APPENDIX A

Configuring Frame Relay QoS Using Frame Relay Legacy Commands

This appendix describes how to configure Frame Relay QoS using Frame Relay legacy commands using the following procedures:

1. Creating a Map Class, page A-1
2. Enabling Weighted Fair Queuing on the Interface, page A-2
3. Specifying a DLCI Queue Weight, page A-2
4. Specifying Tail Drop, page A-3
5. Enabling RED Drop, page A-3
6. Configuring Frame Relay IP RTP Priority, page A-4
7. Configuring Frame Relay Traffic Shaping, page A-5

Creating a Map Class

To create a Frame Relay map class, enter the following command in global configuration mode:

Command	Purpose
`Router(config)# map-class frame-relay` *class-map-name*	Creates a Frame Relay map class. *class-map-name* is the name of the class map.

Example A-1 creates a map class named camry.

Example A-1 Configuring a Frame Relay Map Class

```
Router(config)# map-class frame-relay camry
Router(config-map-class)#
```

Enabling Weighted Fair Queuing on the Interface

Frame Relay weighted fair queuing allows each PVC to receive a weighted share of the link bandwidth.

To enable Frame Relay weighted fair queuing to the interface, enter the following command in map class configuration mode:

Command	Purpose
`Router(config-map-class)# frame-relay interface-queue fair`	Enables weighted fair queuing. **Note** An error message displays if a service policy is already attached to the interface.

Example A-2 enables Frame Relay weighted fair queuing.

Example A-2 Enabling Frame Relay Weighted Fair Queuing

```
Router(config-map-class)# frame-relay interface-queue fair
Router(config-map-class)#
```

Specifying a DLCI Queue Weight

Frame Relay weighted fair queuing allows each PVC to receive a weighted share of the link bandwidth. To allocate more or less bandwidth share to a PVC, use the following command in map class configuration mode:

Command	Purpose
`Router(config-map-class)# frame-relay interface-queue fair weight` *weight*	Specifies a DLCI queue weight. *weight* is the value assigned to the DLCI (1 to 99). If you do not specify a value, the router treats all PVCs equally.

Example A-3 sets the DLCI queue weight to 80.

Example A-3 Configuring a DLCI Queue Weight

```
Router(config-map-class)# frame-relay interface-queue fair weight 80
Router(config-map-class)#
```

Specifying Tail Drop

You can only specify one packet drop policy on a Frame Relay interface, either tail drop or RED drop.

To set the maximum threshold for packet tail drop, enter the following command in map class configuration mode. This procedure is optional.

Command	Purpose
`Router(config-map-class)# frame-relay interface-queue fair queue-limit` *limit*	Configures tail drop. *limit* is the queue maximum threshold value for packet drop. Valid values are from 32 to 16,384 in powers of 2 (for example, 64, 128, 256, and so on).

Example A-4 sets the threshold limit for packet drop to 2048.

Example A-4 Configuring Tail Drop

```
Router(config-map-class)# frame-relay interface-queue fair queue-limit 2048
Router(config-map-class)#
```

Enabling RED Drop

You can only specify one packet drop policy on a Frame Relay interface. You cannot configure RED on a priority queue.

To enable RED drop to distribute traffic losses in the event of a buffer overflow, enter the following commands in map class configuration mode:

	Command	Purpose
Step 1	`Router(config-map-class)# frame-relay interface-queue fair random-detect`	(Optional) Enables RED drop.
Step 2	`Router(config-map-class)# frame-relay interface-queue fair random-detect exponential-weight-constant` *exp*	Specifies an exponential weight constant. *exp* is the exponential-weight-constant value and is a number from 1 to 16.
Step 3	`Router(config-map-class)# frame-relay interface-queue fair random-detect precedence` [*precedence min_threshold max_threshold probability*]	Specifies precedence parameters. *precedence* is a number between 0 and 7, where 0 represents low-priority traffic and 7 represents high-priority traffic. *min_threshold* is the number of packets from 1 to 32,768. *max_threshold* is the number of packets from 1 to 32,768. *probability* is the mark-probability-denominator value. Valid values are from 1 to 65,536.

Example A-5 sets the exponential weight constant to 16.

Example A-5 Setting the Exponential Weight Constant

```
Router(config-map-class)# frame-relay interface-queue fair random-detect
Router(config-map-class)# frame-relay interface-queue fair random-detect
exponential-weight-constant 16
```

Example A-6 sets the precedence parameters.

Example A-6 Setting the Precedence Parameters

```
Router(config-map-class)# frame-relay interface-queue fair random-detect precedence 2
32000 32000 65000
Router(config-map-cl)#
```

Configuring Frame Relay IP RTP Priority

The Frame Relay IP RTP Priority feature allows Voice over Internet Protocol (VoIP) packet classification at the DLCI level. This classification is based on whether or not the packet IP Real-Time Protocol (RTP) has an even-numbered UDP port within (or outside of) the specified range. Only even ports are matched because they carry the real-time data streams. Odd ports are not matched because they only carry control information. If the IP RTP even-numbered port is within the configured range, the packet is identified as a real-time packet. Real-time packets are given higher priority than regular data packets and are queued to a priority output queue.

Note You must configure the Frame Relay weighted fair queue interface to use this feature (see "Specifying a DLCI Queue Weight" section on page A-2). A priority queue is created for each DLCI that has **frame-relay ip rtp priority** configured.

Parameters for Frame Relay IP RTP priority are configured using the **map-class frame-relay** command (see the "Creating a Map Class" section on page A-1). When a map class with IP RTP defined is associated with a Frame Relay interface, all DLCIs created for that interface are assigned a priority queue for the RTP packets. A different map class can be associated with a different DLCI, which means each DLCI is assigned a different RTP port range.

The following are examples of priority queues:

- DLCI 2 RTP queue
- DLCI 1 RTP queue

The following are examples of non-priority queues:

- NCQ
- DLCI 3 data queue
- DLCI 2 data queue

To specify the RTP priority policy, enter the following command in map class configuration mode:

Command	Purpose
`Router(config-map-c)# frame-relay ip rtp priority` *min_udp_port port_range bandwidth*	Specifies the Frame Relay IP RTP priority policy. *min_udp_port* is the starting UDP destination port number. Valid numbers are from 2000 to 65,535. *port_range* is the range of UDP destination port values. Valid values are from 0 to 16,383. *bandwidth* is the maximum allowed bandwidth in kbps. Valid values are from 1 to 2000.

Example A-7 sets the IP RTP priority parameters for the map class named voice-traffic and applies the class to serial interface 1/0/0/1:1. The example also assigns DLCI 927 and DLCI 928 to the voice-traffic class.

Example A-7 Setting the IP RTP Priority Parameters

```
Router(config)# map-class frame-relay voice-traffic
Router(config-map-c)# frame-relay ip rtp priority 16384 16383 128
Router(config-map-c)# exit
Router(config)# interface serial 1/0/0/1:1
Router(config-if)# ip address 1.1.0.2 255.255.255.252
Router(config-if)# encapsulation frame-relay
Router(config-if)# frame-relay interface-queue fair
Router(config-if)# frame-relay class voice-traffic
Router(config-if)# frame-relay interface-dlci 927
Router(config-if)# frame-relay interface-dlci 928
```

Configuring Frame Relay Traffic Shaping

The Cisco 10000 series router supports Frame Relay traffic shaping (FRTS) and generic traffic shaping (GTS) on Frame Relay interfaces; however, you cannot enable both of these interfaces at the same time. While GTS supports traffic at the modular CLI level, FRTS supports traffic shaping per PVC on the interface by using Frame Relay encapsulation.

To configure Frame Relay traffic shaping, enter the following commands beginning in interface configuration mode:

	Command	Purpose
Step 1	`Router(config)# map-class frame-relay` *map-class-name*	Creates a Frame Relay map class for Frame Relay traffic shaping. *map-class-name* is the name of the map class.
Step 2	`Router(config-map-c)# frame-relay traffic-rate` *bps* or `Router(config-map-c)# frame-relay cir` *bps* and `Router(config-map-c)# frame-relay mincir` *bps*	Sets the traffic shape rate. *bps* is a number from 1 to 45,000,000. Sets the committed information rate (CIR). Sets the minimum rate in times of congestion.
Step 3	`Router(config-map-c)# exit`	Exits map-class configuration mode.
Step 4	`Router(config)# interface` *type slot/subslot/port*	Configures the interface you specify.
Step 5	`Router(config-if)# ip address` *ip-address subnet-mask*	Configures an IP address and subnet mask for the interface.
Step 6	`Router(config-if)# encapsulation frame-relay`	Enables Frame Relay encapsulation.
Step 7	`Router(config-if)# frame-relay traffic-shaping`	Enables Frame Relay traffic shaping.
Step 8	`Router(config-if)# frame-relay class` *map-class-name*	Applies the Frame Relay map class to the DLCIs on this interface.
Step 9	`Router(config-if)# frame-relay interface-dlci` *dlci-number*	Creates the DLCIs on this interface. The DLCIs inherit the Frame Relay map class applied on this interface.

Example A-8 creates a Frame Relay map class named frts with a committed information rate of 64,000 and a minimum rate of 64,000 during congestion. The example enables traffic shaping on serial interface 1/0/0/1:1. DLCI 27, created on the interface, inherits the frts class applied to the interface.

Example A-8 Enabling Frame Relay Traffic Shaping

```
Router(config)# map-class frame-relay frts
Router(config-map-c)# frame-relay cir 64000
Router(config-map-c)# frame-relay mincir 64000
Router(config-map-c)# exit
Router(config)# interface serial 1/0/0/1:1
Router(config-if)# ip address 1.1.0.2 255.255.255.252
Router(config-if)# encapsulation frame-relay
Router(config-if)# frame-relay traffic-shaping
Router(config-if)# frame-relay class frts
Router(config-if)# frame-relay interface-dlci 27
```