minimum queue threshold and the maximum queue threshold, the packet is either dropped or queued, depending on the packet drop probability. See the [“Packet-Drop Probability”](#) section for details.
- If the average queue size is greater than the maximum queue threshold, the packet is automatically dropped.

## Average Queue Size

The average queue size is based on the previous average and the current size of the queue. The formula is as follows, where  $n$  is the exponential weight factor, a user-configurable value:

$$\text{average} = (\text{old\_average} * (1 - 1/2^n)) + (\text{current\_queue\_size} * 1/2^n)$$



**Note**

---

Cisco recommends using the default value for the exponential weight factor. Change this value from the default value only if you have determined that your situation would benefit from using a different value.

---

For high values of  $n$ , the previous average queue size becomes more important. A large factor smooths out the peaks and lows in queue length. The average queue size is unlikely to change very quickly, avoiding drastic swings in size. The WRED process is slow to start dropping packets, but it may continue dropping packets for a time after the actual queue size has fallen below the minimum threshold. The slow-moving average accommodates temporary bursts in traffic.

If the value of  $n$  becomes too high, WRED does not react to congestion. Packets are transmitted or dropped as if WRED were not in effect.

For low values of  $n$ , the average queue size closely tracks the current queue size. The resulting average may fluctuate with changes in the traffic levels. In this case, the WRED process responds quickly to long queues. Once the queue falls below the minimum threshold, the process stops dropping packets.

If the value of  $n$  becomes too low, WRED overreacts to temporary traffic bursts and drops traffic unnecessarily.

## Packet-Drop Probability

The probability that a packet will be dropped is based on the minimum threshold, maximum threshold, and mark probability denominator.

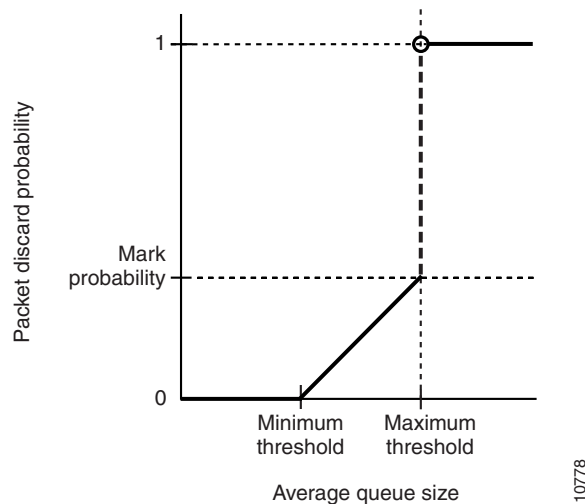
When the average queue size is above the minimum threshold, RED starts dropping packets. The rate of packet drop increases linearly as the average queue size increases, until the average queue size reaches the maximum threshold.

The mark probability denominator is the fraction of packets dropped when the average queue size is at the maximum threshold. For example, if the denominator is 512, one out of every 512 packets is dropped when the average queue is at the maximum threshold.

When the average queue size is above the maximum threshold, all packets are dropped.

Figure 1 summarizes the packet drop probability.

**Figure 1 WRED Packet Drop Probability**



The minimum threshold value should be set high enough to maximize the link utilization. If the minimum threshold is too low, packets may be dropped unnecessarily, and the transmission link will not be fully used.

The difference between the maximum threshold and the minimum threshold should be large enough to avoid global synchronization of TCP hosts (global synchronization of TCP hosts can occur as multiple TCP hosts reduce their transmission rates). If the difference between the maximum and minimum thresholds is too small, many packets may be dropped at once, resulting in global synchronization.

## Benefits

### Bandwidth Allocation

CBWFQ allows you to specify the amount of guaranteed bandwidth to be allocated for a traffic class. Taking into account available bandwidth on the interface, you can configure up to 64 traffic classes and control bandwidth allocation among them. If excess bandwidth is available, the excess bandwidth is divided amongst the traffic classes in proportion to their configured bandwidths.

Flow-based WFQ allocates bandwidth equally among all flows.

### Coarser Granularity and Scalability

CBWFQ allows you to define what constitutes a traffic class based on criteria that exceed the confines of flow. CBWFQ allows you to use access control lists and protocols or input interface names to define how traffic is classified, thereby providing coarser granularity. You need not maintain traffic classification on a flow basis. Moreover, you can configure up to 64 discrete traffic classes in a service policy.

### Consistent Traffic Flows

When WRED is not configured, output buffers fill during periods of congestion. When the buffers are full, tail drop occurs; all additional packets are dropped. Since the packets are dropped all at once, global synchronization of TCP hosts can occur as multiple TCP hosts reduce their transmission rates. The congestion clears, and the TCP hosts increase their transmission rates, resulting in waves of congestion followed by periods when the transmission link is not fully used.

WRED reduces the chances of tail drop by selectively dropping packets when the output interface begins to show signs of congestion. By dropping some packets early rather than waiting until the buffer is full, WRED avoids dropping large numbers of packets at once and minimizes the chances of global synchronization. Thus, RED allows the transmission line to be used fully at all times.

In addition, WRED statistically drops more packets from large users than small. Therefore, traffic sources that generate the most traffic are more likely to be slowed down than traffic sources that generate little traffic.

WRED provides separate thresholds and weights for different IP precedences, allowing you to provide different qualities of service for different traffic. Standard traffic may be dropped more frequently than premium traffic during periods of congestion.

## Restrictions

### Using the `bandwidth` Command on VIP Default Traffic Class

On a VIP, all traffic that does not match a user-defined traffic class is classified as part of the default traffic class. The implicit bandwidth allocated to the default traffic class is equal to the link bandwidth minus all of the user-defined bandwidth given to the user-defined traffic classes (with the `bandwidth` command). At least 1 percent of the link bandwidth is always reserved for the default traffic class.

Because the bandwidth of the default traffic class is implicit (the default traffic class receives all remaining bandwidth not given to the user-defined traffic classes), the `bandwidth` command cannot be used with the default traffic class when you configure a VIP.

### Using the `match protocol` Command on a VIP

With a VIP, do not use the `match protocol` command to create a traffic class with a non-IP protocol as a match criterion. The VIP does not support matching of non-IP protocols.

### WRED Restrictions

WRED has the following restrictions:

- WRED is useful only when the bulk of the traffic is TCP/IP traffic. With TCP, dropped packets indicate congestion, so the packet source reduces its transmission rate. With other protocols, packet sources may not respond or may resend dropped packets at the same rate. Thus, dropping packets does not necessarily decrease congestion.
- WRED treats non-IP traffic as precedence 0, the lowest precedence. Therefore, non-IP traffic is usually more likely to be dropped than IP traffic.

- You cannot configure WRED on the same interface as RSP-based custom queuing, priority queuing, or weighted fair queuing (WFQ). However, you can configure both WRED and DWFQ on the same interface.

## Prerequisites

### Weighted Fair Queueing

Attaching a service policy to an interface disables WFQ on that interface if WFQ is configured for the interface. For this reason, you should ensure that WFQ is not enabled on such an interface.

For information on WFQ, see the “Configuring Weighted Fair Queueing” chapter of the Cisco IOS Release 12.2 *Quality of Service Solutions Configuration Guide*.

### WRED

Attaching a service policy configured to use WRED to an interface disables WRED on that interface. If any of the traffic classes that you configure in a policy map use WRED for packet drop instead of tail drop, you must ensure that WRED is not configured on the interface to which you intend to attach that service policy.

### Access Control Lists

You can specify a numbered access list as the match criterion for any traffic class that you create. For this reason, you should know how to configure access lists.

### Cisco Express Forwarding

In order to use WRED, Cisco Express Forwarding switching must be enabled on the interface. Refer to the Cisco Express Forwarding feature documentation for configuration information.

### Modular Quality of Service Command-Line Interface

You can configure CBWFQ and WRED using the Modular Quality of Service Command-Line Interface.

For information on configuring QoS features with the Modular QoS CLI, see the *Modular Quality of Service Command-Line Interface* document on CCO and the Documentation CD-ROM.

## Configuration Tasks

CBWFQ is configured using user-defined traffic classes and service policies. Traffic classes and service policies are configured in the Modular QoS CLI. For information on configuring the Modular QoS CLI, see the *Modular Quality of Service Command Line Interface* document on CCO and the Documentation CD-ROM.

This section contains the following information:

- [Configuring a Service Policy in the Policy Map](#)
- [Configuring a Service Policy with WRED in the Policy Map](#)
- [Configuring a Service Policy Including IP Precedence Using WRED](#)
- [Modifying the Bandwidth for an Existing Service Policy](#)
- [Modifying the Queue Limit for an Existing Service Policy](#)

### Configuring a Service Policy in the Policy Map

CBWFQ is configured using the Modular QoS CLI. In the Modular QoS CLI, a traffic class must be created before you can configure a service policy. The following example assumes that a traffic class has already been created. For information on configuring the Modular QoS CLI, including information on configuring traffic classes, see the *Modular Quality of Service Command-Line Interface* document on CCO and the Documentation CD-ROM.

In the following example, a service policy (specified with the **policy-map** command) containing a bandwidth and queue-limit specification is created.

To configure Quality of Service features in a service policy, use the **policy-map** command in global configuration mode and specify the service policy name and then use the following service policy configuration commands to configure QoS policies for a specified traffic class. For each traffic class that you define, you can use one or more of the following service policy configuration commands to configure the service policy. For example, you might specify bandwidth for one traffic class and both bandwidth and queue limit for another traffic class.

	Command	Purpose
Step 1	Router(config)# <b>policy-map</b> <i>policy-map</i>	Specifies the name of the service policy to be created or modified.
Step 2	Router(config-pmap)# <b>class</b> <i>class-name</i>	Specifies the name of the traffic class to be associated with the service policy.
Step 3	Router(config-pmap-c)# <b>bandwidth</b> <i>bandwidth-kbps</i>	Specifies the amount of bandwidth in kilobits per second (kbps) to be assigned to packets that meet the match criteria of the associated traffic class.
Step 4	Router(config-pmap-c)# <b>queue-limit</b> <i>number-of-packets</i>	Specifies the maximum number of packets that can be enqueued that meet the matching criteria of the associated traffic class.

After configuring the service policy with the **policy-map** command, you must still attach the service policy to an interface before it is successfully enabled. For information on attaching a service policy to an interface, see the *Modular Quality of Service Command-Line Interface* document on CCO and the Documentation CD-ROM.

## Configuring a Service Policy with WRED in the Policy Map

To configure a service policy and create traffic classes (including a default traffic class) that make up the service policy, use the **policy-map** command in global configuration mode to specify the service policy name. Then use the following commands in policy map configuration mode to configure the service policy for a traffic class. The service policy's default traffic class is the traffic class to which traffic is directed if that traffic does not satisfy the match criteria of other traffic classes whose policy is defined in the service policy. To configure policy for more than one traffic class in the same policy map, repeat Step 2 through Step 4.

To attach a service policy to an interface and enable CBWFQ on the interface, you must create a policy map. You can configure traffic class policies for as many traffic classes as are defined on the router, up to the maximum of 64.

	Command	Purpose
Step 1	Router(config)# <b>policy-map</b> <i>policy-map</i>	Specifies the name of the service policy to be created or modified.
Step 2	Router(config-pmap)# <b>class</b> <i>class-name</i>	Specifies the name of a traffic class to be created and included in the service policy.
Step 3	Router(config-pmap-c)# <b>bandwidth</b> <i>bandwidth-kbps</i>	Specifies the amount of bandwidth in kilobits per second (kbps) to be assigned to the traffic class.
Step 4	Router(config-pmap-c)# <b>random-detect</b> <b>exponential-weighting-constant</b> <i>exponent</i>	Configures the exponential weight factor used in calculating the average queue length.
Step 5	Router(config-pmap-c)# <b>fair-queue</b> [ <b>queue-limit</b> <i>queue-values</i> ]	Specifies the number of queues to be reserved for the traffic class.
Step 6	Router(config-pmap-c)# <b>queue-limit</b> <i>number-of-packets</i>	Specifies the maximum number of packets that can be enqueued for the specified traffic class.

After configuring the service policy with the **policy-map** command, you must still attach the service policy to an interface before it is successfully enabled. For information on attaching a service policy to an interface, see the *Modular Quality of Service Command-Line Interface* document on CCO and the Documentation CD-ROM.

## Configuring a Service Policy Including IP Precedence Using WRED

To configure a service policy using WRED to specify IP precedence, use the following commands beginning in global configuration mode:

	Command	Purpose
Step 1	Router(config)# <b>policy-map</b> <i>policy-map</i>	Specifies the name of the service policy to be created or modified.
Step 2	Router(config-pmap)# <b>class</b> <i>class-name</i>	Specifies the name of a traffic class to associate with the service policy.
Step 3	Router(config-pmap-c)# <b>bandwidth</b> <i>bandwidth-kbps</i>	Specifies the amount of bandwidth in kbps to be assigned to the traffic class.

	Command	Purpose
Step 4	Router(config-pmap-c) # <b>random-detect exponential-weighting-constant</b> <i>exponent</i>	Configures the exponential weight factor used in calculating the average queue length.
Step 5	Router(config-pmap-c) # <b>random-detect precedence</b> <i>precedence min-threshold max-threshold mark-prob-denominator</i>	Configures the parameters for packets with a specific IP precedence. The minimum threshold for IP precedence 0 corresponds to half the maximum threshold for the interface. Repeat this command for each precedence.

After configuring the service policy with the **policy-map** command, you must still attach the service policy to an interface before it is successfully enabled. For information on attaching a service policy to an interface, see the *Modular Quality of Service Command-Line Interface* document on CCO and the Documentation CD-ROM.

## Modifying the Bandwidth for an Existing Service Policy

To change the amount of bandwidth allocated for an existing traffic class, use the following commands.

	Command	Purpose
Step 1	Router(config) # <b>policy-map</b> <i>policy-map</i>	Specifies the name of the service policy to be created or modified.
Step 2	Router(config-pmap) # <b>class</b> <i>class-name</i>	Specifies the name of a traffic class whose bandwidth you want to modify.
Step 3	Router(config-pmap-c) # <b>bandwidth</b> <i>bandwidth-kbps</i>	Specifies the new amount of bandwidth in kbps per second to be used to reconfigure the traffic class.

After configuring the service policy with the **policy-map** command, you must still attach the service policy to an interface before it is successfully enabled. For information on attaching a service policy to an interface, see the *Modular Quality of Service Command-Line Interface* document on CCO and the Documentation CD-ROM.

## Modifying the Queue Limit for an Existing Service Policy

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

	Command	Purpose
Step 1	Router(config) # <b>policy-map</b> <i>policy-map</i>	Specifies the name of the service policy to be created or modified.
Step 2	Router(config-pmap) # <b>class</b> <i>class-name</i>	Specifies the name of a traffic class whose queue limit you want to modify.
Step 3	Router(config-pmap-c) # <b>queue-limit</b> <i>number-of-packets</i>	Specifies the new maximum number of packets that can be enqueued for the traffic class to be reconfigured. The default and maximum number of packets is 64.

After configuring the service policy with the **policy-map** command, you must still attach the service policy to an interface before it is successfully enabled. For information on attaching a service policy to an interface, see the *Modular Quality of Service Command-Line Interface* document on CCO and the Documentation CD-ROM.

## Monitoring and Maintaining CBWFQ and WRED

Use the **show policy-map** [interface [interface-spec [input | output [class class-name]]]] command to display the configuration of a service policy and its associated traffic classes. Forms of this command are listed in the table below. See the *Modular Quality of Service Command-Line Interface* document for more information about the **show policy-map** command.

Command	Purpose
Router# <b>show policy-map</b>	Displays all configured service policies.
Router# <b>show policy-map</b> policy-map-name	Displays the user-specified service policy.
Router# <b>show policy-map interface</b>	Displays statistics and configurations of all input and output policies, which are attached to an interface.
Router# <b>show policy-map interface</b> interface-spec	Displays configuration and statistics of the input and output policies attached to a particular interface.
Router# <b>show policy-map interface</b> interface-spec input	Displays configuration and statistics of the input policy attached to an interface.
Router# <b>show policy-map interface</b> interface-spec output	Displays configuration statistics of the output policy attached to an interface.
Router# <b>show policy-map</b> [interface [interface-spec [input   output] [ class class-name]]]	Displays the configuration and statistics for the class name configured in the policy.

## Configuration Examples

This section provides the following configuration examples:

- [Configuring a Traffic Class](#)
- [Creating a Service Policy](#)
- [Attaching a Service Policy to an Interface](#)
- [Enabling Hierarchical Queuing](#)

### Configuring a Traffic Class

In the following example, two traffic classes are created and their match criteria are defined. For the first traffic class, called class1, the numbered ACL 101 is used as the match criterion. For the second traffic class, called class2, the access control list ACL 102 is used as the match criterion. Packets are checked against the contents of these ACLs to determine if they belong to the traffic class.

```
Router(config)# class-map class1
Router(config-cmap)# match access-group 101
Router(config-cmap)# exit
```

```
Router(config)# class-map class2
Router(config-cmap)# match access-group 102
Router(config-cmap)# exit
```

Additional information regarding traffic classes is contained in the *Modular Quality of Service Command-Line Interface* document on CCO and the Documentation CD-ROM.

## Creating a Service Policy

In the following example, a service policy called policy1 is defined to associate Quality of Service (QoS) features with the two traffic classes—class1 and class2. The match criteria for these traffic classes were defined in the previous “[Configuring a Traffic Class](#)” section.

For class1, the QoS policies include bandwidth allocation request and maximum packet count limit for the queue reserved for the traffic class. For class2, the policy specifies only a bandwidth allocation request, so the default queue limit of 64 packets is assumed.

```
Router(config)# policy-map policy1

Router(config-pmap)# class class1
Router(config-pmap-c)# bandwidth 3000
Router(config-pmap-c)# queue-limit 30
Router(config-pmap)# exit

Router(config-pmap)# class class2
Router(config-pmap-c)# bandwidth 2000
Router(config-pmap)# exit
```

Additional information regarding service policy configurations is available in the *Modular Quality of Service Command-Line Interface* document on CCO and the Documentation CD-ROM.

## Attaching a Service Policy to an Interface

The following example shows how to attach an existing service policy to an interface. After you define a service policy, you can attach it to one or more interfaces to specify a service policy for those interfaces. Although you can assign the same service policy to multiple interfaces, each interface can have only one service policy attached at the input and one policy map attached at the output at one time.

```
Router(config)# interface fe1/0/0
Router(config-if)# service output policy1
Router(config-if)# exit
```

Additional information regarding attaching service policy configurations to interfaces is available in the *Modular Quality of Service Command-Line Interface* document on CCO and the Documentation CD-ROM.

## Enabling Hierarchical Queuing

You can nest a traffic policy within another policy by using the **service-policy** command in policy-map class configuration mode. A traffic policy that contains a nested traffic policy is called a hierarchical traffic policy.

A hierarchical traffic policy contains the following components:

- A child policy: A previously defined traffic policy that is being associated with the new traffic policy through the use of the **service-policy** command
- A parent policy: A new traffic policy that includes the defined traffic policy.

Cisco IOS Release 12.0(26)S introduces support for QOS queuing policies (both flat and hierarchical) on the Cisco 7200 series routers. Hierarchical queuing policies are supported on the Cisco 7200 series routers in the Cisco IOS 12.2 and 12.3 releases. However, those queuing policies require shaping in the parent policy. For an example of a queuing policy that has shaping in the parent policy and bandwidth in the child policy, see the feature module *Class-Based Shaping*.

In Cisco IOS Release 12.0(26)S, hierarchical queuing policies do not require shaping in the parent policy. You can specify queuing in the parent policy and in the child policy. The following example illustrates a traffic policy that specifies the **bandwidth (policy-map class)** command at both the child- and parent-level policies.

The policy named parent is attached to a link of bandwidth 100 kbps. The policy named parent includes two classes:

The class called customer1 has the following characteristics:

- Customer1 is ensured of minimum bandwidth guarantee of 70 kbps.
- The UDP traffic of customer1 is ensured a minimum bandwidth of 20 kbps
- The TCP traffic of customer1 is ensured a minimum bandwidth of 40 kbps.
- The remaining 10 kbps bandwidth for customer1 is divided proportionately between TCP and UDP and other unclassified traffic.

The class called customer2 has the following characteristics:

- Customer2 is ensured a minimum bandwidth guarantee of 30 kbps.
- The UDP traffic of customer2 is ensured a minimum bandwidth of 10 kbps.
- The TCP traffic of customer2 is ensured a minimum bandwidth of 20 kbps.

```

policy child_customer1
  class udp_customer1
    bandwidth 20
  class tcp_customer1
    bandwidth 40

policy child_customer2
  class udp_customer2
    bandwidth 10
  class tcp_customer2
    bandwidth 20

policy parent
  class customer1
    bandwidth 70
    service-policy child_customer1
  class customer2
    bandwidth 30
    service-policy child_customer2

```

## Additional References

The following sections provide references related to Class-Based Weighted Fair Queueing and Weighted Random Early Detection.

### Related Documents

Related Topic	Document Title
Quality of Service	<i>Cisco IOS Quality of Service Solutions Command Reference, Release 12.3</i> <i>Cisco IOS Quality of Service Solutions Configuration Guide, Release 12.3</i>
Modular Quality of Service	<i>Modular Quality of Service Command Line Interface</i>

### Standards

Standards	Title
None	—

### MIBs

MIBs	MIBs Link
None	To locate and download MIBs for selected platforms, Cisco IOS releases, and feature sets, use Cisco MIB Locator found at the following URL: <a href="http://www.cisco.com/go/mibs">http://www.cisco.com/go/mibs</a>

### RFCs

RFCs	Title
None	—

### Technical Assistance

Description	Link
Technical Assistance Center (TAC) home page, containing 30,000 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.	<a href="http://www.cisco.com/public/support/tac/home.shtml">http://www.cisco.com/public/support/tac/home.shtml</a>

# Command Reference

This section documents new or modified commands. All other commands used with this feature are documented in the Cisco IOS Release 12.3 command references.

- [bandwidth \(policy-map class\)](#)
- [fair-queue \(policy-map class\)](#)
- [queue-limit](#)
- [random-detect exponential-weighting-constant](#)

## bandwidth (policy-map class)

To specify or modify the bandwidth allocated for a class belonging to a policy map, use the **bandwidth** command in policy-map class configuration mode. To remove the bandwidth specified for a class, use the **no** form of this command.

**bandwidth** { *bandwidth-kbps* | **remaining percent** *percentage* | **percent** *percentage* }

**no bandwidth** { *bandwidth-kbps* | **remaining percent** *percentage* | **percent** *percentage* }

### Syntax Description

<i>bandwidth-kbps</i>	Amount of bandwidth, in number of kbps, to be assigned to the class. The amount of bandwidth varies according to the interface and platform in use.
<b>remaining percent</b>	Amount of guaranteed bandwidth, based on a relative percent of available bandwidth.
<i>percentage</i>	Used in conjunction with the <b>remaining percent</b> keyword, a percentage. The percentage can be a number from 1 to 100.
<b>percent</b>	Amount of guaranteed bandwidth, based on an absolute percent of available bandwidth.
<i>percentage</i>	Used in conjunction with the <b>percent</b> keyword, the percentage of the total available bandwidth to be set aside for the priority class. The percentage can be a number from 1 to 100.

### Defaults

No bandwidth is specified

### Command Modes

Policy-map class configuration

### Command History

Release	Modification
12.0(5)T	This command was introduced.
12.0(5)XE	This command was incorporated into Cisco IOS Release 12.0(5)XE and implemented on Versatile Interface Processor (VIP)-enabled Cisco 7500 series routers.
12.0(7)T	The <b>percent</b> keyword was added.
12.1(5)T	This command was integrated into Cisco IOS Release 12.1(5)T and implemented on VIP-enabled Cisco 7500 series routers.
12.2(2)T	The <b>remaining percent</b> keyword was added.
12.0(26)S	This command was integrated into Cisco IOS Release 12.0(26)S.

### Usage Guidelines

You should use the **bandwidth** command when you configure a policy map for a class defined by the **class-map** command. The **bandwidth** command specifies the bandwidth for traffic in that class. Class-based weighted fair queuing (CBWFQ) derives the weight for packets belonging to the class from the bandwidth allocated to the class. CBWFQ then uses the weight to ensure that the queue for the class is serviced fairly.

### Specifying Bandwidth as a Percentage

Besides specifying the amount of bandwidth in kbps, you can specify bandwidth as a percentage of either the available bandwidth or the total bandwidth. During periods of congestion, the classes are serviced in proportion to their configured bandwidth percentages. Available bandwidth is equal to the interface bandwidth minus the sum of all bandwidths reserved by the Resource Reservation Protocol (RSVP) feature, the IP RTP Priority feature, and the Low Latency Queuing (LLQ) feature.



#### Note

It is important to remember that when the **bandwidth remaining percent** command is configured, hard bandwidth guarantees may not be provided and only relative bandwidths are assured. That is, class bandwidths are always proportional to the specified percentages of the interface bandwidth. When the link bandwidth is fixed, class bandwidth guarantees are in proportion to the configured percentages. If the link bandwidth is unknown or variable, class bandwidth guarantees in kbps cannot be computed.

### Bandwidth Command Restrictions

The following restrictions apply to the **bandwidth** command:

- The amount of bandwidth configured should be large enough to also accommodate Layer 2 overhead.
- A policy map can have all the class bandwidths specified in kbps or all the class bandwidths specified in percentages but not a mix of both in the same class. However, the unit for the **priority** command in the priority class can be different from the bandwidth unit of the nonpriority class.
- When the **bandwidth percent** command is configured, and a policy map containing class policy configurations is attached to the interface to stipulate the service policy for that interface, available bandwidth is assessed. If a policy map cannot be attached to a particular interface because of insufficient interface bandwidth, the policy is removed from all interfaces to which it was successfully attached. This restriction does not apply to the **bandwidth remaining percent** command.

For more information on bandwidth allocation, refer to the chapter “Congestion Management Overview” in the *Cisco IOS Quality of Service Solutions Configuration Guide*.

Note that when the policy map containing class policy configurations is attached to the interface to stipulate the service policy for that interface, available bandwidth is assessed. If a policy map cannot be attached to a particular interface because of insufficient interface bandwidth, then the policy is removed from all interfaces to which it was successfully attached.

### Queue Limits

The **bandwidth** command can be used with the Modular Command-Line Interface (MQC) to specify the bandwidth for a particular class. When used with the MQC, the **bandwidth** command uses a default queue limit for the class. This queue limit can be modified using the **queue-limit** command, thereby overriding the default set by the **bandwidth** command.



#### Note

Using the **queue-limit** command to modify the default queue-limit is especially important for higher-speed interfaces, in order to meet the minimum bandwidth guarantees required by the interface.

**Examples****CBWFQ Bandwidth Guarantee Example**

The following example shows how bandwidth is guaranteed when only CBWFQ is configured:

```
! The following commands create a policy map with two classes:
policy-map policy1
  class class1
    bandwidth percent 50
  exit

  class class2
    bandwidth percent 25
  exit
end

!The following commands attach the policy to interface serial3/2:
interface serial3/2
  service output policy1
end
```

The following output from the **show policy-map** command shows the configuration for the policy map called policy1:

```
Router# show policy-map policy1

Policy Map policy1
  Class class1
    Weighted Fair Queueing
    Bandwidth 50 (%) Max Threshold 64 (packets)
  Class class2
    Weighted Fair Queueing
    Bandwidth 25 (%) Max Threshold 64 (packets)
```

The output from the **show policy-map interface** command shows that 50 percent of the interface bandwidth is guaranteed for the class called class1, and 25 percent is guaranteed for the class called class2. The output displays the amount of bandwidth as both a percentage and a number of kbps.

```
Router# show policy-map interface serial3/2

Serial3/2

Service-policy output:policy1

Class-map:class1 (match-all)
  0 packets, 0 bytes
  5 minute offered rate 0 bps, drop rate 0 bps
  Match:none
  Weighted Fair Queueing
    Output Queue:Conversation 265
    Bandwidth 50 (%)
    Bandwidth 772 (kbps) Max Threshold 64 (packets)
    (pkts matched/bytes matched) 0/0
    (depth/total drops/no-buffer drops) 0/0/0

Class-map:class2 (match-all)
  0 packets, 0 bytes
  5 minute offered rate 0 bps, drop rate 0 bps
  Match:none
  Weighted Fair Queueing
    Output Queue:Conversation 266
    Bandwidth 25 (%)
    Bandwidth 386 (kbps) Max Threshold 64 (packets)
    (pkts matched/bytes matched) 0/0
    (depth/total drops/no-buffer drops) 0/0/0
```

```
Class-map:class-default (match-any)
  0 packets, 0 bytes
  5 minute offered rate 0 bps, drop rate 0 bps
Match:any
```

In this example, interface serial3/2 has a total bandwidth of 1544 kbps. During periods of congestion, 50 percent (or 772 kbps) of the bandwidth is guaranteed to the class called class1, and 25 percent (or 386 kbps) of the link bandwidth is guaranteed to the class called class2.

### CBWFQ and LLQ Bandwidth Allocation Example

The following output from the **show policy-map** command shows the configuration for a policy map called p1:

```
Router# show policy-map p1

Policy Map p1
  Class voice
    Weighted Fair Queueing
    Strict Priority
    Bandwidth 500 (kbps) Burst 12500 (Bytes)
  Class class1
    Weighted Fair Queueing
    Bandwidth remaining 50 (%) Max Threshold 64 (packets)
  Class class2
    Weighted Fair Queueing
    Bandwidth remaining 25 (%) Max Threshold 64 (packets)
```

The following output from the **show policy-map interface** command on serial interface 3/2 shows that 500 kbps of bandwidth is guaranteed for the class called voice1. The classes called class1 and class2 receive 50 percent and 25 percent of the remaining bandwidth, respectively. Any unallocated bandwidth is divided proportionally among class1, class2, and any best-effort traffic classes.



#### Note

Note that in this sample output (unlike many of the others earlier in this section) the bandwidth is displayed only as a percentage. Bandwidth expressed as a number of kbps is not displayed because the **bandwidth remaining percent** keyword was used with the **bandwidth** command. The **bandwidth remaining percent** keyword allows you to allocate bandwidth as a relative percentage of the total bandwidth available on the interface.

```
Router# show policy-map interface serial3/2

Serial3/2

Service-policy output:p1

Class-map:voice (match-all)
  0 packets, 0 bytes
  5 minute offered rate 0 bps, drop rate 0 bps
Match:ip precedence 5
  Weighted Fair Queueing
    Strict Priority
    Output Queue:Conversation 264
    Bandwidth 500 (kbps) Burst 12500 (Bytes)
    (total drops/bytes drops) 0/0

Class-map:class1 (match-all)
  0 packets, 0 bytes
  5 minute offered rate 0 bps, drop rate 0 bps
Match:none
```

```

Weighted Fair Queueing
  Output Queue:Conversation 265
  Bandwidth remaining 50 (%) Max Threshold 64 (packets)
  (pkts matched/bytes matched) 0/0
  (depth/total drops/no-buffer drops) 0/0/0

Class-map:class2 (match-all)
  0 packets, 0 bytes
  5 minute offered rate 0 bps, drop rate 0 bps
  Match:none
  Weighted Fair Queueing
    Output Queue:Conversation 266
    Bandwidth remaining 25 (%) Max Threshold 64 (packets)
    (pkts matched/bytes matched) 0/0
    (depth/total drops/no-buffer drops) 0/0/0

Class-map:class-default (match-any)
  0 packets, 0 bytes
  5 minute offered rate 0 bps, drop rate 0 bps
  Match:any

```

**Related Commands**

<b>Command</b>	<b>Description</b>
<b>class (policy-map)</b>	Specifies the name of the class whose policy you want to create or change, and the default class (commonly known as the class-default class) before you configure its policy.
<b>class-map</b>	Creates a class map to be used for matching packets to a specified class.
<b>max-reserved-bandwidth</b>	Changes the percent of interface bandwidth allocated for CBWFQ, LLQ, and IP RTP Priority.
<b>policy-map</b>	Creates or modifies a policy map that can be attached to one or more interfaces to specify a service policy.
<b>queue-limit</b>	Specifies or modifies the maximum number of packets the queue can hold for a class policy configured in a policy map.
<b>random-detect (interface)</b>	Enables WRED or WRED.
<b>random-detect exponential-weighting-constant</b>	Configures the WRED exponential weight factor for the average queue size calculation.
<b>random-detect precedence</b>	Configures WRED parameters for a particular IP precedence.
<b>show policy-map</b>	Displays the configuration of all classes for a specified service policy map or all classes for all existing policy maps.
<b>show policy-map interface</b>	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.

## fair-queue (policy-map class)

To specify the number of queues to be reserved for use by a traffic class, use the **fair-queue** command in QoS policy-map class configuration mode. To delete the configured number of queues from the traffic class, use the **no** form of this command.

**fair-queue** [*dynamic-queues*]

**no fair-queue** [*dynamic-queues*]

<b>Syntax Description</b>	<i>dynamic-queues</i>	(Optional) A number specifying the number of dynamic conversation queues. The number can be in the range of 16 to 4096.
---------------------------	-----------------------	---

**Defaults** No queues are reserved.

**Command Modes** QoS policy-map class configuration

<b>Command History</b>	<b>Release</b>	<b>Modification</b>
	12.0(5)T	This command was introduced.
	12.0(5)XE	This command was integrated into Cisco IOS Release 12.0(5)XE and implemented on Versatile Interface Processor (VIP)-enabled Cisco 7500 series routers.
	12.1(5)T	This command was integrated into Cisco IOS Release 12.1(5)T and was implemented on VIP-enabled Cisco 7500 series routers.
	12.0(26)S	This command was integrated into Cisco IOS Release 12.0(26)S. The command was enhanced to support any traffic class on VIP and non-VIP platforms.

**Usage Guidelines** On VIP and non-VIP platforms, the **fair-queue** command can be used for any traffic class, not just the default traffic class. The **fair-queue** command can be used in conjunction with either the **queue-limit** command or the **random-detect exponential-weighting-constant** command.

**Examples** The following example configures the default traffic class for the policy map called policy9 to reserve ten queues for packets that do not satisfy match criteria specified for other traffic classes whose policy is configured in the same service policy. Because the **queue-limit** command is configured, tail drop is used for each queue when the maximum number of packets is enqueued and additional packets arrive.

```
policy-map policy9
  class class-default
    fair-queue 10
    queue-limit 20
```

The following example configures a service policy called policy8 that is associated with a user-defined traffic class called class1. The **fair-queue** command reserves 20 queues to be used for the service policy. For congestion avoidance, Weighted Random Early Detection (WRED) packet drop is used, not tail drop.

```
policy-map policy8
  class class1
  fair-queue 20
    random-detect exponential-weighting-constant 14
```

**Related Commands**

Command	Description
<b>class class-default</b>	Specifies the default traffic class for a service policy map.
<b>queue-limit</b>	Specifies or modifies the maximum number of packets the queue can hold for a class policy configured in a policy map.
<b>random-detect</b> <b>exponential-weighting-constant</b>	Configures the WRED exponential weight factor for the average queue size calculation.

# queue-limit

To specify or modify the maximum number of packets the queue can hold for a class policy configured in a policy map, use the **queue-limit** command in policy-map class configuration mode. To remove the queue packet limit from a class, use the **no** form of this command.

**queue-limit** *number-of-packets*

**no queue-limit** *number-of-packets*

## Syntax Description

<i>number-of-packets</i>	A number in the range from 1 to 64 specifying the maximum number of packets that the queue for this class can accumulate.
--------------------------	---

## Defaults

On the Versatile Interface Processor (VIP)-based platforms, the default value is chosen as a function of the bandwidth assigned to the traffic class. The default value is also based on available buffer memory. If sufficient buffer memory is available, the default **queue-limit** value is equal to the number of 250-byte packets that would lead to a latency of 500 milliseconds (ms) when the packets are delivered at the configured rate. For example, if two 250-byte packets are required to lead to a latency of 500 ms, the default *number-of-packets* value would be 2.

On all other platforms, the default is 64.

## Command Modes

Policy-map class configuration

## Command History

Release	Modification
12.0(5)T	This command was introduced.
12.0(5)XE	This command was integrated into Cisco IOS Release 12.0(5)XE. Support for VIP-enabled Cisco 7500 series routers was added.
12.1(5)T	This command was implemented on the VIP-enabled Cisco 7500 series routers.
12.0(26)S	This command was integrated into Cisco IOS Release 12.0(26)S

## Usage Guidelines

Weighted fair queuing (WFQ) creates a queue for every class for which a class map is defined. Packets satisfying the match criteria for a class accumulate in the queue reserved for the class until they are sent, which occurs when the queue is serviced by the fair queuing process. When the maximum packet threshold you defined for the class is reached, enqueueing of any further packets to the class queue causes tail drop or, if Weighted Random Early Detection (WRED) is configured for the class policy, packet drop to take effect.

### Overriding Queue Limits Set by the Bandwidth Command

The **bandwidth** command can be used with the Modular Command-Line Interface (MQC) to specify the bandwidth for a particular class. When used with the MQC, the **bandwidth** command uses a default queue limit for the class. This queue limit can be modified using the **queue-limit** command, thereby overriding the default set by the **bandwidth** command.

**Note**

Using the **queue-limit** command to modify the default queue-limit is especially important for higher-speed interfaces, in order to meet the minimum bandwidth guarantees required by the interface.

**Examples**

The following example configures a policy map called policy11 to contain policy for a class called acl203. Policy for this class is set so that the queue reserved for it has a maximum packet limit of 40.

```
policy-map policy11
  class acl203
    bandwidth 2000
    queue-limit 40
```

**Related Commands**

Command	Description
<b>class (policy-map)</b>	Specifies the name of the class whose policy you want to create or change, and the default class (commonly known as the class-default class) before you configure its policy.
<b>class class-default</b>	Specifies the default traffic class whose bandwidth is to be configured or modified.
<b>policy-map</b>	Creates or modifies a policy map that can be attached to one or more interfaces to specify a service policy.

## random-detect exponential-weighting-constant

To configure the Weighted Random Early Detection (WRED) exponential weight factor for the average queue size calculation for the queue, use the **random-detect exponential-weighting-constant** command in interface configuration mode. To configure the exponential weight factor for the average queue size calculation for the queue reserved for a class, use the **random-detect exponential-weighting-constant** command in policy-map class configuration mode. To return the value to the default, use the **no** form of this command.

**random-detect exponential-weighting-constant** *exponent*

**no random-detect exponential-weighting-constant**

<b>Syntax Description</b>	<i>exponent</i>	Exponent from 1 to 16 used in the average queue size calculation.
---------------------------	-----------------	---

<b>Defaults</b>	The default exponential weight factor is 9.
-----------------	---

<b>Command Modes</b>	Interface configuration when used on an interface Policy-map class configuration when used to specify class policy in a policy map, or when used in the Modular Quality of Service Command-Line Interface (MQC)
----------------------	--

<b>Command History</b>	<b>Release</b>	<b>Modification</b>
	11.1 CC	This command was introduced.
	12.0(5)T	This command was made available as a policy-map class configuration command.
	12.0(5)XE	This command was integrated into Cisco IOS Release 12.0(5)XE and implemented on VIP-enabled Cisco 7500 series routers.
	12.1(5)T	This command was integrated into Cisco IOS Release 12.1(5)T and implemented on VIP-enabled Cisco 7500 series routers.
	12.0(26)S	This command was integrated into Cisco IOS Release 12.0(26)S.

<b>Usage Guidelines</b>	WRED is a congestion avoidance mechanism that slows traffic by randomly dropping packets when congestion exists. WRED is similar to DWRED but uses the Versatile Interface Processor (VIP) instead of the Route Switch Processor (RSP). WRED and DWRED are most useful with protocols like TCP that respond to dropped packets by decreasing the transmission rate.
-------------------------	---

Use this command to change the exponent used in the average queue size calculation for the WRED services. You can also use this command to configure the exponential weight factor for the average queue size calculation for the queue reserved for a class.

**Note**

The default WRED or DWRED parameter values are based on the best available data. We recommend that you do not change the parameters from their default values unless you have determined that your applications would benefit from the changed values.

**Examples**

The following example configures WRED on an interface with a weight factor of 10:

```
interface Hssi0/0/0
  description 45Mbps to R1
  ip address 10.200.14.250 255.255.255.252
  random-detect
  random-detect exponential-weighting-constant 10
```

The following example configures the policy map called policy1 to contain policy specification for the class called class1. During times of congestion, WRED packet drop is used instead of tail drop. The weight factor used for the average queue size calculation for the queue for class1 is 12.

```
! The following commands create the class map called class1:
class-map class1
  match input-interface FE0/1
```

```
! The following commands define policy1 to contain policy specification for class1:
policy-map policy1
  class class1
    bandwidth 1000
    random-detect
    random-detect exponential-weighting-constant 12
```

The following example configures policy for a traffic class named int10 to configure the exponential weight factor as 12. This is the weight factor used for the average queue size calculation for the queue for traffic class int10. WRED packet drop is used for congestion avoidance for traffic class int10, not tail drop.

```
policy-map policy12
  class int10
    bandwidth 2000
    random-detect exponential-weighting-constant 12
```

Related Commands	Command	Description
	<b>bandwidth (policy-map class)</b>	Specifies or modifies the bandwidth allocated for a class belonging to a policy map.
	<b>exponential-weighting-constant</b>	Configures the exponential weight factor for the average queue size calculation for a WRED parameter group.
	<b>fair-queue (class-default)</b>	Specifies the number of dynamic queues to be reserved for use by the class-default class as part of the default class policy.
	<b>precedence</b>	Configures precedence levels for a VC or PVC class that can be assigned to a VC or PVC bundle and thus applied to all of the members of that bundle.
	<b>precedence (WRED group)</b>	Configures a WRED group for a particular IP Precedence.
	<b>random-detect dscp</b>	Changes the minimum and maximum packet thresholds for the DSCP value.
	<b>random-detect (per VC)</b>	Enables per-VC WRED or per-VC WRED.
	<b>random-detect precedence</b>	Configures WRED parameters for a particular IP Precedence.
	<b>show policy-map</b>	Displays the configuration of all classes for a specified service policy map or all classes for all existing policy maps.
	<b>show policy-map interface</b>	Displays the configuration of all classes configured for all service policies on the specified interface or displays the classes for the service policy for a specific PVC on the interface.
	<b>show queue</b>	Displays the contents of packets inside a queue for a particular interface or VC.
	<b>show queueing</b>	Lists all or selected configured queueing strategies.

CCIP, CCSP, the Cisco Arrow logo, the Cisco *Powered Network* mark, Cisco Unity, Follow Me Browsing, FormShare, and StackWise are trademarks of Cisco Systems, Inc.; Changing the Way We Work, Live, Play, and Learn, and iQuick Study are service marks of Cisco Systems, Inc.; and Aironet, ASIST, BPX, Catalyst, CCDA, CCDP, CCIE, CCNA, CCNP, Cisco, the Cisco Certified Internetwork Expert logo, Cisco IOS, the Cisco IOS logo, Cisco Press, Cisco Systems, Cisco Systems Capital, the Cisco Systems logo, Empowering the Internet Generation, Enterprise/Solver, EtherChannel, EtherSwitch, Fast Step, GigaStack, Internet Quotient, IOS, IP/TV, iQ Expertise, the iQ logo, iQ Net Readiness Scorecard, LightStream, MGX, MICA, the Networkers logo, Networking Academy, Network Registrar, *Packet*, PIX, Post-Routing, Pre-Routing, RateMUX, Registrar, ScriptShare, SlideCast, SMARTnet, StrataView Plus, Stratm, SwitchProbe, TeleRouter, The Fastest Way to Increase Your Internet Quotient, TransPath, and VCO are registered trademarks of Cisco Systems, Inc. and/or its affiliates in the U.S. and certain other countries.

All other trademarks mentioned in this document or Web site 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. (0304R)

■ random-detect exponential-weighting-constant