



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

The Class-Based Weighted Fair Queueing (CBWFQ) and Weighted Random Early Detection (WRED) for the Versatile Interface Processor (VIP) using the Modular Quality of Service Command-Line Interface (Modular QoS CLI) feature brings CBWFQ and WRED to the VIP.

This feature module includes the following sections:

- Feature Overview, page 1
- Benefits, page 4
- Supported Platforms, page 6
- Supported Standards, MIBs, and RFCs, page 6
- Prerequisites, page 6
- Configuration Tasks, page 7
- Command Reference, page 13

Feature Overview

Class-based weighted fair queueing (CBWFQ) extends the standard WFQ functionality to provide support for user-defined traffic classes on the Versatile Interface Processor (VIP). 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 weighted random early detection (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.

The Distributed WRED (DWRED) feature uses the VIP rather than the RSP to perform the queuing; therefore, it requires a Cisco 7500 series router or Cisco 7000 series router with the RSP7000.

Resource Reservation Protocol Interaction with CBWFQ on a VIP

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.

When an RSVP flow is created, the VIP queuing system reserves the unit of bandwidth allocation in an RSVP queue, similar to the way a traffic class queue is allotted to a CBWFQ traffic class. CBWFQ traffic classes are unaffected by the RSVP flows.

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:

$$\text{average} = (\text{old_average} * (1 - 1/2^n)) + (\text{current_queue_size} * 1/2^n)$$

where n is the exponential weight factor, a user-configurable value.

**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

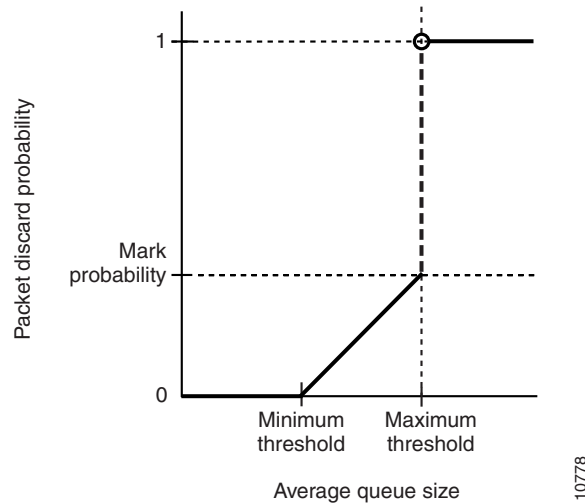
The probability that a packet will be dropped is based on the minimum threshold, maximum threshold, and mark probability denominator.

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 RED 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.

RED 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, RED 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, RED 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 on a VIP 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 for a VIP 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

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.

DWRED Restrictions

DWRED 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 DWRED on the same interface as RSP-based custom queuing, priority queuing, or weighted fair queuing (WFQ). However, you can configure both DWRED and DWFQ on the same interface.

Related Documents

For related information on this feature, refer to the following documents:

- Cisco IOS Release 12.0 *Quality of Service Solutions Configuration Guide*
- Cisco IOS Release 12.0 *Quality of Service Solutions Command Reference*
- *Modular Quality of Service Command-Line Interface*

CBWFQ supports standard and extended numbered access lists only. For information on creating access lists, see the appropriate Cisco IOS Release 12.0 configuration guides and command references.

Supported Platforms

- Cisco 7000 series routers with the RSP7000
- Cisco 7500 series with a Versatile Interface Processor

Supported Standards, MIBs, and RFCs

Standards

None

MIBs

None

For descriptions of supported MIBs and how to use MIBs, see the Cisco MIB web site on CCO at <http://www.cisco.com/public/sw-center/netmgmt/cmtk/mibs.shtml>

RFCs

None

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 12.0 *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 DWRED, Distributed 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 for the VIP 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 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)# policy-map <i>policy-map</i>	Specifies the name of the service policy to be created or modified.
Step 2	Router(config-pmap)# class <i>class-name</i>	Specifies the name of the traffic class to be associated with the service policy.
Step 3	Router(config-pmap-c)# bandwidth <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)# queue-limit <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)# policy-map <i>policy-map</i>	Specifies the name of the service policy to be created or modified.
Step 2	Router(config-pmap)# class <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)# bandwidth <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)# random-detect exponential-weighting-constant <i>exponent</i>	Configures the exponential weight factor used in calculating the average queue length.
Step 5	Router(config-pmap-c)# fair-queue [queue-limit <i>queue-values</i>]	Specifies the number of queues to be reserved for the traffic class.
Step 6	Router(config-pmap-c)# queue-limit <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)# policy-map <i>policy-map</i>	Specifies the name of the service policy to be created or modified.
Step 2	Router(config-pmap)# class <i>class-name</i>	Specifies the name of a traffic class to associate with the service policy.
Step 3	Router(config-pmap-c)# bandwidth <i>bandwidth-kbps</i>	Specifies the amount of bandwidth in kbps to be assigned to the traffic class.
Step 4	Router(config-pmap-c)# random-detect exponential-weighting-constant <i>exponent</i>	Configures the exponential weight factor used in calculating the average queue length.
Step 5	Router(config-pmap-c)# random-detect precedence <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)# policy-map <i>policy-map</i>	Specifies the name of the service policy to be created or modified.
Step 2	Router(config-pmap)# class <i>class-name</i>	Specifies the name of a traffic class whose bandwidth you want to modify.
Step 3	Router(config-pmap-c)# bandwidth <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)# policy-map <i>policy-map</i>	Specifies the name of the service policy to be created or modified.
Step 2	Router(config-pmap)# class <i>class-name</i>	Specifies the name of a traffic class whose queue limit you want to modify.
Step 3	Router(config-pmap-c)# queue-limit <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 for the VIP Using the Modular QoS CLI

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.

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

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.

Command Reference

This section documents new or modified commands that configure the CBWFQ feature. All other commands used with this feature are documented in the Cisco IOS Release 12.0 command references.

- **bandwidth**
- **fair-queue**
- **queue-limit**
- **random-detect exponential-weighting-constant**
- **random-detect precedence**

bandwidth

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

bandwidth *bandwidth-kbps*

no bandwidth *bandwidth-kbps*

Syntax Description

bandwidth-kbps Amount of bandwidth in kilobits per second to be assigned to the traffic class.

Defaults

No default behavior.

Command Modes

Policy map configuration

Command History

Release	Modification
12.0(5)XE	This command was introduced.

Usage Guidelines

- You use the **bandwidth** command when you configure a service policy (with the **policy-map** command) for a traffic class defined by the **class-map** command. The **bandwidth** command specifies the guaranteed bandwidth for traffic in that traffic class.
- On a Versatile Interface Processor (VIP), all traffic that does not match a user-defined traffic class is classified as a member of the default traffic class. The implicit bandwidth allocated to the default traffic class on a VIP 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 for a VIP is implicit (the default traffic class gets all remaining bandwidth not given to the user-defined traffic classes), the **bandwidth** command can not be used on the default traffic class when you configure a VIP.

Examples

The following example modifies the bandwidth for a traffic class called `acl22`. The default traffic class belongs to a service policy map called `polmap6`. The bandwidth is being set at 2000.

```
policy-map polmap6
  class acl22
    bandwidth 2000
  queue-limit 30
```

Related Commands

Command	Description
class	Specifies the traffic class whose bandwidth specification is to be configured or modified.
class class-default	Specifies the default traffic class whose bandwidth is to be configured or modified.
policy-map	Specifies the policy map to which the traffic class belongs whose bandwidth is to be configured or modified.
random-detect exponential-weighting-constant	Configures the exponential weight factor used in calculating the average queue length.
random-detect precedence	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.

fair-queue

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

fair-queue [**queue-limit** *queue-value*]

no fair-queue [**queue-limit** *queue-value*]

Syntax Description

queue-limit	A keyword used to specify or modify the maximum number of packets that a per-flow queue can hold.
<i>queue-value</i>	A number specifying the maximum number of packets that each per-flow queue can accumulate.

Defaults

No default behavior or values.

Command Modes

Policy map configuration

Command History

Release	Modification
12.0(5)XE	This command was introduced.

Usage Guidelines

On a Versatile Interface Processor (VIP), the **fair-queue** command can be used for any traffic class (as opposed to non-VIP platforms, which can only use the **fair-queue** command in 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 10 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) which 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) or (DWRED) 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
class class-default	Specifies the default traffic class for a service policy map.
queue-limit	Specifies or modifies the maximum number of packets that can accumulate in the queue reserved for the traffic class before tail drop or (if WRED is configured as part of the service policy) packet drop is enacted.
random-detect exponential-weighting-constant	Configures the exponential weight factor used in calculating the average queue length.

queue-limit

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

queue-limit *number-of-packets*

no queue-limit *number-of-packets*

Syntax Description

number-of-packets A number in the range of 1 through 64 specifying the maximum number of packets that the queue for this traffic class can accumulate.

Defaults

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 two.

Command Modes

Policy map configuration

Command History

Release	Modification
12.0(5)XE	This command was introduced.

Usage Guidelines

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

Examples

The following example configures a service policy called policy11. The QoS policy for this service policy 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
class	Specifies the traffic class whose queue limit is to be configured or modified.
class class-default	Specifies the default traffic class whose bandwidth is to be configured or modified.
policy-map	Specifies the service policy to which the traffic class belongs whose queue limit is to be configured or modified.

random-detect exponential-weighting-constant

To configure the weighted random early detection (WRED) and distributed WRED (DWRED) on an interface to specify the exponential weight factor for the average queue size calculation for the queue, use the **random-detect exponential-weighting-constant** interface command.

WRED (DWRED) is configured as a QoS policy in a service policy. To specify the exponential weight factor for the average queue size calculation for the queue reserved for a service policy, use the **random-detect exponential-weighting-constant** policy map configuration command. 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

Syntax Description

exponent Exponent from 1 to 16 used in the average queue size calculation.

Defaults

The default exponential weight factor is 9.

Command Modes

Interface configuration when used on an interface.

Policy map configuration when used to specify a service policy in a policy map.

Command History

Release	Modification
11.1 CC	This command was introduced.
12.0(5)XE	This command was introduced in policy map configuration mode.

Usage Guidelines

WRED (DWRED) is a congestion avoidance mechanism that slows traffic by randomly dropping packets when there is congestion. WRED is only useful with protocols like TCP, which respond to dropped packets by decreasing the transmission rate.

Use this command to change the exponent used in the average queue size calculation for WRED and DWRED services.



Note

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

You can use the **random-detect exponential-weighting-constant** command to configure the exponential weight factor for the average queue size calculation for the queue for a service policy or for WRED (DWRED) on an interface.

You can configure WRED (DWRED) as part of the policy for a standard traffic class or the default traffic class. The WRED (DWRED) **random-detect exponential-weighting-constant** command and the WFQ **queue-limit** command are mutually exclusive.

If you configure WRED (DWRED), its packet drop capability is used to manage the queue when packets exceeding the configured maximum count are enqueued. If you configure the WFQ **queue-limit** command in a service policy, tail drop is used.

Note that if you use WRED (DWRED) packet drop instead of tail drop for one or more traffic classes composing a service policy, you must ensure that WRED (DWRED) is not configured for the interface to which you attach that service policy. Attaching a service policy configured to use WRED (DWRED) to an interface disables WRED (DWRED) on that interface.

A VIP2-50 interface processor is strongly recommended when the aggregate line rate of the port adapters on the VIP is greater than DS3. A VIP2-50 interface processor is required for OC-3 rates. To use DWRED, distributed Cisco Express Forwarding (DCEF) switching must first be enabled on the interface. For more information on DCEF, refer to the *Cisco IOS Switching Services Configuration Guide* and the *Cisco IOS Switching Services Command Reference*.

Examples

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

```
router(config)# interface hssi0/0/0
router(config-if)# description 45mbps to R1
router(config-if)# ip address 192.168.14.250 255.255.255.252
router(config-if)# random-detect
router(config-if)# random-detect exponential-weighting-constant 10
```

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
bandwidth	Specifies (or modifies) the bandwidth allocated for a service policy.
fair-queue	Specifies the number of queues to be reserved for a service policy.
random-detect	Specifies the service policy whose queue limit is to be configured or modified.

random-detect precedence	Configures the WRED parameters for a particular IP Precedence.
show policy interface	Displays configurations for all service policies on the specified interface.

random-detect precedence

To configure Weighted Random Early Detection (WRED) and Distributed WRED (DWRED) parameters for a particular IP precedence, use the **random-detect precedence** command. To return the values to the default for the precedence, use the **no** form of this command.

To configure WRED as a QoS policy in a service policy, specify the parameters for a particular IP precedence and use the **random-detect precedence** policy map configuration command. To return the values to the default for the precedence, use the **no** form of this command.

random-detect precedence *precedence min-threshold max-threshold mark-prob-denominator*

no random-detect precedence *precedence min-threshold max-threshold mark-prob-denominator*

Syntax Description

<i>precedence</i>	IP precedence number. The value range is 0 to 7 as well as RSVP. For Cisco 7000 series routers with an RSP7000 and Cisco 7500 series routers with a VIP2-40 (VIP2-50 strongly recommended), the precedence value ranges from 0 to 7 only; see Table 1.
<i>min-threshold</i>	Minimum threshold, in number of packets. The value range of this argument is 1 to 4096. When the average queue length reaches the minimum threshold, WRED drops all packets with the specified IP precedence.
<i>max-threshold</i>	Maximum threshold, in number of packets. The value range of this argument is the value of the <i>min-threshold</i> argument 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>	Denominator for 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. The value range is 1 to 65536. The default is 10; one out of every ten packets is dropped at the maximum threshold.

Defaults

For all IP precedence values, the *mark-prob-denominator* is 10, and the *max-threshold* is based on the output buffering capacity and the transmission speed for the interface.

The default *min-threshold* depends on the IP precedence value. The *min-threshold* for IP precedence 0 corresponds to half of the *max-threshold*. The values for the remaining IP precedence values fall between half the *max-threshold* and the *max-threshold* at evenly spaced intervals.

Table 1 lists the default minimum threshold value for each IP precedence.

Table 1 Default WRED and DWRED Minimum Threshold Values

IP Precedence	Minimum Threshold Value (Fraction of Maximum Threshold Value)
0	8/16
1	9/16
2	10/16
3	11/16

Table 1 Default WRED and DWRED Minimum Threshold Values (continued)

IP Precedence	Minimum Threshold Value (Fraction of Maximum Threshold Value)
4	12/16
5	13/16
6	14/16
7	15/16

Command Modes

Interface configuration when used on an interface

Policy map configuration when used to specify a service policy

Command History

Release	Modification
11.1 CC	This command was introduced.
12.0(5)T	This command was introduced in policy map configuration mode.

Usage Guidelines

When you configure the **random-detect** command on an interface, packets are given preferential treatment based on the IP Precedence of the packet. Use the **random-detect precedence** command to adjust the treatment for different precedences.

If you want WRED (DWRED) to ignore the precedence when determining which packets to drop, enter this command with the same parameters for each precedence. Remember to use reasonable values for the minimum and maximum thresholds.

Note that if you use the **random-detect precedence** command to adjust the treatment for different precedences within a service policy, you must ensure that WRED (DWRED) is not configured for the interface to which you attach that service policy. Attaching a service policy configured to use WRED (DWRED) to an interface using the Modular QoS CLI disables the previous WRED (DWRED) configuration on that interface.

**Note**

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

A VIP2-50 interface processor is strongly recommended when the aggregate line rate of the port adapters on the VIP is greater than DS3. A VIP2-50 interface processor is required for OC-3 rates. To use DWRED, distributed Cisco Express Forwarding (DCEF) switching must first be enabled on the interface. For more information on DCEF, refer to the *Cisco IOS Switching Services Configuration Guide* and the *Cisco IOS Switching Services Command Reference*.

Examples

The following example enables WRED (DWRED) using Legacy CLI (non Modular QoS CLI) on the interface and specifies parameters for the different IP precedences:

```
interface Hssi0/0/0
description 45Mbps to R1
ip address 200.200.14.250 255.255.255.252
random-detect
random-detect precedence 0 32 256 100
random-detect precedence 1 64 256 100
random-detect precedence 2 96 256 100
random-detect precedence 3 120 256 100
random-detect precedence 4 140 256 100
random-detect precedence 5 170 256 100
random-detect precedence 6 290 256 100
random-detect precedence 7 210 256 100
random-detect precedence rsdp 230 256 100
```

The following example uses the Modular QoS CLI to configure a service policy called policy10. Traffic class int101 has these characteristics: a minimum of 2000 kbps of bandwidth are expected to be delivered to this service policy in the event of congestion and a weight factor of 10 is used to calculate the average queue size. For congestion avoidance, WRED packet drop is used, not tail drop. IP Precedence is reset for levels 0 through 5.

```
policy-map policy10
class acl10
bandwidth 2000
random-detect exponential-weighting-constant 10
random-detect precedence 0 32 256 100
random-detect precedence 1 64 256 100
random-detect precedence 2 96 256 100
random-detect precedence 3 120 256 100
random-detect precedence 4 140 256 100
random-detect precedence 5 170 256 100
```

Related Commands

Command	Description
bandwidth	Specifies (or modifies) the bandwidth allocated for a service policy.
fair-queue	Specifies the number of hashes queues to be reserved for a service policy.
random-detect	Specifies the service policy whose queue limit is to be configured or modified.
random-detect precedence	Configures the WRED parameters for a particular IP Precedence.
show policy interface	Displays configurations for all service policies on the specified interface.

■ random-detect precedence