



Low Latency Queueing for Frame Relay

This document describes the Low Latency Queueing for Frame Relay feature. It includes information about the benefits of this new feature, supported platforms, related documents, and more.

This document includes the following sections:

- Feature Overview, page 1
- Supported Platforms, page 4
- Supported Standards, MIBs, and RFCs, page 4
- Prerequisites, page 6
- Configuration Tasks, page 6
- Monitoring and Maintaining Low Latency Queueing for Frame Relay, page 9
- Configuration Examples, page 9
- Command Reference, page 10
- Glossary, page 32

Feature Overview

Low Latency Queueing for Frame Relay is a new feature that provides a strict priority queue (PQ) for voice traffic and weighted fair queues for other classes of traffic. Before the release of this feature, low latency queueing was available at the interface and ATM virtual circuit (VC) levels. It is now available at the Frame Relay VC level when Frame Relay traffic shaping is configured.

Low Latency Queueing, also called priority queueing/class-based weighted fair queueing (PQ/CBWFQ), is a superset of and more flexible than previous Frame Relay Quality of Service offerings, in particular Real-Time Transport Protocol (RTP) prioritization and priority queueing/weighted fair queueing (PQ/WFQ).

With RTP prioritization and PQ/WFQ, traffic that matches a specified User Datagram Protocol (UDP)/RTP port range is considered high priority and allocated to the PQ. With Low Latency Queueing for Frame Relay, you set up classes of traffic according to protocol, interface, or access lists, and then define policy maps to establish how the classes are handled in the priority queue and weighted fair queues.

Queues are set up on a per-permanent virtual circuit (PVC) basis: each PVC has a PQ and an assigned number of fair queues. The fair queues are assigned weights proportional to the bandwidth requirements of each class; a class requiring twice the bandwidth of another will have half the weight. Oversubscription of the bandwidth is not permitted. The command line interface (CLI) will reject a

change of configuration that would cause the total bandwidth to be exceeded. This functionality differs from that of WFQ, in which flows are assigned a weight based on IP precedence. WFQ allows higher precedence traffic to obtain proportionately more of the bandwidth, but the more flows there are, the less bandwidth is available to each flow.

The PQ is policed to ensure that the fair queues are not starved of bandwidth. When you configure the PQ, you specify in kbps the maximum amount of bandwidth available to that queue. Packets that exceed that maximum are dropped. There is no policing of the fair queues.

Low Latency Queueing for Frame Relay is configured using a combination of **class-map**, **policy-map** and Frame Relay map-class commands. The **class-map** command defines traffic classes according to protocol, interface, or access list. The **policy-map** command defines how each class is treated in the queueing system according to bandwidth, priority, queue limit, or Weighted Random Early Detection (WRED). The **service-policy output** map-class command attaches a policy-map to a Frame Relay VC.

Policies not directly related to low latency queueing—for example, traffic shaping, setting IP precedence, and policing—are not supported by the **class-map** and **policy-map** commands for Frame Relay VCs. You must use other configuration mechanisms, such as map-class commands, to configure these policies.

Low Latency Queueing for Frame Relay can be used in conjunction with the features listed in the following sections:

- RTP Prioritization
- Voice over Frame Relay
- Frame Relay Fragmentation
- IP Cisco Express Forwarding Switching

RTP Prioritization

RTP prioritization provides a strict priority queueing scheme for voice traffic. Voice traffic is identified by its RTP port numbers and classified into a priority queue configured by the **frame-relay ip rtp priority** map-class command. You classify traffic as voice by specifying an RTP port number range. If traffic matches the specified range, it is classified as voice and queued in the low latency queueing PQ, as well as the interface priority queue. If traffic does not fall within the specified RTP port range, it is classified by the service policy of the low latency queueing scheme.

The **ip rtp priority** command is available in both interface configuration mode and map-class frame-relay configuration mode. Only the **frame relay ip rtp priority** map-class configuration command is supported in this feature.

Voice over Frame Relay

Voice over Frame Relay (VoFR) uses the low latency queueing PQ rather than its own priority queueing mechanism. The **frame-relay voice bandwidth** map-class command configures the total bandwidth available for VoFR traffic. The visible bandwidth made available to the other queues will be the minimum committed information rate (CIR) less the voice bandwidth.

The **frame-relay voice bandwidth** map-class command also configures a call admission control function, which ensures that there is sufficient VoFR bandwidth remaining before allowing a call. There is no policing of the voice traffic once the call has been established.

For VoFR with no data, all voice and call control packets are queued in the low latency queuing PQ. For VoFR with data, a VoFR PVC may carry both voice and data packets in different subchannels. VoFR data packets are fragmented and interleaved with voice packets to ensure good latency bounds for voice packets as well as scalability for voice and data traffic.

Note that when VoFR is enabled, there is no need to configure a priority class map for voice. The only VoFR commands to be used with Low Latency Queuing for Frame Relay are the **frame-relay voice bandwidth** map-class configuration command and the **vofr data** interface-dlci configuration command.

**Note**

It is possible—though not recommended—to configure other traffic for the PQ at the same time as VoFR. Doing so could cause delays because interleaving non-VoFR packets in the PQ would not be possible, causing the PQ (and any VoFR packets on it) to be held up during fragmentation until the entire fragmented packet has been transmitted.

Frame Relay Fragmentation

The purpose of Frame Relay fragmentation (FRF.12) is to support voice and data packets on lower-speed links without causing excessive delay to the voice packets. Large data packets are fragmented and interleaved with the voice packets.

When FRF.12 is configured with low latency queuing, small packets classified for the PQ pass through unfragmented onto both the low latency queuing PQ and the high priority interface queue. Large packets destined for PQ are shaped and fragmented when dequeued.

Use the **frame-relay fragment** and **service-policy** map-class configuration commands to enable low latency queuing with FRF.12 .

IP Cisco Express Forwarding Switching

IP Cisco express forwarding (CEF) switching is not affected by low latency queuing functionality.

Benefits

Strict Priority Service

Strict priority queueing improves quality of service by allowing delay-sensitive traffic, such as voice, to be pulled from the queue and sent before other classes of traffic.

Flexibility

Low Latency Queueing for Frame Relay allows you to define classes of traffic according to protocol, interface, or access lists. You can then assign characteristics to those classes, including priority, bandwidth, queue limit, and WRED.

Restrictions

Only the following class-map and policy-map commands are supported:

- The class-map **match** command
- The policy-map **priority**, **bandwidth**, **queue-limit**, **random-detect**, and **fair-queue** commands

Related Features and Technologies

The following features and technologies are related to low latency queueing for Frame Relay:

- Frame Relay Traffic Shaping (FRTS)
- RTP prioritization
- Priority Queueing (PQ)
- Voice over Frame Relay (VoFR)
- Weighted Fair Queueing (WFQ)

Related Documents

The following documents provide information related to low latency queueing for Frame Relay:

- *Class-Based Weighted Fair Queueing*, Cisco IOS Release 12.0(5)T
- *IP RTP Priority*, Cisco IOS Release 12.0(5)T
- *Cisco IOS Quality of Service Solutions Configuration Guide*, Release 12.1
- *Cisco IOS Quality of Service Solutions Command Reference*, Release 12.1

Supported Platforms

The Low Latency Queueing for Frame Relay feature runs on the following platforms:

- Cisco 805
- Cisco 1600
- Cisco 1700 series
- Cisco 2500
- Cisco 2600 series
- Cisco 3600 series
- Cisco 3810
- Cisco 7200 series

Supported Standards, MIBs, and RFCs

Standards

No new or modified standards are supported by this feature.

MIBs

No new or modified MIBs are supported by this feature.

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

RFCs

No new or modified RFCs are supported by this feature.

Prerequisites

The following tasks must be completed before Low Latency Queueing for Frame Relay can be enabled:

- Frame Relay traffic shaping must be enabled on the interface.
- An output service policy must be configured in the map-class associated with the interface, subinterface, or data-link connection identifier (DLCI).
- Any queue other than a first-in, first-out (FIFO) queue that is configured in the map-class must be removed. Low Latency Queueing for Frame Relay cannot be configured if there is already a non-FIFO queue configured, except for the default queue that is created when fragmentation is enabled.

Configuration Tasks

See the following sections for configuration tasks for the Low Latency Queueing for Frame Relay feature. Each task in the list is identified as either optional or required.

- Defining Class Maps (Required)
- Configuring Class Policy in the Policy Map (Required)
- Attaching the Service Policy and Enabling Low Latency Queueing for Frame Relay (Required)
- Verifying Configuration of Policy Maps and Their Classes (Optional)

Defining Class Maps

To create a class map containing match criteria against which a packet is checked to determine if it belongs to a class, begin with the **class-map** command in global configuration mode.

	Command	Purpose
Step 1	<code>Router(config)# class-map <i>class-map-name</i></code>	Specifies the name of the class map to be created.
Step 2	<code>Router(config-cmap)# match access-group {<i>access-group</i> name <i>access-group-name</i>}</code>	Specifies the name of the access control list (ACL) against whose contents packets are checked to determine if they belong to the class.
	or	
	<code>Router(config-cmap)# match input-interface <i>interface-name</i></code>	Specifies the name of the input interface used as a match criterion against which packets are checked to determine if they belong to the class.
	or	
	<code>Router(config-cmap)# match protocol <i>protocol</i></code>	Specifies the name of the protocol used as a match criterion against which packets are checked to determine if they belong to the class.

For more details about defining class maps, see the *Cisco IOS Quality of Service Solutions Configuration Guide*.

Configuring Class Policy in the Policy Map

To configure a policy map and create class policies that make up the service policy, begin with the **policy-map** command to specify the policy map name. Then use one or more of the following commands to configure the policy for a standard class or the default class:

- **priority**
- **bandwidth**
- **queue-limit** or **random-detect**
- **fair-queue** (for class-default class only)

For each class that you define, you can use one or more of the commands listed above to configure the class policy. For example, you might specify bandwidth for one class and both bandwidth and queue limit for another class.

The default class of the policy map (commonly known as the class-default class) is the class to which traffic is directed if that traffic does not satisfy the match criteria of the other classes defined in the policy map.

You can configure class policies for as many classes as are defined on the router, up to the maximum of 64. However, the total amount of bandwidth allocated for all classes in a policy map must not exceed the minimum CIR configured for the VC less any bandwidth reserved by the **frame-relay voice bandwidth** and **frame-relay ip rtp priority** commands. If the minimum CIR is not configured, it defaults to one half of the CIR. If all of the bandwidth is not allocated, the remaining bandwidth is allocated proportionally among the classes on the basis of their configured bandwidth.

To configure class policies in a policy map, perform the tasks in the following sections:

- Configuring Class Policy for a Low Latency Queueing Priority Queue
- Configuring Class Policy Using a Specified Bandwidth and WRED Packet Drop
- Configuring the Class-Default Class Policy

Configuring Class Policy for a Low Latency Queueing Priority Queue

To configure a policy map and give priority to a class within the policy map, begin with the **policy-map** command in global configuration mode.

	Command	Purpose
Step 1	Router(config)# policy-map <i>policy-map</i>	Specifies the name of the policy map to be created or modified.
Step 2	Router(config-pmap)# class <i>class-name</i>	Specifies the name of a class to be created and included in the service policy.
Step 3	Router(config-pmap-c)# priority <i>bandwidth-kbps</i>	Creates a strict priority class and specifies the amount of bandwidth in kbps to be assigned to the class.

Configuring Class Policy Using a Specified Bandwidth and WRED Packet Drop

To configure a policy map and create class policies that make up the service policy, begin with the **policy map** command in global configuration mode.

	Command	Purpose
Step 1	Router(config)# policy-map <i>policy-map</i>	Specifies the name of the policy map to be created or modified.
Step 2	Router(config-pmap)# class <i>class-name</i>	Specifies the name of a 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 to be assigned to the class in kbps or as a percentage of the available bandwidth. Bandwidth must be specified in kbps or as a percentage consistently across classes. (Bandwidth of the priority queue must be specified in kbps.)
Step 4	Router(config-pmap-c)# random-detect	Enables WRED.

To configure policy for more than one class in the same policy map, repeat Steps 2 through 4.

Configuring the Class-Default Class Policy

The class-default class is used to classify traffic that does not fall into one of the defined classes. Even though the class-default class is predefined when you create the policy map, you still have to configure it. If a default class is not configured, then traffic that does not match any of the configured classes is given “best-effort” treatment, which means that the network will deliver the traffic if it can, without any assurance of reliability, delay prevention, or throughput.

To configure a policy map and the class-default class, begin with the **policy-map** command in global configuration mode.

	Command	Purpose
Step 1	Router(config)# policy-map <i>policy-map</i>	Specifies the name of the policy map to be created or modified.
Step 2	Router(config-pmap)# class class-default <i>default-class-name</i>	Specifies the default class so that you can configure or modify its policy.
Step 3	Router(config-pmap-c)# bandwidth <i>bandwidth-kbps</i> or Router(config-pmap-c)# fair-queue <i>[number-of-dynamic-queues]</i>	Specifies the amount of bandwidth in kbps to be assigned to the class. Specifies the number of dynamic queues to be reserved for use by flow-based WFQ running on the default class. The number of dynamic queues is derived from the bandwidth of the interface.
Step 4	Router(config-pmap-c)# queue-limit <i>number-of-packets</i>	Specifies the maximum number of packets that the queue for the default class can accumulate.

For more details about configuring class policy in the policy map, see the *Cisco IOS Quality of Service Solutions Configuration Guide*.

Attaching the Service Policy and Enabling Low Latency Queueing for Frame Relay

To attach a service policy to the output interface and enable Low Latency Queueing for Frame Relay, use the following map-class configuration command. When Low Latency Queueing is enabled, all classes configured as part of the service policy map are installed in the fair queueing system.

Command	Purpose
Router(config-map-class)# service-policy output <i>policy-map</i>	Attaches the specified service policy map to the output interface and enables Low Latency Queueing for Frame Relay.

Verifying Configuration of Policy Maps and Their Classes

To display the contents of a specific policy map or all policy maps configured on an interface, use one of the following commands in global configuration mode:

Command	Purpose
Router# show frame-relay pvc <i>dlci</i>	Displays statistics about the PVC and the configuration of classes for the policy map on the specified DLCI.
Router# show policy-map interface <i>interface-name</i>	When FRTS is configured, displays the configuration of classes for all Frame Relay VC-level policy maps. When FRTS is not configured, displays the configuration of classes for the interface-level policy.
Router# show policy-map interface <i>interface-name</i> dlci <i>dlci</i>	When FRTS is configured, displays the configuration of classes for the policy map on the specified DLCI.

Monitoring and Maintaining Low Latency Queueing for Frame Relay

For a list of commands that can be used to monitor Low Latency Queueing for Frame Relay, see the previous section, “Verifying Configuration of Policy Maps and Their Classes.”

Configuration Examples

This section provides a configuration example for Low Latency Queueing for Frame Relay configuration.

Low Latency Queueing for Frame Relay Configuration Example

The following example shows how to configure a PVC shaped to a 64K CIR with fragmentation. The shaping queue is configured with a class for voice, two data classes for IP precedence traffic, and a default class for best-effort traffic. WRED is used as the drop policy on one of the data classes.

The following commands define class maps and the match criteria for the class maps:

```
!
class-map voice
  match access-group 101
!
class-map immediate-data
  match access-group 102
!
class-map priority-data
  match access-group 103

!
access-list 101 permit udp any any range 16384 32767
access-list 102 permit ip any any precedence immediate
access-list 103 permit ip any any precedence priority
```

The following commands create and define a policy map called “mypolicy”:

```
!
policy-map mypolicy
  class voice
    priority 16
  class immediate-data
    bandwidth 32
    random-detect
  class priority-data
    bandwidth 16
  class class-default
    fair-queue 64
    queue-limit 20
```

The following commands enable Frame Relay fragmentation and attach the policy map to DLCI 100:

```
!
interface Serial1/0.1 point-to-point
  frame-relay interface-dlci 100
  class fragment
!
map-class frame-relay fragment
  frame-relay cir 64000
  frame-relay mincir 64000
  frame-relay bc 640
  frame-relay fragment 50
  service-policy output mypolicy
```

Command Reference

This section documents modified commands. All other commands used with this feature are documented in the Cisco IOS Release 12.1 command reference publications.

service-policy

To attach a policy map to an input interface or virtual circuit (VC), or an output interface or VC to be used as the service policy for that interface or VC, use the **service-policy** global configuration command. To remove a service policy from an input or output interface or input or output VC, use the **no** form of this command.

```
service-policy {input | output} policy-map
```

```
no service-policy {input | output}
```

Syntax Description		
input	Attaches the specified policy map to the input interface or input VC.	
output	Attaches the specified policy map to the output interface or output VC.	
<i>policy-map</i>	The name of a service policy map (created using the policy-map command) to be attached.	

Defaults No service policy is specified.

Command Modes

- Global configuration
- VC submode (for a standalone VC)
- Bundle-vc configuration (for ATM VC bundle members)
- Map-class configuration (for Frame Relay VCs)

Command History	Release	Modification
	12.0(5)T	This command was introduced.
	12.1(2)T	This command was modified to enable low latency queueing on Frame Relay VCs.

Usage Guidelines You can attach a single policy map to one or more interfaces or one or more VCs to specify the service policy for those interfaces or VCs.

Currently a service policy specifies class-based weighted fair queueing (CBWFQ). The class policies that make up the policy map are then applied to packets that satisfy the class map match criteria for the class.

To enable Low Latency Queueing for Frame Relay (PQ/CBWFQ), you must first enable Frame Relay traffic shaping on the interface using the **frame-relay traffic-shaping** command in interface configuration mode. You will then attach an output service policy to the Frame Relay VC using the **service-policy** command in map-class configuration mode.

For a policy map to be successfully attached to an interface or ATM VC, the aggregate of the configured minimum bandwidths of the classes that make up the policy map must be less than or equal to 75 percent of the interface bandwidth or the bandwidth allocated to the VC. For a Frame Relay VC, the total

amount of bandwidth allocated must not exceed the minimum CIR configured for the VC less any bandwidth reserved by the **frame-relay voice bandwidth** and **frame-relay ip rtp priority** map-class commands. If not configured, the minimum CIR defaults to half of the CIR.

Configuring CBWFQ on a physical interface is possible only if the interface is in the default queueing mode. Serial interfaces at E1 (2.048 Mbps) and below use WFQ by default; other interfaces use FIFO by default. Enabling CBWFQ on a physical interface overrides the default interface queueing method. Enabling CBWFQ on an ATM PVC does not override the default queueing method.

Attaching a service policy and enabling CBWFQ on an interface renders ineffective any commands related to fancy queueing such as commands pertaining to fair queueing, custom queueing, priority queueing, and Weighted Random Early Detection (WRED). You can configure these features only after you remove the policy map from the interface.

You can modify a policy map attached to an interface or a VC, changing the bandwidth of any of the classes that make up the map. Bandwidth changes that you make to an attached policy map are effective only if the aggregate of the bandwidth amounts for all classes that make up the policy map, including the modified class bandwidth, is less than or equal to 75 percent of the interface bandwidth or the VC bandwidth. If the new aggregate bandwidth amount exceeds 75 percent of the interface bandwidth or VC bandwidth, the policy map is not modified.

Examples

The following examples show how to attach the service policy map called “policy9” to DLCI 100 on output interface Serial1 and enables Low Latency Queueing for Frame Relay:

```
interface Serial1/0.1 point-to-point
  frame-relay interface-dlci 100
  class fragment
!
map-class frame-relay fragment
  service-policy output policy9
```

The following example illustrates attaching the service policy map called “policy9” to the input interface Serial1:

```
interface Serial1
  service-policy input policy9
```

The following example illustrates attaching the service policy map called “policy9” to the input permanent virtual circuit (PVC) called “cisco”:

```
pvc cisco 0/34
  service-policy input policy9
vbr-nt 5000 3000 500
precedence 4-7
```

The following example illustrates attaching the policy called “policy9” to the output interface serial1 to specify the service policy for the interface and enable CBWFQ on it:

```
interface serial1
  service-policy output policy9
```

The following example illustrates attaching the service policy map called “policy9” to the output PVC called “cisco”:

```
pvc cisco 0/5
  service-policy output policy9
vbr-nt 4000 2000 500
precedence 2-3
```

Related Commands	Command	Description
	policy-map	Creates or modifies a policy map that can be attached to one or more interfaces to specify a service policy.
	show frame-relay pvc	Displays statistics about PVCs for Frame Relay interfaces.
	show policy-map	Displays the configuration of all classes that make up the specified service policy map or all classes for all existing policy maps.
	show policy-map interface	Displays the configuration of classes configured for service policies on the specified interface or PVC.

show frame-relay pvc

To display statistics about permanent virtual circuits (PVCs) for Frame Relay interfaces, use the **show frame-relay pvc** command in privileged EXEC mode.

```
show frame-relay pvc [interface interface ][dldci]
```

Syntax Description	Parameter	Description
	interface	(Optional) Indicates a specific interface for which PVC information will be displayed.
	<i>interface</i>	(Optional) Interface number containing the DLCIs for which you wish to display PVC information.
	<i>dldci</i>	(Optional) A specific DLCI number used on the interface. Statistics for the specified PVC are displayed when a DLCI is also specified.

Defaults No default behavior or values.

Command Modes Privileged EXEC

Command History	Release	Modification
	10.0	This command was introduced.
	12.0(1)T	This command was modified to display statistics about virtual access interfaces used for PPP connections over Frame Relay.
	12.0(3)XG	This command was modified to include the fragmentation type and size associated with a particular PVC when fragmentation is enabled on the PVC.
	12.0(4)T	This command was modified to include the fragmentation type and size associated with a particular PVC when fragmentation is enabled on the PVC.
	12.0(5)T	This command was modified to include information on the special voice queue that is created using the queue keyword of the frame-relay voice bandwidth command.
	12.1(2)T	This command was modified to include information about the policy map attached to a specific PVC.

Usage Guidelines Use this command to monitor the PPP link control protocol (LCP) state as being open with an “up” state, or closed with a “down” state.

When “vofr” or “vofr cisco” has been configured on the PVC, and a voice bandwidth has been allocated to the class associated with this PVC, configured voice bandwidth and used voice bandwidth are also displayed.

Statistics Reporting

To obtain statistics about PVCs on all Frame Relay interfaces, use this command with no arguments.

To obtain statistics about a PVC that include policy-map configuration, use this command with the DLCI argument.

Per-VC counters are not incremented at all when either autonomous or silicon switching engine (SSE) switching is configured; therefore, PVC values will be inaccurate if either switching method is used.

Traffic Shaping

Congestion control mechanisms are currently not supported, but the switch passes forward explicit congestion notification (FECN) bits, backward explicit congestion notification (BECN) bits, and discard eligible (DE) bits unchanged from entry to exit points in the network.

If a Local Management Interface (LMI) status report indicates that a PVC is not active, then it is marked as inactive. A PVC is marked as deleted if it is not listed in a periodic LMI status message.

Examples

The various displays in this section show sample output for a variety of PVCs. Some of the PVCs carry data only; some carry a combination of voice and data.

The following is sample output from the **show frame-relay pvc** command for a PVC shaped to a 64K CIR with fragmentation. A policy map is attached to the PVC and is configured with a priority class for voice, two data classes for IP precedence traffic, and a default class for best-effort traffic. WRED is used as the drop policy on one of the data classes:

```
ed2-36b# show frame-relay pvc 100

PVC Statistics for interface Serial1/0 (Frame Relay DTE)

DLCI = 100, DLCI USAGE = LOCAL, PVC STATUS = INACTIVE, INTERFACE = Serial1/0.1

input pkts 0          output pkts 0          in bytes 0
out bytes 0          dropped pkts 0          in FECN pkts 0
in BECN pkts 0        out FECN pkts 0          out BECN pkts 0
in DE pkts 0          out DE pkts 0
out bcast pkts 0      out bcast bytes 0
pvc create time 00:00:42, last time pvc status changed 00:00:42
service policy mypolicy
Class voice
  Weighted Fair Queueing
    Strict Priority
    Output Queue: Conversation 72
      Bandwidth 16 (kbps) Packets Matched 0
      (pkts discards/bytes discards) 0/0
Class immediate-data
  Weighted Fair Queueing
    Output Queue: Conversation 73
      Bandwidth 60 (%) Packets Matched 0
      (pkts discards/bytes discards/tail drops) 0/0/0
      mean queue depth: 0
      drops: class random tail min-th max-th mark-prob
             0      0      0      64   128   1/10
             1      0      0      71   128   1/10
             2      0      0      78   128   1/10
             3      0      0      85   128   1/10
             4      0      0      92   128   1/10
             5      0      0      99   128   1/10
             6      0      0     106  128   1/10
             7      0      0     113  128   1/10
             rsvp  0      0     120  128   1/10
Class priority-data
  Weighted Fair Queueing
    Output Queue: Conversation 74
      Bandwidth 40 (%) Packets Matched 0 Max Threshold 64 (packets)
```

```

      (pkts discards/bytes discards/tail drops) 0/0/0
Class class-default
  Weighted Fair Queueing
    Flow Based Fair Queueing
      Maximum Number of Hashed Queues 64 Max Threshold 20 (packets)
      Output queue size 0/max total 600/drops 0
      fragment type end-to-end          fragment size 50
      cir 64000      bc 640      be 0      limit 80      interval 10
      mincir 64000   byte increment 80   BECN response no
      frags 0        bytes 0          frags delayed 0          bytes delayed 0
      shaping inactive
      traffic shaping drops 0

```

The following is sample output from the **show frame-relay pvc** command that shows the PVC statistics for serial interface 5 (slot 1 and DLCI 55 is up) during a PPP session over Frame Relay:

```

Router# show frame-relay pvc 55

PVC Statistics for interface Serial5/1 (Frame Relay DTE)
DLCI = 55, DLCI USAGE = LOCAL, PVC STATUS = ACTIVE, INTERFACE = Serial5/1.1
  input pkts 9          output pkts 16          in bytes 154
  out bytes 338         dropped pkts 6          in FECN pkts 0
  in BECN pkts 0       out FECN pkts 0        out BECN pkts 0
  in DE pkts 0          out DE pkts 0
  out bcast pkts 0      out bcast bytes 0
  pvc create time 00:35:11, last time pvc status changed 00:00:22
  Bound to Virtual-Access1 (up, cloned from Virtual-Template5)

```

The following is sample output from the **show frame-relay pvc** command for a PVC carrying Voice over Frame Relay configured via the **vofr cisco** command. The **frame-relay voice bandwidth** command has been configured on the class associated with this PVC, as has fragmentation. The fragmentation employed is proprietary to Cisco.

A sample configuration for this scenario is shown first, followed by the output for the **show frame-relay pvc** command:

```
interface serial 0
    encapsulation frame-relay
    frame-relay traffic-shaping
    frame-relay interface-dlci 108
        vofr cisco
        class vofr-class
    map-class frame-relay vofr-class
        frame-relay fragment 100
        frame-relay fair-queue
        frame-relay cir 64000
        frame-relay voice bandwidth 25000
Router# show frame-relay pvc 108
PVC Statistics for interface Serial0 (Frame Relay DTE)
DLCI = 108, DLCI USAGE = LOCAL, PVC STATUS = STATIC, INTERFACE = Serial0
input pkts 1260          output pkts 1271          in bytes 95671
out bytes 98604          dropped pkts 0            in FECN pkts 0
in BECN pkts 0          out FECN pkts 0          out BECN pkts 0
in DE pkts 0            out DE pkts 0
out bcast pkts 1271     out bcast bytes 98604
pvc create time 09:43:17, last time pvc status changed 09:43:17
Service type VoFR-cisco
configured voice bandwidth 25000, used voice bandwidth 0
voice reserved queues 24, 25
fragment type VoFR-cisco          fragment size 100
cir 64000      bc 64000      be 0          limit 1000  interval 125
mincir 32000   byte increment 1000  BECN response no
pkts 2592      bytes 205140  pkts delayed 1296   bytes delayed 102570
shaping inactive
shaping drops 0
Current fair queue configuration:
Discard      Dynamic      Reserved
threshold   queue count  queue count
64           16           2
Output queue size 0/max total 600/drops 0
```

Note that the “fragment type” field in the **show frame-relay pvc** display can have the following entries:

- VoFR-cisco—Indicates that fragmented packets will contain the Cisco proprietary header
- VoFR—Indicates that fragmented packets will contain the FRF.11 Annex C header
- end-to-end—Indicates that pure FRF.12 fragmentation is carried on this virtual circuit

Below is sample output from the **show frame-relay pvc** command for an application employing pure FRF.12 fragmentation. A sample configuration for this scenario is shown first, followed by the output for the **show frame-relay pvc** command:

```
interface serial 0
  encapsulation frame-relay
  frame-relay traffic-shaping
  frame-relay interface-dlci 110
    class frag
  map-class frame-relay frag
    frame-relay fragment 100
    frame-relay fair-queue
    frame-relay cir 64000

Router# show frame-relay pvc 110
PVC Statistics for interface Serial0 (Frame Relay DTE)
DLCI = 110, DLCI USAGE = LOCAL, PVC STATUS = STATIC, INTERFACE = Serial0
input pkts 0          output pkts 243      in bytes 0
out bytes 7290       dropped pkts 0       in FECN pkts 0
in BECN pkts 0      out FECN pkts 0     out BECN pkts 0
in DE pkts 0        out DE pkts 0
out bcast pkts 243  out bcast bytes 7290
pvc create time 04:03:17, last time pvc status changed 04:03:18
fragment type end-to-end      fragment size 100
cir 64000   bc 64000   be 0   limit 1000   interval 125
mincir 32000   byte increment 1000   BECN response no
pkts 486   bytes 14580   pkts delayed 243   bytes delayed 7290
shaping inactive
shaping drops 0
Current fair queue configuration:
Discard   Dynamic   Reserved
threshold queue count queue count
64        16         2
Output queue size 0/max total 600/drops 0
```

Note that when voice is not configured, voice bandwidth output is not displayed.

The following is sample output from the **show frame-relay pvc** command for multipoint subinterfaces carrying data only. The output displays both the subinterface number and the DLCI. This display is the same whether the PVC is configured for static or dynamic addressing. Note that neither fragmentation nor voice is configured on this PVC.

```
Router# show frame-relay pvc
DLCI = 300, DLCI USAGE = LOCAL, PVC STATUS = ACTIVE, INTERFACE = Serial0.103
input pkts 10  output pkts 7  in bytes 6222
out bytes 6034  dropped pkts 0  in FECN pkts 0
in BECN pkts 0  out FECN pkts 0  out BECN pkts 0
in DE pkts 0   out DE pkts 0
outbcast pkts 0  outbcast bytes 0
pvc create time 0:13:11  last time pvc status changed 0:11:46
DLCI = 400, DLCI USAGE = LOCAL, PVC STATUS = ACTIVE, INTERFACE = Serial0.104
input pkts 20  output pkts 8  in bytes 5624
out bytes 5222  dropped pkts 0  in FECN pkts 0
in BECN pkts 0  out FECN pkts 0  out BECN pkts 0
in DE pkts 0   out DE pkts 0
outbcast pkts 0  outbcast bytes 0
pvc create time 0:03:57  last time pvc status changed 0:03:48
```

The following is sample output from the **show frame-relay pvc** command for a PVC carrying voice and data traffic, with a special queue specifically for voice traffic created using the **frame-relay voice bandwidth** command **queue** keyword:

```
Router# show frame-relay pvc interface serial 1 45

PVC Statistics for interface Serial11 (Frame Relay DTE)

DLCI = 45, DLCI USAGE = LOCAL, PVC STATUS = STATIC, INTERFACE = Serial11

input pkts 85          output pkts 289          in bytes 1730
out bytes 6580        dropped pkts 11          in FECN pkts 0
in BECN pkts 0        out FECN pkts 0          out BECN pkts 0
in DE pkts 0          out DE pkts 0
out bcast pkts 0      out bcast bytes 0
pvc create time 00:02:09, last time pvc status changed 00:02:09
Service type VoFR
configured voice bandwidth 25000, used voice bandwidth 22000
fragment type VoFR      fragment size 100
cir 20000      bc 1000      be 0      limit 125      interval 50
mincir 20000      byte increment 125      BECN response no
fragments 290      bytes 6613      fragments delayed 1      bytes delayed 33
shaping inactive
traffic shaping drops 0
Voice Queueing Stats: 0/100/0 (size/max/dropped)
~~~~~
Current fair queue configuration:
Discard      Dynamic      Reserved
threshold  queue count  queue count
64           16           2
Output queue size 0/max total 600/drops 0
```

Table 1 provides a listing of the fields in these displays and a description of each field.

Table 1 show frame-relay pvc Field Descriptions

Field	Description
DLCI	One of the data-link connection identifier (DLCI) numbers for the PVC.
DLCI USAGE	Lists SWITCHED when the router or access server is used as a switch, or LOCAL when the router or access server is used as a DTE.
PVC STATUS	Status of the PVC: ACTIVE, INACTIVE, or DELETED.
INTERFACE	Specific subinterface associated with this DLCI.
input pkts	Number of packets received on this PVC.
output pkts	Number of packets sent on this PVC.
in bytes	Number of bytes received on this PVC.
out bytes	Number of bytes sent on this PVC.
dropped pkts	Number of incoming and outgoing packets dropped by the router at the Frame Relay level.
in FECN pkts	Number of packets received with the FECN bit set.
in BECN pkts	Number of packets received with the BECN bit set.
out FECN pkts	Number of packets sent with the FECN bit set.
out BECN pkts	Number of packets sent with the BECN bit set.
in DE pkts	Number of DE packets received.

Table 1 *show frame-relay pvc Field Descriptions (continued)*

Field	Description
out DE pkts	Number of DE packets sent.
out bcst pkts	Number of output broadcast packets.
out bcst bytes	Number of output broadcast bytes.
pvc create time	Time at which the PVC was created.
last time pvc status changed	Time at which the PVC changed status (active to inactive).
Service type	Type of service performed by this PVC. Can be VoFR or VoFR-cisco.
configured voice bandwidth	Amount of bandwidth in bits per second reserved for voice traffic on this PVC.
used voice bandwidth	Amount of bandwidth in bits per second currently being used for voice traffic.
voice reserved queues	Queue numbers reserved for voice traffic on this PVC. This field was removed in Cisco IOS Release 12.0(5)T.
service policy	Name of the output service policy applied to the VC.
Class	Class of traffic being displayed. Output is displayed for each configured class in the policy.
Output Queue	The WFQ conversation to which this class of traffic is allocated.
Bandwidth	Bandwidth in kpbs or percentage configured for this class.
Packets Matched	Number of packets that matched this class.
Max Threshold	Maximum queue size for this class when WRED is not used.
pkts discards	Number of packets discarded for this class.
bytes discards	Number of bytes discarded for this class.
tail drops	Number of packets discarded for this class because the queue was full.
mean queue depth	Average queue depth based on the actual queue depth on the interface and the exponential weighting constant. It is a moving average. The minimum and maximum thresholds are compared against this value to determine drop decisions.
drops:	WRED parameters.
class	IP Precedence value
random	Number of packets randomly dropped when the mean queue depth is between the minimum threshold value and the maximum threshold value for the specified IP Precedence value.
tail	Number of packets dropped when the mean queue depth is greater than the maximum threshold value for the specified IP Precedence value.
min-th	Minimum WRED threshold in number of packets.
max-th	Maximum WRED threshold in number of packets.
mark-prob	Fraction of packets dropped when the average queue depth is at the maximum threshold.
Maximum Number of Hashed Queues	(Applies to class-default only) Number of queues available for unclassified flows.

Table 1 *show frame-relay pvc Field Descriptions (continued)*

Field	Description
fragment type	Type of fragmentation configured for this PVC. Possible types are: end-to-end—Fragmented packets contain the standard FRF.12 header VoFR—Fragmented packets contain the FRF.11 Annex C header VoFR-cisco—Fragmented packets contain the Cisco proprietary header
fragment size	Size of the fragment payload in bytes.
cir	Current committed information rate (CIR), in bits per second.
bc	Current committed burst size, in bits.
be	Current excess burst size, in bits.
limit	Maximum number of bytes transmitted per internal interval (excess plus sustained).
interval	Interval being used internally (may be smaller than the interval derived from Bc/CIR; this happens when the router determines that traffic flow will be more stable with a smaller configured interval).
mincir	Minimum committed information rate (CIR) for the PVC.
byte increment	Number of bytes that will be sustained per internal interval.
BECN response	Indication that Frame Relay has BECN Adaptation configured.
pkts	Number of packets associated with this PVC that have gone through the traffic shaping system.
frags	Total number of fragments shaped on this VC.
bytes	Number of bytes associated with this PVC that have gone through the traffic shaping system.
pkts delayed	Number of packets associated with this PVC that have been delayed by the traffic shaping system.
frags delayed	Number of fragments delayed in the shaping queue before being transmitted.
bytes delayed	Number of bytes associated with this PVC that have been delayed by the traffic shaping system.
shaping	Indication that shaping will be active for all PVCs that are fragmenting data; otherwise, shaping will be active if the traffic being sent exceeds the CIR for this circuit.
shaping drops	Number of packets dropped by the traffic shaping process.
Voice Queueing Stats	Statistics showing the size of packets, the maximum number of packets, and the number of packets dropped in the special voice queue created using the frame-relay voice bandwidth command queue keyword.
Discard threshold	Maximum number of packets that can be stored in each packet queue. If additional packets are received after a queue is full, they will be discarded.
Dynamic queue count	Number of packet queues reserved for best-effort traffic.
Reserved queue count	Number of packet queues reserved for voice traffic.
Output queue size	Size in bytes of each output queue.
max total	Maximum number of packets of all types that can be queued in all queues.

Table 1 *show frame-relay pvc Field Descriptions (continued)*

Field	Description
drops	Number of frames dropped by all output queues.

Related Commands

Command	Description
frame-relay pvc	Configures Frame Relay PVCs for FRF.8 Frame Relay-ATM Service Interworking.
service-policy	Attaches a policy map to an input interface or VC, or an output interface or VC, to be used as the service policy for that interface or VC.
show dial-peer voice	Displays configuration information and call statistics for dial peers.
show frame-relay fragment	Displays Frame Relay fragmentation details.
show frame-relay vofr	Displays details about FRF.11 subchannels being used on Voice over Frame Relay DLCIs.
show interfaces serial	Displays information about a serial interface.
show policy-map interface	Displays the configuration of classes configured for service policies on the specified interface or PVC.
show traffic-shape queue	Displays information about the elements queued at a particular time at the VC (DLCI) level.

show policy-map interface

To display the configuration of all classes configured for all service policies on the specified interface or to display the classes for the service policy for a specific permanent virtual circuit (PVC) on the interface, use the **show policy-map interface** global configuration command.

```
show policy-map interface interface-name [vc [vpi] vci ][dlci dlci]
```

Syntax Description	
<i>interface-name</i>	Name of the interface or subinterface whose policy configuration is to be displayed.
vc	(Optional) For ATM interfaces only, shows the policy configuration for a specified PVC. The name can be up to 16 characters long.
<i>vpi</i>	(Optional) ATM network virtual path identifier (VPI) for this PVC. The absence of the “/” and a <i>vpi</i> value defaults the <i>vpi</i> value to 0. On the Cisco 7200 and 7500 series routers, this value ranges from 0 to 255. The arguments <i>vpi</i> and <i>vci</i> cannot both be set to 0; if one is 0, the other cannot be 0. If this value is omitted, information for all VCs on the specified ATM interface or subinterface is displayed.
<i>vci</i>	(Optional) ATM network virtual channel identifier (VCI) for this PVC. This value ranges from 0 to 1 less than the maximum value set for this interface by the atm vc-per-vc command. Typically, lower values from 0 to 31 are reserved for specific traffic (for example, F4 OAM, SVC signalling, ILMI, and so on) and should not be used. The VCI is a 16-bit field in the header of the ATM cell. The VCI value is unique only on a single link, not throughout the ATM network, because it has local significance only. The arguments <i>vpi</i> and <i>vci</i> cannot both be set to 0; if one is 0, the other cannot be 0.
dlci	(Optional) Indicates a specific PVC for which policy configuration will be displayed.
<i>dlci</i>	(Optional) A specific DLCI number used on the interface. Policy configuration for the corresponding PVC will be displayed when a DLCI is specified.

Defaults There is no default behavior.

Command Modes Global configuration

Command History

Release	Modification
12.0(5)T	This command was introduced.
12.1(2)T	This command was modified to display information about the policy for all Frame Relay PVCs on the interface, or, if a DLCI is specified, the policy for that specific PVC.

Usage Guidelines

The **show policy-map interface** command displays the configuration for classes on the specified interface or the specified PVC only if a service policy has been attached to the interface or the PVC.

You can use the *pvc-name* argument to display output for a PVC only for Enhanced ATM port adapters (PA-A3) that support per-VC queueing.

The counters displayed after entering the **show policy-map interface** command are updated only if congestion is present on the interface.

The **show policy-map interface** command will display policy information about Frame Relay PVCs only if Frame Relay traffic shaping is enabled on the interface.

Examples

The following example shows how to display class configuration and policy map statistics for all VCs on interface s1/0. A policy map called “mypolicy” is attached to DLCI 100, and a policy-map called “test” is attached to DLCI 200.

```
ed2-36b# show policy-map interface s1/0
Serial1/0.1: DLCI 100 -
output : mypolicy
Class voice
  Weighted Fair Queueing
  Strict Priority
  Output Queue: Conversation 72
    Bandwidth 16 (kbps) Packets Matched 0
    (pkts discards/bytes discards) 0/0
Class immediate-data
  Weighted Fair Queueing
  Output Queue: Conversation 73
    Bandwidth 60 (%) Packets Matched 0
    (pkts discards/bytes discards/tail drops) 0/0/0
    mean queue depth: 0
    drops: class random tail min-th max-th mark-prob
           0 0 0 64 128 1/10
           1 0 0 71 128 1/10
           2 0 0 78 128 1/10
           3 0 0 85 128 1/10
           4 0 0 92 128 1/10
           5 0 0 99 128 1/10
           6 0 0 106 128 1/10
           7 0 0 113 128 1/10
           rsvp 0 0 120 128 1/10
Class priority-data
  Weighted Fair Queueing
  Output Queue: Conversation 74
    Bandwidth 40 (%) Packets Matched 0 Max Threshold 64 (packets)
    (pkts discards/bytes discards/tail drops) 0/0/0
Class class-default
  Weighted Fair Queueing
  Flow Based Fair Queueing
  Maximum Number of Hashed Queues 64 Max Threshold 20 (packets)
Serial1/0.2: DLCI 200 -
output : test
Class tcp
  Weighted Fair Queueing
  Output Queue: Conversation 25
    Bandwidth 20 (kbps) Packets Matched 0
    (pkts discards/bytes discards/tail drops) 0/0/0
    mean queue depth: 0
    drops: class random tail min-th max-th mark-prob
           0 0 0 64 128 1/10
           1 0 0 71 128 1/10
           2 0 0 78 128 1/10
           3 0 0 85 128 1/10
           4 0 0 92 128 1/10
           5 0 0 99 128 1/10
           6 0 0 106 128 1/10
           7 0 0 113 128 1/10
           rsvp 0 0 120 128 1/10
```

The following example shows how to display the configuration of classes that make up the policy map for a specific Frame Relay VC on interface s1/0.

```
ed2-36b# show policy-map interface s1/0.1 dlc1 100
Serial1/0.1: DLCI 100 -
output : mypolicy
Class voice
  Weighted Fair Queueing
  Strict Priority
  Output Queue: Conversation 72
    Bandwidth 16 (kbps) Packets Matched 0
    (pkts discards/bytes discards) 0/0
Class immediate-data
  Weighted Fair Queueing
  Output Queue: Conversation 73
    Bandwidth 60 (%) Packets Matched 0
    (pkts discards/bytes discards/tail drops) 0/0/0
  mean queue depth: 0
  drops: class random tail min-th max-th mark-prob
         0      0      0    64   128   1/10
         1      0      0    71   128   1/10
         2      0      0    78   128   1/10
         3      0      0    85   128   1/10
         4      0      0    92   128   1/10
         5      0      0    99   128   1/10
         6      0      0   106   128   1/10
         7      0      0   113   128   1/10
         rsvp   0      0   120   128   1/10
Class priority-data
  Weighted Fair Queueing
  Output Queue: Conversation 74
    Bandwidth 40 (%) Packets Matched 0 Max Threshold 64 (packets)
    (pkts discards/bytes discards/tail drops) 0/0/0
Class class-default
  Weighted Fair Queueing
  Flow Based Fair Queueing
  Maximum Number of Hashed Queues 64 Max Threshold 20 (packets)
```

The following example shows how to display configurations for classes on the output interface e1/1:

```
Router# show policy-map interface output e1/1
Ethernet1/1 output : pol
Weighted Fair Queueing
Class class1
  Output Queue: Conversation 264
    Bandwidth 937 (kbps) Max Threshold 64 (packets)
    (total/discards/tail drops) 11548/0/0
Class class2
  Output Queue: Conversation 265
    Bandwidth 937 (kbps) Max Threshold 64 (packets)
    (total/discards/tail drops) 11546/0/0
Class class3
  Output Queue: Conversation 266
    Bandwidth 937 (kbps) Max Threshold 64 (packets)
    (total/discards/tail drops) 11546/0/0
Class class4
  Output Queue: Conversation 267
    Bandwidth 937 (kbps) Max Threshold 64 (packets)
    (total/discards/tail drops) 11702/0/0
Class class5
  Output Queue: Conversation 268
    Bandwidth 937 (kbps) Max Threshold 64 (packets)
    (total/discards/tail drops) 11701/0/0
Class class6
  Output Queue: Conversation 269
    Bandwidth 937 (kbps) Max Threshold 64 (packets)
    (total/discards/tail drops) 11702/0/0
Class class7
  Output Queue: Conversation 270
    Bandwidth 937 (kbps) Max Threshold 64 (packets)
    (total/discards/tail drops) 11857/0/0
Class class8
  Output Queue: Conversation 271
    Bandwidth 937 (kbps) Max Threshold 64 (packets)
    (total/discards/tail drops) 11858/1/0
```

The following example shows how to display configurations for classes that make up the service policy for the output VC 0/101 on the output interface atm2/0.6:

```

qos4-72a# show policy-map interface atm2/0.6
ATM2/0.6: VC 0/101 - output : p1
  Weighted Fair Queueing
    Class c-vc1-c1
      Output Queue: Conversation 264
      Bandwidth 31 (kbps)
      mean queue depth: 1
      drops: class random tail min-th max-th mark-prob
             0      0      0    100   200   1/10
             1      0      0    105   200   1/10
             2      0      0    110   200   1/10
             3      0      0    115   200   1/10
             4      0      0    120   200   1/10
             5      0      0    125   200   1/10
             6      0      0    130   200   1/10
             7      0      0    135   200   1/10
             rsvp  0      0    140   200   1/10
    Class c-vc1-c2
      Output Queue: Conversation 265
      Bandwidth 54 (kbps)
      mean queue depth: 1
      drops: class random tail min-th max-th mark-prob
             0      0      0     60   100   1/10
             1      0      0     65   100   1/10
             2      0      0     70   100   1/10
             3      0      0     75   100   1/10
             4      0      0     80   100   1/10
             5      0      0     83   100   1/10
             6      0      0     85   100   1/10
             7      0      0     87   100   1/10
             rsvp  0      0     90   100   1/10
    Class c-vc1-c3
      Output Queue: Conversation 266
      Bandwidth 77 (kbps)
      mean queue depth: 0
      drops: class random tail min-th max-th mark-prob
             0      0      0      1    10   1/10
             1      0      0      2    10   1/10
             2      0      0      3    10   1/10
             3      0      0      4    10   1/10
             4      0      0      5    10   1/10
             5      0      0      6    10   1/10
             6      0      0      7    10   1/10
             7      0      0      7    10   1/10
             rsvp  0      0      7    10   1/10
    Class c-vc1-c4
      Output Queue: Conversation 267
      Bandwidth 100 (kbps)
      mean queue depth: 9
      drops: class random tail min-th max-th mark-prob
             0      0      0      1    10   1/10
             1      9     220    2     10   1/10
             2     24     645    3     10   1/10
             3     22     844    4     10   1/10
             4      0      0      5     10   1/10
             5     23     351    6     10   1/10
             6     28     213    7     10   1/10
             7     59     540    7     10   1/10
             rsvp  0      0      7     10   1/10
    Class c-vc1-c5
      Output Queue: Conversation 268

```

■ show policy-map interface

```

Bandwidth 123 (kbps)
mean queue depth: 150
drops: class random tail min-th max-th mark-prob
      0 120 1777 50 150 1/50
      1 136 1549 60 150 1/50
      2 88 2354 70 150 1/50
      3 121 1569 80 150 1/50
      4 122 1717 80 150 1/50
      5 0 0 90 150 1/50
      6 0 0 100 150 1/50
      7 105 2058 110 150 1/50
      rsvp 0 0 120 150 1/50
Class c-vc1-c6
Output Queue: Conversation 269
Bandwidth 146 (kbps) Max Threshold 64 (packets)
(total/discards/tail drops) 50216/32696/0
Class c-vc1-c7
Output Queue: Conversation 270
Bandwidth 216 (kbps) Max Threshold 64 (packets)
(total/discards/tail drops) 74577/51994/0
Class class-default
Flow Based Fair Queueing
Number of Hashed Queues 256
drops: class random tail min-th max-th mark-prob
      0 101 828 50 150 1/50
      1 87 1154 60 150 1/50
      2 115 476 70 150 1/50
      3 116 444 80 150 1/50
      4 123 338 80 150 1/50
      5 92 1042 90 150 1/50
      6 79 1068 100 150 1/50
      7 110 740 110 150 1/50
      rsvp 0 0 120 150 1/50

```

Table 2 provides a listing of the fields in these displays and a description of each field.

Table 2 *show policy-map interface Field Descriptions*

Field	Description
output	Name of the output service policy applied to the VC.
Class	Class of traffic being displayed. Output is displayed for each configured class in the policy.
Output Queue	The WFQ conversation to which this class of traffic is allocated.
Bandwidth	Bandwidth in kbps or percentage configured for this class.
Packets Matched	Number of packets that matched this class.
Max Threshold	Maximum queue size for this class when WRED is not used.
pkts discards	Number of packets discarded for this class.
bytes discards	Number of bytes discarded for this class.
tail drops	Number of packets discarded for this class because the queue was full.
mean queue depth	Average queue depth based on the actual queue depth on the interface and the exponential weighting constant. It is a moving average. The minimum and maximum thresholds are compared against this value to determine drop decisions.
drops:	WRED parameters.

Table 2 *show policy-map interface Field Descriptions (continued)*

Field	Description
class	IP Precedence value
random	Number of packets randomly dropped when the mean queue depth is between the minimum threshold value and the maximum threshold value for the specified IP Precedence value.
tail	Number of packets dropped when the mean queue depth is greater than the maximum threshold value for the specified IP Precedence value.
min-th	Minimum WRED threshold in number of packets.
max-th	Maximum WRED threshold in number of packets.
mark-prob	Fraction of packets dropped when the average queue depth is at the maximum threshold.
Maximum Number of Hashed Queues	(Applies to class-default only) Number of queues available for unclassified flows.

Related Commands

Command	Description
show frame-relay pvc	Displays statistics about PVCs for Frame Relay interfaces.
show policy-map	Displays the configuration of all classes that make up the specified service policy map or all classes for all existing policy maps.
show policy-map class	Displays the configuration for the specified class of the specified policy map.

Glossary

CBWFQ—Class-Based Weighted Fair Queueing. Extends the standard WFQ functionality to provide support for user-defined traffic classes.

CIR—Committed Information Rate. Rate at which a Frame Relay network agrees to transfer information under normal conditions, averaged over a minimum increment of time.

Class-Based Weighted Fair Queueing—See CBWFQ.

DLCI—Data-link connection identifier. Value that specifies a permanent virtual circuit (PVC) or switched virtual circuit (SVC) in a Frame Relay network.

FIFO queueing—First-in, first-out queueing. FIFO involves buffering and forwarding of packets in the order of arrival. FIFO embodies no concept of priority or classes of traffic. There is only one queue, and all packets are treated equally. Packets are sent out an interface in the order in which they arrive.

Frame Relay Traffic Shaping—See FRTS.

FRF.12—The FRF.12 Implementation Agreement was developed to allow long data frames to be fragmented into smaller pieces and interleaved with real-time frames. In this way, real-time voice and non-real-time data frames can be carried together on lower-speed links without causing excessive delay to the real-time traffic.

FRTS—Frame Relay Traffic Shaping. FRTS uses queues on a Frame Relay network to limit surges that can cause congestion. Data is buffered and then sent into the network in regulated amounts to ensure that the traffic will fit within the promised traffic envelope for the particular connection.

PQ/CBWFQ—Priority Queueing/Class-Based Weighted Fair Queueing. A feature that brings strict priority queueing to CBWFQ. Strict priority queueing allows delay-sensitive data such as voice to be dequeued and sent first (before packets in other queues are dequeued), giving delay-sensitive data preferential treatment over other traffic.

RTP—Real-Time Transport Protocol. One of the IPv6 protocols. RTP is designed to provide end-to-end network transport functions for applications transmitting real-time data, such as audio, video, or simulation data, over multicast or unicast network services. RTP provides services such as payload type identification, sequence numbering, time-stamping, and delivery monitoring to real-time applications.

UDP—User Datagram Protocol. Connectionless transport layer protocol in the TCP/IP protocol stack. UDP is a simple protocol that exchanges datagrams without acknowledgment or guaranteed delivery, requiring that error processing and retransmission be handled by other protocols.

VoFR—Voice over Frame Relay. Enables a router to carry voice traffic over a Frame Relay network. When sending voice traffic over Frame Relay, the voice traffic is segmented and encapsulated for transit across the Frame Relay network using FRF.12 encapsulation.

Voice over Frame Relay—See VoFR.

WFQ—Weighted Fair Queueing. Congestion management algorithm that identifies conversations (in the form of traffic streams), separates packets that belong to each conversation, and ensures that capacity is shared fairly among these individual conversations. WFQ is an automatic way of stabilizing network behavior during congestion and results in increased performance and reduced retransmission.

WRED—Weighted Random Early Detection. Combines IP Precedence and standard Random Early Detection (RED) to allow for preferential handling of voice traffic under congestion conditions without exacerbating the congestion. WRED uses and interprets IP Precedence to give priority to voice traffic over data traffic, dropping only data packets.