

[Solutions](#) [Products](#) [Ordering](#) [Support](#) [Partners](#) [Training](#) [Corporate](#)

Tech Notes



# Understanding Queueing on Frame Relay Router Interfaces

[TAC Notice: What's Changing on TAC Web](#)

Document ID: 22304

## Contents

[Introduction](#)[Prerequisites](#)[Requirements](#)[Components Used](#)[Conventions](#)[Layers of Queues](#)[PVC Queueing](#)[Interface-Level Queueing](#)[FIFO Queueing](#)[Dual FIFO](#)[PIPQ](#)[Tuning the TX Ring](#)[NetPro Discussion Forums - Featured Conversations](#)[Related Information](#)**Help us help you.****Please rate this document.**

- Excellent
- Good
- Average
- Fair
- Poor

**This document solved my problem.**

- Yes
- No
- Just browsing

**Suggestions for improvement:**

(256 character limit)

**Optional contact information:**

Name:

Email:

## Introduction

This document reviews the hierarchical queueing architecture on serial interfaces configured with Frame Relay encapsulation. When configured with Frame Relay Traffic Shaping (FRTS), Frame Relay interfaces support the following queue layers:

- PVC queue
- Interface-level queue

## Prerequisites

### Requirements

Readers of this document should be knowledgeable of:

- [Frame Relay configuration](#)
- Cisco 2600, 3600, and 7200 Series routers
- [FRTS](#)

## Components Used

The configurations used in this document were captured on a Cisco 7200 Series router with the following hardware and software:

- PA-MC-4T1 Multichannel T1 Port Adapter
- Cisco IOS<sup>®</sup> Software Release 12.2(6)

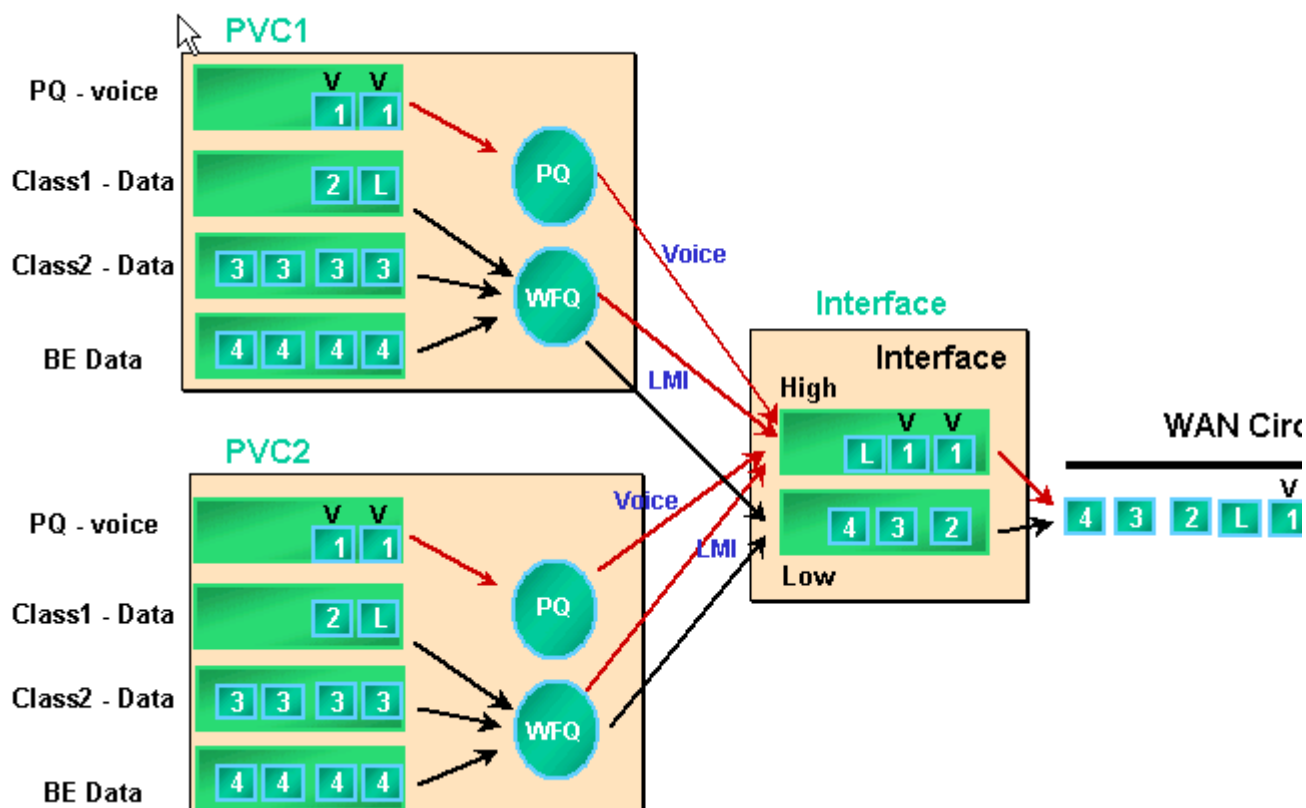
The information presented in this document was created from devices in a specific lab environment. All of the devices used in this document started with a cleared (default) configuration. If you are working in a live network, ensure that you understand the potential impact of any command before using it.

## Conventions

For more information on document conventions, refer to [Cisco Technical Tips Conventions](#).

## Layers of Queues

The following figure illustrates the two layers of queues when FRTS is applied to the interface. Applying FRTS and Frame Relay Forum Implementation Agreements (FRF.12) causes the interface-level queue to change to dual FIFO queueing depending on the platforms that support this queueing technique. The two queues include a high-priority queue to carry Voice over IP (VoIP) and certain control packets and a low-priority queue to carry all other packets. For more information on dual FIFO queueing, see the [Dual FIFO](#) section.



Frame Relay interfaces support interface queues as well as PVC queues when FRTS and PVC queuing are enabled. Each PVC queue also supports a separate weighted fair queuing (WFQ) system, if the PVC queue is configured as WFQ.

## PVC Queuing

Both Frame Relay and ATM interfaces can support multiple virtual circuits (VCs). Depending on the hardware, these interfaces support PVC queues, which ensure one congested VC does not consume all the memory resources and impact other (non-congested) VCs.

The [frame-relay traffic-shaping](#) command enables both traffic shaping and PVC queuing for all VCs on a Frame Relay interface. PVC traffic shaping provides more control over traffic flow on an individual VC. Traffic shaping combined with VC queuing limits the interface bandwidth consumption for a single VC. Without any shaping, a VC can consume all interface bandwidth and starve other VCs.

If you do not specify shaping values, default values for average rate and burst size are applied. When the offered load to the VC exceeds the shaping values, excess packets are stored in the VC's packet-buffering queue. Once the packets are buffered, you can apply a queuing mechanism and effectively control the order of packets dequeued from the VC queue to the interface queue. By default, the PVC queues use first come, first served queuing with a queue limit of 40 packets. Use the [frame-relay holdq](#) command in map-class configuration mode to change this value. Alternately, you can apply low latency queuing (LLQ) or class-based weighted fair queuing (CBWFQ) using a Quality of Service (QoS) policy configured with the commands of the modular QoS command-line interface (CLI) (MQC). In addition, you can apply WFQ directly inside the map-class with the [fair queue](#) command. This command configures your router to classify traffic according to flow and places these flows in their own subqueues. Thus, the [fair queue](#) command creates a WFQ system per VC.

Detailed queueing mechanisms for the PVC queues are described below.

1. Execute the **show frame-relay pvc 20** command. The Frame Relay Data Link Connection Identifier (DLCI) is identified by the 20. The following output shows no queueing information because FRTS is not enabled.

```
Router# show frame PVC 20

PVC Statistics for interface Serial6/0:0 (Frame Relay DTE)

DLCI = 20, DLCI USAGE = LOCAL, PVC STATUS = DELETED, INTERFACE = Serial

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 00:00:38, last time PVC status changed 00:00:25
```

2. Configure FRTS using the **frame-relay traffic-shaping** command in interface configuration mode under the physical interface. Execute the **show frame-relay PVC [dlci]** command again.

```
Router# show frame-relay PVC 20

PVC Statistics for interface Serial6/0:0 (Frame Relay DTE)

DLCI = 20, DLCI USAGE = LOCAL, PVC STATUS = DELETED, INTERFACE = Serial

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 00:04:59, last time PVC status changed 00:04:46
cir 56000      bc 7000      be 0          byte limit 875      interval 125

!--- Shaping parameters.

mincir 28000      byte increment 875      Adaptive Shaping none
pkts 0           bytes 0                pkts delayed 0        bytes delayed 0
shaping inactive
traffic shaping drops 0
Queueing strategy: fifo
!--- Queue mechanism.

Output queue 0/40, 0 drop, 0 dequeued
!--- Queue size.
```

3. By default, the PVC queues use an output queue limit of 40 packets. Use the **frame-relay holdq** command to configure a non-default value.

```
Router(config)# map-class frame-relay shaping
Router(config-map-class)# no frame-relay adaptive-shaping
Router(config-map-class)# frame-relay holdq 50

Router(config)# interface serial 6/0:0.1
```

```

Router(config-subif)# frame-relay interface-dlci 20
%PVC is already defined
Router(config-fr-dlci)# class shaping
Router(config-fr-dlci)# end
Router# sh frame PVC 20

```

PVC Statistics for interface Serial6/0:0 (Frame Relay DTE)

DLCI = 20, DLCI USAGE = LOCAL, PVC STATUS = DELETED, INTERFACE = Serial

```

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 00:11:06, last time PVC status changed 00:10:53
cir 56000   BC 7000   be 0          byte limit 875   interval
mincir 28000   byte increment 875   Adaptive Shaping none
pkts 0      bytes 0          pkts delayed 0      bytes delayed
shaping inactive
traffic shaping drops 0
Queueing strategy: FIFO
Output queue 0/50, 0 drop, 0 dequeued
!--- Queue size.

```

4. The PVC queues also support [CBWFQ](#) and [LLQ](#), which you can configure using a service policy and the commands of the MQC. The following sample output was captured on the Frame Relay PVC after a QoS service policy was applied.

```

Router(config)# class-map gold
Router(config-cmap)# match ip dscp 46
Router(config-cmap)# class-map silver
Router(config-cmap)# match ip dscp 26
Router(config-cmap)# policy-map sample
Router(config-pmap)# class gold
Router(config-pmap-c)# priority 64
Router(config-pmap-c)# class silver
Router(config-pmap-c)# bandwidth 32

```

```

Router(config)# map-class frame-relay map1
Router(config-map-class)# service-policy output sample

```

```

Router(config-if)# frame-relay interface-dlci 20
Router(config-fr-dlci)# class map1
Router# show frame-relay PVC 20

```

PVC Statistics for interface Serial6/0:0 (Frame Relay DTE)

DLCI = 20, DLCI USAGE = LOCAL, PVC STATUS = DELETED, INTERFACE = Serial

```

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 00:12:50, last time PVC status changed 00:12:37
cir 56000   bc 7000   be 0          byte limit 875   interval 125
mincir 28000   byte increment 875   Adaptive Shaping none
pkts 0      bytes 0          pkts delayed 0      bytes delayed 0
shaping inactive

```

```

traffic shaping drops 0
service policy sample

Service-policy output: sample

Class-map: gold (match-all)
  0 packets, 0 bytes
  5 minute offered rate 0 bps, drop rate 0 BPS
  Match: ip dscp 46
  Weighted Fair Queueing
  Strict Priority
  Output Queue: Conversation 24
  Bandwidth 64 (kbps) Burst 1600 (Bytes)
  (pkts matched/bytes matched) 0/0
  (total drops/bytes drops) 0/0

Class-map: silver (match-all)
  0 packets, 0 bytes
  5 minute offered rate 0 BPS, drop rate 0 BPS
  Match: ip dscp 26
  Weighted Fair Queueing
  Output Queue: Conversation 25
  Bandwidth 32 (kbps) Max Threshold 64 (packets)
!--- Queue information.

  (pkts matched/bytes matched)
  (depth/total drops/no-buffer drops) 0/0/0

Class-map: class-default (match-any)
  0 packets, 0 bytes
  5 minute offered rate 0 BPS, drop rate 0 BPS
  Match: any
  Output queue size 0/max total 600/drops 0
!--- Queue size.

```

Originally, the **frame-relay holdq <size>** map-class command was used to configure the size of FIFO traffic shaping queues only. The maximum size was 512. In Cisco IOS Software Release 12.2, and from IOS Software Release 12.2(4) this command also affects the maximum buffers in CBWFQ traffic shaping queues, as enabled by the **service-policy** output map-class command. The maximum size is now 1024. The defaults, which remain unchanged, are 40 for FIFO and 600 for CBWFQ.

## Interface-Level Queueing

After the Frame Relay frames are enqueued in a PVC queue, they are dequeued to interface-level queues. Traffic from all VCs passes through the interface-level queues.

Depending on the configured features, the Frame Relay interface-level queue uses one of the following mechanisms.

Feature	Default Queueing Mechanism
FRTS	FIFO
FRF.12	Dual FIFO
PIPQ	PIPQ

**Note:** PIPQ (PVC Interface Priority Queueing) overrides FIFO and dual FIFO. In other words, if you enable FRF.12, the interface queueing strategy remains PIPQ.

## FIFO Queueing

The following steps explain how the FRTS configuration alters the applied queueing mechanism to FIFO.

1. Create a channelized interface using the **channel-group** command.

```
Router(config)# controller t1 6/0
Router(config-controller)# channel-group 0 ?
    timeslots  List of timeslots in the channel group

Router(config-controller)# channel-group 0 timeslots ?
    <1-24> List of timeslots which comprise the channel

Router(config-controller)# channel-group 0 timeslots 12
```

2. Execute the **show interface serial 6/0:0** command and confirm the T1 interface is using the default "Queueing strategy: weighted fair". First, a packet is enqueued to a fancy queue at the VC level. It is then sent to the interface queue. In this case, WFQ would be applied.

```
Router# show interface serial 6/0:0
Serial6/0:0 is up, line protocol is up (looped)
  Hardware is Multichannel T1
  MTU 1500 bytes, BW 64 Kbit, DLY 20000 usec,
    reliability 253/255, txload 1/255, rxload 1/255
  Encapsulation HDLC, crc 16, Data non-inverted
  Keepalive set (10 sec)
  Last input 00:00:08, output 00:00:08, output hang never
  Last clearing of "show interface" counters never
  Input queue: 0/75/0/0 (size/max/drops/flushes); Total output drops:
  Queueing strategy: weighted fair
  !--- Queue mechanism.

  Output queue: 0/1000/64/0 (size/max total/threshold/drops)
  !--- Queue size.

  Conversations 0/1/16 (active/max active/max total)
  !--- Queue information.

  Reserved Conversations 0/0 (allocated/max allocated)
  !--- Queue information.

  Available Bandwidth 48 kilobits/sec
  !--- Queue information.

  5 minute input rate 0 bits/sec, 0 packets/sec
  5 minute output rate 0 bits/sec, 0 packets/sec
    5 packets input, 924 bytes, 0 no buffer
    Received 0 broadcasts, 14 runts, 0 giants, 0 throttles
    14 input errors, 0 CRC, 0 frame, 0 overrun, 0 ignored, 0 abort
    17 packets output, 2278 bytes, 0 underruns
    0 output errors, 0 collisions, 0 interface resets
    0 output buffer failures, 0 output buffers swapped out
    0 carrier transitions
  no alarm present
  Timeslot(s) Used:12, subrate: 64Kb/s, transmit delay is 0 flags
```

```
!--- Queue information.
```

- When the queueing strategy is WFQ, you can use the [show queueing](#) and [show queue](#) commands to confirm.

```
Router# show queueing interface serial 6/0:0
Interface Serial6/0:0 queueing strategy: fair
  Input queue: 0/75/0/0 (size/max/drops/flushes); Total output drops: 0
  Queueing strategy: weighted fair
  Output queue: 0/1000/64/0 (size/max total/threshold/drops)
    Conversations 0/1/16 (active/max active/max total)
    Reserved Conversations 0/0 (allocated/max allocated)
    Available Bandwidth 48 kilobits/sec
```

```
Router# show queue serial 6/0:0
  Input queue: 0/75/0/0 (size/max/drops/flushes); Total output drops: 0
  Queueing strategy: weighted fair
  Output queue: 0/1000/64/0 (size/max total/threshold/drops)
    Conversations 0/1/16 (active/max active/max total)
    Reserved Conversations 0/0 (allocated/max allocated)
    Available Bandwidth 48 kilobits/sec
```

- Apply FRTS using the **frame-relay traffic-shaping** command in interface configuration mode.

```
Router(config)# interface serial 6/0:0
Router(config-if)# frame-relay traffic-shaping
```

- Applying FRTS prompts the router to change the queueing strategy on the interface-level queues to FIFO.

```
Router# show interface serial 6/0:0
Serial6/0:0 is up, line protocol is down (looped)
  Hardware is Multichannel T1
  MTU 1500 bytes, BW 64 Kbit, DLY 20000 usec,
    reliability 255/255, txload 1/255, rxload 1/255
  Encapsulation FRAME-RELAY, crc 16, Data non-inverted
  Keepalive set (10 sec)
  LMI enq sent 13, LMI stat recvd 0, LMI upd recvd 0, DTE LMI down
  LMI enq recvd 19, LMI stat sent 0, LMI upd sent 0
  LMI DLCI 1023 LMI type is CISCO frame relay DTE
  Broadcast queue 0/64, broadcasts sent/dropped 0/0, interface broadcas
  Last input 00:00:06, output 00:00:06, output hang never
  Last clearing of "show interface" counters 00:02:16
  Queueing strategy: FIFO      !--- queue mechanism
  Output queue 0/40, 0 drops; input queue 0/75, 0 drops
  5 minute input rate 0 bits/sec, 0 packets/sec
  5 minute output rate 0 bits/sec, 0 packets/sec
    19 packets input, 249 bytes, 0 no buffer
    Received 0 broadcasts, 0 runts, 0 giants, 0 throttles
    0 input errors, 0 CRC, 0 frame, 0 overrun, 0 ignored, 0 abort
    19 packets output, 249 bytes, 0 underruns
    0 output errors, 0 collisions, 0 interface resets
    0 output buffer failures, 0 output buffers swapped out
    0 carrier transitions
  no alarm present
  Timeslot(s) Used:12, subrate: 64Kb/s, transmit delay is 0 flags
```



6. Since the queueing strategy is now FIFO, the **show queue** and **show queueing** command output changes.

```
Router# show queueing interface serial 6/0:0
Interface Serial6/0:0 queueing strategy: none
Router#

Router# show queue serial 6/0:0
'Show queue' not supported with FIFO queueing.
```

Cisco IOS Software Release 12.2(4)T introduces the [Adaptive Frame Relay Traffic Shaping for Interface Congestion](#) feature, which is designed to minimize the effects of delay and packet drops caused by interface congestion. The Adaptive Frame Relay Traffic Shaping for Interface Congestion feature helps ensure that packet drop occurs at the VC queues.

When this new feature is enabled, the traffic-shaping mechanism monitors interface congestion. When the congestion level exceeds a configured value called queue depth, the sending rate of all PVCs is reduced to the minimum committed information rate (minCIR). As soon as interface congestion drops below the queue depth, the traffic-shaping mechanism changes the sending rate of the PVCs back to the committed information rate (CIR). This process guarantees the minCIR for PVCs when there is interface congestion.

## Dual FIFO

Frame Relay Queueing, which appears in the output of the **show interface serial** command as Dual FIFO, uses two priority levels. The high-priority queue handles voice packets and control packets such as Local Management Interface (LMI). The low-priority queue handles fragmented packets (data or non-voice packets).

The interface-level queueing mechanism automatically changes to dual FIFO when you enable one of the following features:

- FRF.12 Fragmentation -- This is enabled with the [frame-relay fragment](#) command in map-class configuration mode. Data packets larger than the packet size specified in the **frame-relay fragment** command are first enqueued to a WFQ subqueue. They are then dequeued and fragmented. After fragmentation, the first segment is transmitted. The remaining segments wait for the next available transmission time for that VC, as determined by the shaping algorithm. At this point, small voice packets and fragmented data packets are interleaved from other PVCs.
- Real-time Transport Protocol (RTP) Prioritization -- Originally, small data packets were also classified as belonging to the high-priority queue simply because of their size. Cisco IOS Software Release 12.0(6)T changed this behavior using the RTP Prioritization (VoIPoFR) feature. It reserves the high-priority queue for voice and LMI control packets only. VoIPoFR classifies VoIP packets by matching on the RTP UDP port range defined in a Frame Relay map-class. All RTP traffic within this port range is enqueued to a priority queue for the VC. In addition, voice packets go into the high priority queue at the interface level. All other packets go into the non-priority queue at the interface level.

**Note:** This functionality assumes that FRF.12 is configured.

Use the show interface command to view the size of the two queues. The steps below show the dual FIFO queues and describe how to change the queue sizes.

1. Execute the **show interface serial** command. The high-priority queue uses a queue limit that is twice the size of the low-priority queue limit.

```

Router# show interface serial 6/0:0
Serial6/0:0 is up, line protocol is down
  Hardware is Multichannel T1
  MTU 1500 bytes, BW 64 Kbit, DLY 20000 usec,
    reliability 255/255, txload 1/255, rxload 1/255
  Encapsulation FRAME-RELAY, crc 16, Data non-inverted
  Keepalive set (10 sec)
  LMI enq sent 236, LMI stat recvd 0, LMI upd recvd 0, DTE LMI down
  LMI enq recvd 353, LMI stat sent 0, LMI upd sent 0
  LMI DLCI 1023 LMI type is CISCO frame relay DTE
  Broadcast queue 0/64, broadcasts sent/dropped 0/0, interface broadcast
  Last input 00:00:02, output 00:00:02, output hang never
  Last clearing of "show interface" counters 00:39:22
  Queueing strategy: dual FIFO!
  --- Queue mechanism.

  Output queue: high size/max/dropped 0/256/0
  !--- High-priority queue.

  Output queue 0/128, 0 drops; input queue 0/75, 0 drops
  !--- Low-priority queue.

  5 minute input rate 0 bits/sec, 0 packets/sec
  5 minute output rate 0 bits/sec, 0 packets/sec
    353 packets input, 4628 bytes, 0 no buffer
    Received 0 broadcasts, 0 runts, 0 giants, 0 throttles
    0 input errors, 0 CRC, 0 frame, 0 overrun, 0 ignored, 0 abort
    353 packets output, 4628 bytes, 0 underruns
    0 output errors, 0 collisions, 0 interface resets
    0 output buffer failures, 0 output buffers swapped out
    0 carrier transitions
  no alarm present
  Timeslot(s) Used:12, subrate: 64Kb/s, transmit delay is 0 flags

```

2. Use the **hold-queue {value} out** command to change the interface queue sizes.

```

Router(config)# interface serial 6/0:0
Router(config-if)# hold-queue ?
  <0-4096> Queue length

Router(config-if)# hold-queue 30 ?
  in   Input queue
  out  Output queue

Router(config-if)# hold-queue 30 out

```

3. Execute the **show interface serial** command again and note how the "Output queue" max values have changed.

```

Router# show interface serial 6/0:0
Serial6/0:0 is up, line protocol is up
  Hardware is Multichannel T1
  MTU 1500 bytes, BW 64 Kbit, DLY 20000 usec,
    reliability 255/255, txload 1/255, rxload 1/255
  Encapsulation FRAME-RELAY, crc 16, Data non-inverted
  Keepalive set (10 sec)
  LMI enq sent 249, LMI stat recvd 0, LMI upd recvd 0, DTE LMI down

```

```

LMI enq recvd 372, LMI stat sent 0, LMI upd sent 0
LMI DLCI 1023 LMI type is CISCO frame relay DTE
Broadcast queue 0/64, broadcasts sent/dropped 0/0, interface broadcas
Last input 00:00:02, output 00:00:02, output hang never
Last clearing of "show interface" counters 00:41:32
Queueing strategy: dual FIFO
!--- Queue mechanism.

Output queue: high size/max/dropped 0/60/0
!--- High-priority queue.

Output queue 0/30, 0 drops; input queue 0/75, 0 drops
!--- Low-priority queue.

5 minute input rate 0 bits/sec, 0 packets/sec
5 minute output rate 0 bits/sec, 0 packets/sec
 372 packets input, 4877 bytes, 0 no buffer
Received 0 broadcasts, 0 runts, 0 giants, 0 throttles
0 input errors, 0 CRC, 0 frame, 0 overrun, 0 ignored, 0 abort
372 packets output, 4877 bytes, 0 underruns
0 output errors, 0 collisions, 0 interface resets
0 output buffer failures, 0 output buffers swapped out
0 carrier transitions
no alarm present
Timeslot(s) Used:12, subrate: 64Kb/s, transmit delay is 0 flags

```

## PIPQ

[Frame-Relay PIPQ](#) is designed for configurations in which separate VCs are carrying a single traffic type, such as voice or data. This allows you to assign a priority value to each PVC. PIPQ minimizes serialization or queueing delay at the interface level by ensuring that the high-priority VC is serviced first. PIPQ classifies packets by extracting the DLCI and looking up the priority in the appropriate PVC structure. The PIPQ mechanism does not look at the packet contents. Therefore, it makes no decisions based on packet contents.

Use the following commands to configure PIPQ.

1. Enable PIPQ with the **frame-relay interface-queue priority** command on the main interface.

```

Router(config)# interface serial 6/0:0
Router(config-if)# frame-relay interface-queue priority
Router(config-if)# end

```

2. Use the **show interface serial** command to confirm "Queueing strategy: DLCI priority". This command also displays the current size and number of drops for each queue.

```

Router# show interface serial 6/0:0
Serial6/0:0 is up, line protocol is up
Hardware is Multichannel T1
MTU 1500 bytes, BW 64 Kbit, DLY 20000 usec,
  reliability 255/255, txload 1/255, rxload 1/255
Encapsulation FRAME-RELAY, crc 16, Data non-inverted
Keepalive set (10 sec)
LMI enq sent 119, LMI stat recvd 0, LMI upd recvd 0, DTE LMI down
LMI enq recvd 179, LMI stat sent 0, LMI upd sent 0
LMI DLCI 1023 LMI type is CISCO frame relay DTE
Broadcast queue 0/64, broadcasts sent/dropped 0/0, interface broadcas
Last input 00:00:06, output 00:00:06, output hang never

```

```

Last clearing of "show interface" counters 00:19:56
Input queue: 0/75/0/0 (size/max/drops/flushes); Total output drops: 0
Queueing strategy: DLCI priority
!--- Queue mechanism.

Output queue (queue priority: size/max/drops):
  high: 0/20/0, medium: 0/40/0, normal: 0/60/0, low: 0/80/0
!--- Queue size.

5 minute input rate 0 bits/sec, 0 packets/sec
5 minute output rate 0 bits/sec, 0 packets/sec
 179 packets input, 2347 bytes, 0 no buffer
  Received 0 broadcasts, 0 runts, 0 giants, 0 throttles
  0 input errors, 0 CRC, 0 frame, 0 overrun, 0 ignored, 0 abort
 179 packets output, 2347 bytes, 0 underruns
  0 output errors, 0 collisions, 0 interface resets
  0 output buffer failures, 0 output buffers swapped out
  0 carrier transitions
no alarm present
Timeslot(s) Used:12, subrate: 64Kb/s, transmit delay is 0 flags

```

3. Build a Frame Relay map-class and assign a priority level to a VC using the command **frame-relay interface-queue priority {high|medium|normal|low}**. The default PVC priority is normal. All PVCs at the same priority share the same FIFO priority queue. Apply the map-class to the VC. In the following sample output, a PVC with DLCI number 21 is assigned to the high-priority interface queue.

```

Router(config)# map-class frame-relay high_priority_class
Router(config-map-class)# frame-relay interface-queue priority high
Router(config-map-class)# exit

Router(config)# interface serial 6/0:0.2 point
Router(config-subif)# frame-relay interface-dlci 21
Router(config-fr-dlci)# class ?
WORD map class name

Router(config-fr-dlci)# class high_priority_class

```

4. Use the **show frame-relay PVC [dlci]** and **show queueing interface** commands to confirm your configuration change.

```

Router# show frame PVC 21

PVC Statistics for interface Serial6/0:0 (Frame Relay DTE)

DLCI = 21, DLCI USAGE = LOCAL, PVC STATUS = INACTIVE, INTERFACE = Serial6/0:0.2

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 00:00:17, last time PVC status changed 00:00:17
cir 56000      BC 7000      be 0          byte limit 875      interval 1
mincir 28000   byte increment 875  Adaptive Shaping none
pkts 0          bytes 0          pkts delayed 0          bytes delayed
shaping inactive
traffic shaping drops 0
Queueing strategy: FIFO
Output queue 0/40, 0 drop, 0 dequeued

```

```

!--- Size of the PVC queue.

priority high

!--- All frames from this PVC are dequeued to the high-priority queue

!--- at the interface.

Router# show queueing interface serial 6/0:0
Interface Serial6/0:0 queueing strategy: priority

Output queue utilization (queue/count)
      high/13 medium/0 normal/162 low/0

```

- Optionally, configure the size of each interface queue with the following command. The default sizes of the high, medium, normal, and low priority queues are 20, 40, 60, and 80 packets, respectively. To configure a different value, use the command `frame-relay interface-queue priority [high limit<medium limit><normal limit><low limit>]` in the interface configuration mode.

Once enabled, PIPQ overrides any other Frame Relay interface queueing mechanisms, including dual FIFO. If you subsequently enable FRF.12 or FRTS, the interface-level queueing mechanism will not revert to dual FIFO. In addition, PIPQ cannot be enabled if a non-default fancy queueing mechanism already is configured on the interface. It can be enabled in the presence of WFQ if WFQ is the default interface queueing method. Deleting the PIPQ configuration changes the interface-level queueing to the default or to dual FIFO, if FRF.12 is enabled.

PIPQ applies strict priority queueing. If traffic is continuously dequeued to the high-priority queue, the queueing scheduler will schedule the high-priority queue and may effectively starve lower-priority queues. Therefore, take care in assigning PVCs to the high-priority queue.

## Tuning the TX Ring

The TX ring is the unprioritized FIFO buffer used to store frames before transmission. Frame Relay interfaces use a single TX ring that is shared by all VCs. By default, the TX ring size is 64 packets for higher-speed serial WAN interfaces, including the PA-T3+, PA-MC-2T3+, and PA-H. Lower-speed WAN port adapters now automatically tune down the TX ring to a value of 2 packets. In other words, interface drivers set unique default TX ring values based on the bandwidth amount.

Queue	Location	Queueing Methods	Service Policies Apply	Command to Tune
Hardware queue or transmit ring per interface	Port adapter or network module	FIFO only	No	<a href="#">tx-ring-limit</a>
				Varies with

Layer 3 queue per VC	Layer 3 processor system or interface buffers	FIFO, WFQ, CBWFQ, or LLQ	Yes	queuing method: <ul style="list-style-type: none"> <li>• <b>frame-relay holdq</b> with FIFO</li> <li>• <b>queue-limit</b> with CBWFQ</li> </ul>
----------------------	---	--------------------------	-----	---

**Note:** Unlike ATM interfaces such as the PA-A3, Frame Relay interfaces use a single transmit ring for the interface. They do not build a separate ring for each VC.

It is important to know that the TX ring is FIFO and cannot support an alternate queueing mechanism. Thus, tuning down the TX ring to a value of 2 on low-speed interfaces effectively moves most of the packet buffering to the PVC queue where the fancy queueing mechanisms and QoS service policies do apply.

The following table lists serial port adapters for the 7x00 Series for automatic tuning down of the transmit ring.

Port Adapter Part #	TX Ring Limit Auto Tuning
<b>High-Speed Serial Port Adapters</b>	
PA-H and PA-2H	Yes
PA-E3 and PA-T3	Yes
PA-T3+	Yes
<b>Multichannel Serial Port Adapters</b>	
PA-MC-2T3+	Yes
PA-MC-2T1(=), PA-MC-4T1(=), PA-MC-8T1(=), PA-MC-8DSX1(=)	Yes
PA-MC-2E1/120(=), PA-MC-8E1/120(=)	Yes
PA-MC-T3, PA-MC-E3	Yes
PA-MC-8TE1+	Yes
PA-STM1	Yes
<b>Serial Port Adapters</b>	
PA-4T, PA-4T+	Yes
PA-4E1G	Yes
PA-8T-V35, PA-8T-X21, PA-8T-232	Yes

The size of the transmit ring is tuned down automatically when a voice-optimizing feature is enabled. In addition, applying PIPQ causes the transmit ring to be tuned down automatically.

The following output was captured on a 7200 Series router running Cisco IOS Software Release 12.2 (6).

```
7200-16# show controller serial 6/0:0
Interface Serial6/0:0
  f/w rev 1.2.3, h/w rev 163, PMC freedm rev 1 idb = 0x6382B984
  ds = 0x62F87C18, plx_devbase=0x3F020000, pmc_devbase=0x3F000000
  Enabled=TRUE, DSX1 linestate=0x0,
  Ds>tx_limited:1 Ds>tx_count:0 Ds>max_tx_count:20

  alarm present
  Timeslot(s) Used:1-24, subrate: 64Kb/s, transmit delay is 0 flags
  Download delay = 0, Report delay = 0
  IDB type=0xC, status=0x84208080
  Pci shared memory = 0x4B16B200
  Plx mailbox addr = 0x3F020040
  RxFree queue=0x4B2FA280, shadow=0x62F9FA70
  Rx freeq_wt=256, freeq_rd=256, ready_wt=1, ready_rd=0
  TxFree queue=0x4B2FAAC0, shadow=0x62F8FA44
  TX freeq_wt=4099, freeq_rd=4099, ready_wt=4, ready_rd=3
  # of TxFree queue=4095
  Freedm FIFO (0x6292BF64), hp=0x6292C034 indx=26, tp=0x6292CF5C indx=511
  reset_count=0 resurrect_count=0
  TX enqueued=0, throttled=0, unthrottled=0, started=10
  tx_limited=TRUE tx_queue_limit=2
```

*!--- Note "tx\_limited=TRUE" when PIPQ is enabled. The "tx\_queue\_limit" value*

*!--- describes the value of the transmit ring.*

```
7200-16(config)# interface serial 6/0:0
7200-16(config-if)# no frame-relay interface-queue priority
7200-16(config-if)# end
7200-16# show controller serial 6/0:0
Interface Serial6/0:0
  f/w rev 1.2.3, h/w rev 163, PMC freedm rev 1 idb = 0x6382B984
  Ds = 0x62F87C18, plx_devbase=0x3F020000, pmc_devbase=0x3F000000
  Enabled=TRUE, DSX1 linestate=0x0,
  Ds>tx_limited:0 Ds>tx_count:0 Ds>max_tx_count:20

  alarm present
  Timeslot(s) Used:1-24, subrate: 64Kb/s, transmit delay is 0 flags
  Download delay = 0, Report delay = 0
  IDB type=0xC, status=0x84208080
  Pci shared memory = 0x4B16B200
  Plx mailbox addr = 0x3F020040
  RxFree queue=0x4B2FA280, shadow=0x62F9FA70
  Rx freeq_wt=256, freeq_rd=256, ready_wt=1, ready_rd=0
  TxFree queue=0x4B2FAAC0, shadow=0x62F8FA44
  TX freeq_wt=4099, freeq_rd=4099, ready_wt=4, ready_rd=3
  # of TxFree queue=4095
  Freedm FIFO (0x6292BF64), hp=0x6292C034 indx=26, tp=0x6292CF5C indx=511
  reset_count=0 resurrect_count=0
  TX enqueued=0, throttled=0, unthrottled=0, started=11
  tx_limited=FALSE
```

*!--- Transmit ring value has changed.*



## NetPro Discussion Forums - Featured Conversations

Networking Professionals Connection is a forum for networking professionals to share questions, suggestions, and information about networking solutions, products, and technologies. The featured links are some of the most recent conversations available in this technology.

NetPro Discussion Forums - Featured Conversations for RP
Service Providers: MPLS
<a href="#">Bundle L2 EOMPLS VPN</a> - Nov 17, 2005 <a href="#">Need Application based LSPs. L4VPN?</a> - Nov 16, 2005 <a href="#">tag-swithing mtu 1600 results in giants</a> - Nov 16, 2005 <a href="#">SNMP access to interface in VRF mode</a> - Nov 14, 2005 <a href="#">MPLS in CCIE Service Provider LAB Exam</a> - Nov 14, 2005
Virtual Private Networks: Services
<a href="#">VPN Connection slow</a> - Nov 16, 2005 <a href="#">Lan-to-lan VPN with voice. One G.729 call is using over 40-50kbps</a> - Nov 15, 2005 <a href="#">VOIP QoS VPN examples</a> - Nov 10, 2005 <a href="#">EZVPN and QoS for VOIP</a> - Nov 9, 2005 <a href="#">PPTP with NAT</a> - Nov 7, 2005
Virtual Private Networks: Security
<a href="#">Site to Site VPN with third site via Frame Relay</a> - Nov 17, 2005 <a href="#">CSACS SQL integration</a> - Nov 17, 2005 <a href="#">FWSM and VPNSM in same 6509 chassis problem.</a> - Nov 17, 2005 <a href="#">No split-tunnel for software vpn termination on cisco IOS router</a> - Nov 17, 2005 <a href="#">nmap reports open port 25/110 on Cisco devices</a> - Nov 17, 2005

## Related Information

- [Configuring CBWFQ on Frame Relay PVCs](#)
- [Low Latency Queueing for Frame Relay](#)
- [Frame Relay PVC Interface Priority Queueing](#)
- [Configuring Frame Relay Traffic Shaping on 7200 Routers and Lower Platforms](#)
- [Frame Relay Traffic Shaping With Distributed QoS on the Cisco 7500 Series](#)
- [Configuring Packet Marking on Frame Relay PVCs](#)
- [Low Latency Queueing for Frame Relay](#)
- [Frame Relay Support Pages](#)
- [QoS Support Pages](#)
- [Technical Support - Cisco Systems](#)

<a href="#">Home</a>	<a href="#">How to Buy</a>	<a href="#">Login</a>	<a href="#">Register</a>	<a href="#">Feedback</a>	<a href="#">Site Map</a>	<a href="#">Help</a>
----------------------	----------------------------	-----------------------	--------------------------	--------------------------	--------------------------	----------------------

All contents are Copyright © 1992-2005 Cisco Systems, Inc. All rights reserved. [Important Notices](#) and [Privacy Statement](#).

Updated: Aug 10, 2005

Document ID: 22304