



Frame Relay PVC Interface Priority Queueing

This feature module describes the Frame Relay PVC Interface Priority Queueing (FR PIPQ) feature. It includes information on the benefits of this new feature, supported platforms, related documents, and so on.

This document includes the following sections:

- Feature Overview, page 1
- Supported Platforms, page 3
- Supported Standards, MIBs, and RFCs, page 3
- Prerequisites, page 3
- Configuration Tasks, page 4
- Monitoring and Maintaining FR PIPQ, page 5
- Configuration Examples, page 6
- Command Reference, page 7
- Debug Commands, page 38
- Glossary, page 40

Feature Overview

The FR PIPQ feature provides an interface-level priority queueing scheme in which prioritization is based on destination permanent virtual circuit (PVC) rather than packet contents. For example, FR PIPQ allows you to configure a PVC transporting voice traffic to have absolute priority over a PVC transporting signalling traffic, and a PVC transporting signalling traffic to have absolute priority over a PVC transporting data.

FR PIPQ provides four levels of priority: high, medium, normal, and low. The Frame Relay packet is examined at the interface for the data-link connection identifier (DLCI) value. The packet is then sent to the correct priority queue based on the priority level configured for that DLCI.



Note

When using FR PIPQ, configure the network so that different types of traffic are transported on separate PVCs. FR PIPQ is not meant to be used when an individual PVC carries different traffic types that have different quality of service (QoS) requirements.

You assign priority to a PVC within a Frame Relay map class. All PVCs using or inheriting that map class will be classed according to the configured priority. If a PVC does not have a map class associated with it, or if the map class associated with it does not have priority explicitly configured, then the packets on that PVC will be queued on the default “normal” priority queue.

If you do not enable FR PIPQ on the interface using the **frame-relay interface-queue priority** command in interface configuration mode, configuring PVC priority within a map class will not be effective. At this time you have the option to also set the size (in maximum number of packets) of the four priority queues.

FR PIPQ works with or without Frame Relay traffic shaping (FRTS) and FRF.12. The interface-level priority queueing takes the place of the FIFO queueing or dual FIFO queueing normally used by FRTS and FRF.12. PVC priority assigned within FR PIPQ takes precedence over FRF.12 priority, which means that all packets destined for the same PVC will be queued on the same interface queue whether they were fragmented or not.

**Note**

Although high priority PVCs most likely will transport only small packets of voice traffic, you may want to configure FRF.12 on these PVCs anyway to guard against any unexpectedly large packets.

Benefits

FR PIPQ provides four levels of PVC priority: high, medium, normal, and low. This method of queueing ensures that time/delay-sensitive traffic such as voice has absolute priority over signalling traffic, and that signalling traffic has absolute priority over data traffic, providing different PVCs are used for the different types of traffic.

Restrictions

The following restrictions apply to FR PIPQ:

- FR PIPQ is not supported on loopback or tunnel interfaces, or interfaces that explicitly disallow priority queueing.
- FR PIPQ is not supported with hardware compression.
- FR PIPQ cannot be enabled on an interface that is already configured with queueing other than FIFO queueing. FR PIPQ can be enabled if WFQ is configured, as long as WFQ is the default interface queueing method.

Related Features and Technologies

The following features and technologies are related to FR PIPQ:

- FRTS
- FRF.12

Related Documents

The following documents provide information related to FR PIPQ:

- *Cisco IOS Wide-Area Networking Configuration Guide*, Release 12.1
- *Cisco IOS Wide-Area Networking Command Reference*, Release 12.1

Supported Platforms

- Cisco 1000 series
- Cisco 1400 series
- Cisco 1600
- Cisco 1700
- Cisco 1750
- Cisco 2500
- Cisco 2600
- Cisco 3600
- Cisco 3810
- Cisco 4500
- Cisco 4700
- Cisco 7200
- Cisco 7500 (in nondistributed mode)

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 CCO at <http://www.cisco.com/public/sw-center/netmgmt/cmtk/mibs.shtml>.

RFCs

No new or modified standards are supported by this feature.

Prerequisites

The following prerequisites apply to FR PIPQ:

- PVCs should be configured to carry a single type of traffic.
- The network should be configured with adequate call admission control to prevent starvation of any of the priority queues.

Configuration Tasks

See the following sections for configuration tasks for the FR PIPQ feature. Each task in the list is identified as either optional or required:

- Configuring PVC Priority in a Map Class (Required)
- Enabling FR PIPQ and Setting Queue Limits (Required)
- Assigning a Map Class to a PVC (Required)

Configuring PVC Priority in a Map Class

To configure PVC priority within a map class, use the following commands beginning in global configuration mode:

	Command	Purpose
Step 1	Router(config)# map-class frame-relay <i>map-class-name</i>	Specifies a Frame Relay map class.
Step 2	Router(config-map-class)# frame-relay interface-queue priority { <i>high medium normal low</i> }	Assigns a PVC priority level to a Frame Relay map class.

Enabling FR PIPQ and Setting Queue Limits

To enable FR PIPQ and set the priority queue sizes, use the following commands beginning in global configuration mode:

	Command	Purpose
Step 1	Router(config)# interface <i>type number</i> [<i>name-tag</i>]	Configures an interface type and enters interface configuration mode.
Step 2	Router(config-if)# encapsulation frame-relay [<i>cisco ietf</i>]	Enables Frame Relay encapsulation.
Step 3	Router(config-if)# frame-relay interface-queue priority [<i>high-limit medium-limit normal-limit low-limit</i>]	Enables FR PIPQ and sets the priority queue limits.

Assigning a Map Class to a PVC

To assign a map class to a specific PVC, use the following commands beginning in interface configuration mode:

	Command	Purpose
Step 1	Router(config-if)# frame-relay interface-dlci <i>dlci</i>	Specifies a single PVC on a Frame Relay interface.
Step 2	Router(config-fr-dlci)# class <i>map-class-name</i>	Associates a map class with a specified PVC.

Verifying FR PIPQ

To verify the configuration of FR PIPQ, use one or more of the following commands in privileged EXEC mode:

Command	Purpose
Router# show frame-relay pvc [<i>interface interface</i>] [<i>dlci</i>]	Displays statistics about PVCs for Frame Relay interfaces.
Router# show interfaces [<i>type number</i>] [<i>first</i>] [<i>last</i>]	Displays the statistical information specific to a serial interface.
Router# show queueing [<i>custom</i> <i>fair</i> <i>priority</i> <i>random-detect</i>] [<i>interface atm_subinterface</i> [<i>vc</i> [[<i>vpi/</i>] <i>vci</i>]]]	Lists all or selected configured queueing strategies.

Monitoring and Maintaining FR PIPQ

To monitor and maintain FR PIPQ, use one or more of the following commands in privileged EXEC mode:

Command	Purpose
Router# debug priority	Debugs priority output queueing.
Router# show frame-relay pvc [<i>interface interface</i>] [<i>dlci</i>]	Displays statistics about PVCs for Frame Relay interfaces.
Router# show interfaces [<i>type number</i>] [<i>first</i>] [<i>last</i>]	Displays the statistical information specific to a serial interface.
Router# show queue <i>interface-name interface-number</i> [<i>vc</i> [<i>vpi/</i>] <i>vci</i>] [<i>queue-number</i>]	Displays the contents of packets inside a queue for a particular interface or VC.
Router# show queueing [<i>custom</i> <i>fair</i> <i>priority</i> <i>random-detect</i>] [<i>interface atm_subinterface</i> [<i>vc</i> [[<i>vpi/</i>] <i>vci</i>]]]	Lists all or selected configured queueing strategies.

Configuration Examples

This section provides configuration examples for FR PIPQ.

FR PIPQ Configuration Example

This example shows the configuration of four PVCs on serial interface 0. DLCI 100 is assigned high priority, DLCI 200 is assigned medium priority, DLCI 300 is assigned normal priority, and DLCI 400 is assigned low priority.

The following commands configure Frame Relay map classes with PVC priority levels:

```
Router(config)# map-class frame-relay HI
Router(config-map-class)# frame-relay interface-queue priority high
Router(config-map-class)# exit
Router(config)# map-class frame-relay MED
Router(config-map-class)# frame-relay interface-queue priority medium
Router(config-map-class)# exit
Router(config)# map-class frame-relay NORM
Router(config-map-class)# frame-relay interface-queue priority normal
Router(config-map-class)# exit
Router(config)# map-class frame-relay LOW
Router(config-map-class)# frame-relay interface-queue priority low
Router(config-map-class)# exit
```

The following commands enable Frame Relay encapsulation and FR PIPQ on serial interface 0. The sizes of the priority queues are set at a maximum of 20 packets for the high priority queue, 40 for the medium priority queue, 60 for the normal priority queue, and 80 for the low priority queue.

```
Router(config)# interface Serial0
Router(config-if)# encapsulation frame-relay
Router(config-if)# frame-relay interface-queue priority 20 40 60 80
```

The following commands assign priority to four PVCs by associating the DLCIs with the configured map classes:

```
Router(config-if)# frame-relay interface-dlci 100
Router(config-fr-dlci)# class HI
Router(config-fr-dlci)# exit
Router(config-if)# frame-relay interface-dlci 200
Router(config-fr-dlci)# class MED
Router(config-fr-dlci)# exit
Router(config-if)# frame-relay interface-dlci 300
Router(config-fr-dlci)# class NORM
Router(config-fr-dlci)# exit
Router(config-if)# frame-relay interface-dlci 400
Router(config-fr-dlci)# class LOW
Router(config-fr-dlci)# exit
```

Command Reference

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

- **frame-relay interface-queue priority**
- **show frame-relay pvc**
- **show interfaces**
- **show queueing**

frame-relay interface-queue priority

To enable the Frame Relay PVC Interface Priority Queuing (FR PIPQ) feature, use the **frame-relay interface-queue priority** interface configuration command. To disable FR PIPQ, use the **no** form of this command.

frame-relay interface-queue priority [*high-limit medium-limit normal-limit low-limit*]

no frame-relay interface-queue priority

To assign priority to a permanent virtual circuit (PVC) within a Frame Relay map class, use the **frame-relay interface-queue priority** map-class configuration command. To remove priority from a PVC within a Frame Relay map class, use the **no** form of this command.

frame-relay interface-queue priority {**high** / **medium** / **normal** / **low**}

no frame-relay interface-queue priority

Syntax Description		
	<i>high-limit</i>	Size of the high priority queue specified in maximum number of packets.
	<i>medium-limit</i>	Size of the medium priority queue specified in maximum number of packets.
	<i>normal-limit</i>	Size of the normal priority queue specified in maximum number of packets.
	<i>low-limit</i>	Size of the low priority queue specified in maximum number of packets.
	high	Assigns high priority to a PVC.
	medium	Assigns medium priority to a PVC.
	normal	Assigns normal priority to a PVC.
	low	Assigns low priority to a PVC.

Defaults

The default sizes of the high, medium, normal, and low priority queues are 20, 40, 60, and 80 packets, respectively.

When FR PIPQ is enabled on the interface, the default PVC priority is normal priority.

Command Modes

Interface configuration

Map-class configuration

Command History

Release	Modification
12.1(2)T	This command was introduced.

Usage Guidelines

FR PIPQ must be enabled on the interface in order for the map-class configuration of PVC priority to be effective.

Before you configure FR PIPQ using the **frame-relay interface-queue priority** command, the following conditions must be met:

- PVCs should be configured to carry a single type of traffic.
- The network should be configured with adequate call admission control to prevent starvation of any of the priority queues.

You will not be able to configure FR PIPQ if any queueing other than FIFO queueing is already configured at the interface level. You will be able to configure FR PIPQ when weighted fair queueing (WFQ) is in use, as long as WFQ is the default interface queueing method. Disabling FR PIPQ will restore the interface to dual FIFO queueing if FRF.12 is enabled, FIFO queueing if Frame Relay traffic shaping (FRTS) is enabled, or the default queueing method for the interface.

Examples

In the following example, FR PIPQ is enabled on serial interface 0, and the limits of the high, medium, normal, and low priority queues are set to 10, 20, 30, and 40 packets, respectively. PVC 100 is assigned high priority, so all traffic destined for PVC 100 will be sent to the high priority interface queue.

```
interface serial0
  encapsulation frame-relay
  frame-relay interface-queue priority 10 20 30 40
  frame-relay interface-dlci 100
    class high_priority_class
  !
  map-class frame-relay high_priority_class
  frame-relay interface-queue priority high
```

Related Commands

Command	Description
debug priority	Displays priority queueing events.
show frame-relay pvc	Displays statistics about PVCs for Frame Relay interfaces.
show interfaces	Displays statistics for all interfaces configured on the router or access server.
show queue	Displays the contents of packets inside a queue for a particular interface or VC.
show queueing	Lists all or selected configured queueing strategies.

show frame-relay pvc

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

```
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 data-link connection identifiers (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. The command was also modified to include information about the priority configured for a PVC within FR PIPQ.

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 “voifr” or “voifr 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 or the priority configured for that PVC, use this command with the *dldci* 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.

Frame Relay PVC Priority Queueing Example

The following is sample output for a PVC that has been assigned high priority:

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

PVC Statistics for interface Serial0 (Frame Relay DTE)

DLCI = 100, DLCI USAGE = LOCAL, PVC STATUS = ACTIVE, INTERFACE = Serial0

  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 00:00:59, last time pvc status changed 00:00:33
  priority high
```

Low Latency Queueing for Frame Relay Example

The following is sample output from the **show frame-relay pvc** command for a PVC shaped to a 64K committed information rate (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. Weighted Random Early Detection (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
```

■ show frame-relay pvc

```

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 are 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 (VoFR) traffic 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** command 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.12 header
- end-to-end—Indicates that pure FRF.12 fragmentation is carried on this virtual circuit

The following 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 Serial1 (Frame Relay DTE)

DLCI = 45, DLCI USAGE = LOCAL, PVC STATUS = STATIC, INTERFACE = Serial1

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 bcst bytes 0
out bcst pkts 0       out bcst 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 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 device.
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.

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

Field	Description
in DE pkts	Number of DE packets received.
out DE pkts	Number of DE packets sent.
out bcast pkts	Number of output broadcast packets.
out bcast 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).
priority	Priority assigned to the PVC.
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 (bps) reserved for voice traffic on this PVC.
used voice bandwidth	Amount of bandwidth in bps 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 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.
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.

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

Field	Description
Maximum Number of Hashed Queues	(Applies to class default only) Number of queues available for unclassified flows.
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 CIR in bps.
bc	Current Committed Burst (Bc) size in bits.
be	Current Excess Burst (Be) size in bits.
limit	Maximum number of bytes sent 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 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 sent.
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. Additional packets received after a queue is full 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.

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

Field	Description
max total	Maximum number of packets of all types that can be queued in all queues.
drops	Number of frames dropped by all output queues.

Related Commands

Command	Description
frame-relay interface-queue priority	Enables FR PIPQ on a Frame Relay interface, and assigns priority to a PVC within a Frame Relay map class.
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 VoFR 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 interfaces

To display statistics for all interfaces configured on the router or access server, use the **show interfaces** privileged EXEC command. The resulting output varies, depending on the network for which an interface has been configured.

show interfaces [*type number*] [*first*] [*last*] [**accounting**]

Cisco 7200 Series and Cisco 7500 Series with a Packet over SONET Interface Processor

show interfaces [*type slot/port*] [**accounting**]

Cisco 7500 Series with Ports on VIPs

show interfaces [*type slot/port-adapter/port*] [**ethernet** | **serial**]

Syntax Description	
<i>type</i>	(Optional) Interface type. Allowed values for the <i>type</i> argument are async , bri0 , dialer , ethernet , fastethernet , fdi , hssi , loopback , null , serial , tokenring , and tunnel . For the Cisco 4000 series routers, the <i>type</i> argument can be e1 , ethernet , fastethernet , fdi , serial , t1 , and token . For the Cisco 4500 series routers, the <i>type</i> argument can also include atm . For the Cisco 7000 family, the <i>type</i> argument can be atm , e1 , ethernet , fastethernet , fdi , serial , t1 , and tokenring . For the Cisco 7500 series, the <i>type</i> argument can also include pos .
<i>number</i>	(Optional) Port number on the selected interface.
<i>first last</i>	(Optional) For Cisco 2500 and 3000 series routers, ISDN BRI only. The <i>first</i> argument can be either 1 or 2. The <i>last</i> argument can only be 2, indicating B channels 1 and 2. D-channel information is obtained by using the command without the optional arguments.
accounting	(Optional) Displays the number of packets of each protocol type that have been sent through the interface.
<i>slot</i>	(Optional) Number of the slot being configured. Refer to the appropriate hardware manual for slot and port information.
<i>port</i>	(Optional) Number of the port being configured. Refer to the appropriate hardware manual for slot and port information.
<i>port-adapter</i>	(Optional) Number of the port adapter being configured. Refer to the appropriate hardware manual for information about port adapter compatibility.

Command Modes Privileged EXEC

Command History	Release	Modification
	10.0	This command was introduced.
	12.0(3)T	This command was modified to include support for flow-based Weighted Random Early Detection (WRED).
	12.0(4)T	This command was modified to include enhanced display information for dialer bound interfaces.
	12.0(7)T	This command was modified to include the dialer keyword as an interface type, and to reflect the default behavior.
	12.1(2)T	This command was modified to display information about the priority queues within the Frame Relay PVC Interface Priority Queueing (FR PIPQ) feature.

Usage Guidelines

The **show interfaces** command displays statistics for the network interfaces. The resulting display on the Cisco 7200 series routers shows the interface processors in slot order. If you add interface processors after booting the system, they will appear at the end of the list, in the order in which they were inserted.

If you use the **show interfaces** command on the Cisco 7200 series routers without the *slot/port* arguments, information for all interface types will be shown. For example, if you enter the **show interfaces ethernet** command, you will receive information for all Ethernet, serial, Token Ring, and FDDI interfaces. Only by adding the *type slot/port* argument can you specify a particular interface.

If you enter a **show interfaces** command for an interface type that has been removed from the router or access server, interface statistics will be displayed accompanied by the following text: “Hardware has been removed.”

If you use the **show interfaces** command on a router or access server for which interfaces are configured to use weighted fair queueing (WFQ) through the **fair-queue** interface configuration command, additional information is displayed. This information consists of the current and high-water mark number of flows.

If you use the **show interfaces** command on dialer interfaces configured for binding, the display will report statistics on each physical interface bound to the dialer interface; see the examples in the following section for more information.

You will use the **show interfaces** command frequently while configuring and monitoring devices. The various forms of the **show interfaces** commands are shown in detail in the “Examples” section of this command.

Examples

The following is sample output from the **show interfaces** command. Because your display will depend on the type and number of interface cards in your router or access server, only a portion of the display is shown.

```
Router# show interfaces

Ethernet 0 is up, line protocol is up
  Hardware is MCI Ethernet, address is 0000.0c00.750c (bia 0000.0c00.750c)
  Internet address is 131.108.28.8, subnet mask is 255.255.255.0
  MTU 1500 bytes, BW 10000 Kbit, DLY 100000 usec, rely 255/255, load 1/255
  Encapsulation ARPA, loopback not set, keepalive set (10 sec)
  ARP type: ARPA, ARP Timeout 4:00:00
  Last input 0:00:00, output 0:00:00, output hang never
  Last clearing of "show interface" counters 0:00:00
  Output queue 0/40, 0 drops; input queue 0/75, 0 drops
  Five minute input rate 0 bits/sec, 0 packets/sec

  Five minute output rate 2000 bits/sec, 4 packets/sec
    1127576 packets input, 447251251 bytes, 0 no buffer
    Received 354125 broadcasts, 0 runts, 0 giants
    0 input errors, 0 CRC, 0 frame, 0 overrun, 0 ignored, 0 abort
    5332142 packets output, 496316039 bytes, 0 underruns
    0 output errors, 432 collisions, 0 interface resets, 0 restarts
  .
  .
  .
```

Frame Relay PVC Interface Priority Queueing Example

The following is sample output from the **show interfaces** command when FR PIPQ is enabled on serial interface 0:

```
Router# show interfaces Serial0

Serial3/2 is up, line protocol is up
  Hardware is M4T
  MTU 1500 bytes, BW 2048 Kbit, DLY 20000 usec,
    reliability 255/255, txload 1/255, rxload 1/255
  Encapsulation FRAME-RELAY, crc 16, loopback not set
  Keepalive set (10 sec)
  LMI enq sent 7, LMI stat recvd 7, LMI upd recvd 0, DTE LMI up
  LMI enq recvd 0, LMI stat sent 0, LMI upd sent 0
  LMI DLCI 0 LMI type is ANSI Annex D frame relay DTE
  FR SVC disabled, LAPF state down
  Broadcast queue 0/64, broadcasts sent/dropped 0/0, interface broadcasts 0
  Last input 00:00:08, output 00:00:08, output hang never
  Last clearing of "show interface" counters 00:01:31
  Input queue:0/75/0 (size/max/drops); Total output drops:0
  Queueing strategy:DLCI priority
  Output queue (queue priority:size/max/drops):
    high:0/20/0, medium:0/40/0, normal:0/60/0, low:0/80/0
  5 minute input rate 0 bits/sec, 0 packets/sec
  5 minute output rate 0 bits/sec, 0 packets/sec
    7 packets input, 173 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
    7 packets output, 98 bytes, 0 underruns
    0 output errors, 0 collisions, 2 interface resets
    0 output buffer failures, 0 output buffers swapped out
    2 carrier transitions      DCD=up DSR=up DTR=up RTS=up CTS=up
```

Table 2 shows the output fields relevant to FR PIPQ.

Table 2 *show interfaces Field Descriptions (When FR PIPQ Is Enabled)*

Field	Description
Queueing strategy	Queueing strategy configured on the interface. Displays “DLCI priority” when FR PIPQ is enabled.
Output queue	Queue for which statistics will be displayed.
queue priority:	Priority queue for which statistics will be displayed.
size	Number of packets currently in the queue.
max	Maximum number of packets the queue will hold before it begins dropping packets.
drops	Number of packets dropped.

Custom Output Queueing Example

The following is partial sample output when custom output queueing is enabled:

```
Last clearing of "show interface" counters 0:00:06
Input queue: 0/75/0 (size/max/drops); Total output drops: 21
Output queues: (queue #: size/max/drops)
    0: 14/20/14  1: 0/20/6  2: 0/20/0  3: 0/20/0  4: 0/20/0  5: 0/20/0
    6: 0/20/0  7: 0/20/0  8: 0/20/0  9: 0/20/0 10: 0/20/0
```

When custom queueing is enabled, the drops accounted for in the output queues result from bandwidth limitation for the associated traffic and lead to queue length overflow. Total output drops include drops on all custom queues and the system queue. Fields are described with the weighted fair queueing (WFQ) output in Table 3.

WFQ Output Example

In the following sample output, each interface on the router or access server configured to use WFQ, the **show interfaces** command displays the information beginning with the “Input queue:” line:

```
Router# show interfaces

Ethernet 0 is up, line protocol is up
  Hardware is MCI Ethernet, address is 0000.0c00.750c (bia 0000.0c00.750c)
  Internet address is 131.108.28.8, subnet mask is 255.255.255.0
  MTU 1500 bytes, BW 10000 Kbit, DLY 100000 usec, rely 255/255, load 1/255
  Encapsulation ARPA, loopback not set, keepalive set (10 sec)
  ARP type: ARPA, ARP Timeout 4:00:00
  Last input 0:00:00, output 0:00:00, output hang never
  Last clearing of "show interface" counters 0:00:00
  Output queue 0/40, 0 drops; input queue 0/75, 0 drops
  Five minute input rate 0 bits/sec, 0 packets/sec
  Five minute output rate 2000 bits/sec, 4 packets/sec
    1127576 packets input, 447251251 bytes, 0 no buffer
    Received 354125 broadcasts, 0 runts, 0 giants
    0 input errors, 0 CRC, 0 frame, 0 overrun, 0 ignored, 0 abort
    5332142 packets output, 496316039 bytes, 0 underruns
    0 output errors, 432 collisions, 0 interface resets, 0 restarts
  Input queue: 0/75/0 (size/max/drops); Total output drops: 0
  Output queue: 7/64/0 (size/threshold/drops)
    Conversations 2/9 (active/max active)
```

Table 3 provides a listing of input queue and output queue fields shown in the preceding display.

Table 3 *WFQ Output Field Descriptions*

Field	Description
Input queue	
size	Current size of the input queue.
max	Maximum size of the queue.
drops	Number of messages discarded in this interval.
Total output drops	Total number of messages discarded in this session.
Output queue	
size	Current size of the output queue.
threshold	Congestive-discard threshold. Number of messages in the queue after which new messages for high-bandwidth conversations are dropped.
drops	Number of dropped messages.
Conversations: active	Number of currently active conversations.
Conversations: max active	Maximum number of concurrent conversations allowed.

accounting Option Example

To display the number of packets of each protocol type that have been sent through all configured interfaces, use the **show interfaces accounting** keyword EXEC command. When you use the **accounting** option, only the accounting statistics are displayed.

**Note**

Except for protocols that are encapsulated inside other protocols, such as IP over X.25, the **accounting** option also shows the total of all bytes sent and received, including the MAC header. For example, it totals the size of the Ethernet packet or the size of a packet that includes High-Level Data Link Control (HDLC) encapsulation.

Per-packet accounting information is kept for the following protocols:

- Apollo
- AppleTalk
- Address Resolution Protocol (ARP) (for IP, Apollo, Frame Relay, Switched Multimegabit Data Service [SMDS])
- Connectionless Network Service (CLNS)
- DEC MOP—The routers use MOP packets to advertise their existence to Digital Equipment Corporation machines that use the MOP protocol. A router periodically broadcasts MOP packets to identify itself as a MOP host. This results in MOP packets being counted, even when DECnet is not being actively used.
- DECnet
- HP Probe
- IP
- LAN Manager (LAN Network Manager and IBM Network Manager)

- Novell
- Serial Tunnel (SDLC)
- Spanning tree
- SR Bridge
- Transparent Bridge
- VINES
- XNS

DWRED Example

The following is sample output from the **show interfaces** command when Distributed Weighted Random Early Detection (DWRED) is enabled on an interface. Notice that the packet drop strategy is listed as “VIP-based weighted RED.”

```
Router# show interfaces hssi0/0/0

Hssi0/0/0 is up, line protocol is up
  Hardware is cyBus HSSI
  Description: 45Mbps to R1
  Internet address is 200.200.14.250/30
  MTU 4470 bytes, BW 45045 Kbit, DLY 200 usec, rely 255/255, load 1/255
  Encapsulation HDLC, loopback not set, keepalive set (10 sec)
  Last input 00:00:02, output 00:00:03, output hang never
  Last clearing of "show interface" counters never
  Queueing strategy: fifo
  Packet Drop strategy: VIP-based weighted RED
  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
  1976 packets input, 131263 bytes, 0 no buffer
  Received 1577 broadcasts, 0 runts, 0 giants
  0 parity
  4 input errors, 4 CRC, 0 frame, 0 overrun, 0 ignored, 0 abort
  1939 packets output, 130910 bytes, 0 underruns
  0 output errors, 0 applique, 3 interface resets
  0 output buffers copied, 0 interrupts, 0 failures
```

ALC Example

The following is sample output from the **show interfaces** command for the serial 2 interface when ALC is enabled:

```
Router# show interfaces serial 2

Serial2 is up, line protocol is up
Hardware is CD2430
MTU 1500 bytes, BW 115 Kbit, DLY 20000 usec, rely 255/255, load 1/255
Encapsulation ALC, loopback not set
Full-duplex enabled.
    ascus in UP state: 42, 46
    ascus in DOWN state:
    ascus DISABLED:
Last input never, output never, output hang never
Last clearing of "show interface" counters never
Queueing strategy: fifo
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
0 packets input, 0 bytes, 0 no buffer
Received 0 broadcasts, 0 runts, 0 giants
0 input errors, 0 CRC, 0 frame, 0 overrun, 0 ignored, 0 abort
0 packets output, 0 bytes, 0 underruns
0 output errors, 0 collisions, 3 interface resets
0 output buffer failures, 0 output buffers swapped out
DCD=down DSR=down DTR=down RTS=down CTS=down
```

SDLC Example

The following is sample output from the **show interfaces** command for an SDLC primary interface supporting the SDLC function:

```
Router# show interfaces

Serial 0 is up, line protocol is up
Hardware is MCI Serial
MTU 1500 bytes, BW 1544 Kbit, DLY 20000 usec, rely 255/255, load 1/255
Encapsulation SDLC-PRIMARY, loopback not set
    Timers (msec): poll pause 100 fair poll 500. Poll limit 1
    [T1 3000, N1 12016, N2 20, K 7] timer: 56608 Last polled device: none
    SDLLC [ma: 0000.0C01.14--, ring: 7 bridge: 1, target ring: 10
    largest token ring frame 2052]
SDLC addr C1 state is CONNECT
    VS 6, VR 3, RCNT 0, Remote VR 6, Current retransmit count 0
    Hold queue: 0/12 IFRAMES 77/22 RNRs 0/0 SNRMs 1/0 DISCs 0/0
    Poll: clear, Poll count: 0, chain: p: C1 n: C1
    SDLLC [largest SDLC frame: 265, XID: disabled]
Last input 00:00:02, output 00:00:01, output hang never
Output queue 0/40, 0 drops; input queue 0/75, 0 drops
Five minute input rate 517 bits/sec, 30 packets/sec
Five minute output rate 672 bits/sec, 20 packets/sec
    357 packets input, 28382 bytes, 0 no buffer
    Received 0 broadcasts, 0 runts, 0 giants
    0 input errors, 0 CRC, 0 frame, 0 overrun, 0 ignored, 0 abort
    926 packets output, 77274 bytes, 0 underruns
    0 output errors, 0 collisions, 0 interface resets, 0 restarts
    2 carrier transitions
```

Table 4 shows the fields relevant to all SDLC connections.

Table 4 *show interfaces Field Descriptions (When SDLC Is Enabled)*

Field	Description
Timers (msec)	List of timers in milliseconds.
poll pause, fair poll, Poll limit	Current values of these timers, as described in the individual commands in this chapter.
T1, N1, N2, K	Current values for these variables, as described in the individual commands in this chapter.

Table 5 shows other data given for each SDLC secondary configured to be attached to this interface.

Table 5 *SDLC Field Descriptions*

Field	Description
addr	Address of this secondary.
state	Current state of this connection. The possible values are: <ul style="list-style-type: none"> • DISCONNECT—No communication is being attempted to this secondary. • CONNECT—A normal connect state exists between this router and this secondary. • DISCSENT—This router has sent a disconnect request to this secondary and is awaiting its response. • SNRMSENT—This router has sent a connect request (SNRM) to this secondary and is awaiting its response. • THEMBUSY—This secondary has told this router that it is temporarily unable to receive any more information frames. • USBUSY—This router has told this secondary that it is temporarily unable to receive any more information frames. • BOTHBUSY—Both sides have told each other that they are temporarily unable to receive any more information frames. • ERROR—This router has detected an error, and is waiting for a response from the secondary acknowledging this.
VS	Sequence number of the next information frame this station sends.
VR	Sequence number of the next information frame from this secondary that this station expects to receive.
RCNT	Number of correctly sequenced I-frames received when the Cisco IOS software was in a state in which it is acceptable to receive I-frames.
Remote VR	Last frame sent by this station was acknowledged by the other station.
Current retransmit count	Number of times the current I-frame or sequence of I-frames was re-sent.
Hold queue	Number of frames in the hold queue/maximum size of hold queue.
IFRAMEs, RNRs, SNRMs, DISCs	Sent/received count for these frames.

Table 5 SDLC Field Descriptions (continued)

Field	Description
Poll	“Set” if this router has a poll outstanding to the secondary; “clear” if it does not.
Poll count	Number of polls, in a row, given to this secondary at this time.
chain	Indicates the previous (p) and next (n) secondary address on this interface in the round-robin loop of polled devices.

show interfaces accounting Example

The following is sample output from the **show interfaces accounting** command:

```
Router# show interfaces accounting

Interface TokenRing0 is disabled

Ethernet0
      Protocol  Pkts In   Chars In   Pkts Out   Chars Out
          IP      873171   735923409   34624     9644258
          Novell  163849   12361626   57143     4272468
          DEC MOP    0         0           1          77
          ARP      69618    4177080    1529     91740

Interface Serial0 is disabled

Ethernet1
      Protocol  Pkts In   Chars In   Pkts Out   Chars Out
          IP         0         0           37     11845
          Novell     0         0          4591    275460
          DEC MOP    0         0           1          77
          ARP         0         0           7         420

Interface Serial1 is disabled
Interface Ethernet2 is disabled
Interface Serial2 is disabled
Interface Ethernet3 is disabled
Interface Serial3 is disabled
Interface Ethernet4 is disabled
Interface Ethernet5 is disabled
Interface Ethernet6 is disabled
Interface Ethernet7 is disabled
Interface Ethernet8 is disabled
Interface Ethernet9 is disabled

Fddio
      Protocol  Pkts In   Chars In   Pkts Out   Chars Out
          Novell    0         0          183     11163
          ARP        1         49           0          0
```

When the output indicates an interface is “disabled,” the router has received an excessive number of errors (over 5000 in a keepalive period).

Flow-Based WRED Example

The following is sample output from the **show interfaces** command issued for the serial1 interface for which flow-based WRED is enabled. The output shows 8 active flow-based WRED flows, that the maximum number of flows active at any time is 9, and that the maximum number of possible flows configured for the interface is 16.

```
Router# show interfaces serial1

Serial1 is up, line protocol is up
  Hardware is HD64570
  Internet address is 190.1.2.1/24
  MTU 1500 bytes, BW 1544 Kbit, DLY 20000 usec,
  Reliability 255/255, txload 237/255, rxload 1/255
  Encapsulation HDLC, loopback not set
  Keepalive not set
  Last input 00:00:22, output 00:00:00, output hang never
  Last clearing of "show interface" counters 00:17:58
  Input queue: 0/75/0 (size/max/drops); Total output drops: 2479

Queueing strategy: random early detection(RED)
  flows (active/max active/max): 8/9/16
  mean queue depth: 27
  drops: class  random    tail      min-th   max-th   mark-prob
          0      946      0         20      40      1/10
          1      488      0         22      40      1/10
          2      429      0         24      40      1/10
          3      341      0         26      40      1/10
          4      235      0         28      40      1/10
          5       40      0         31      40      1/10
          6       0       0         33      40      1/10
          7       0       0         35      40      1/10
          rsvp    0       0         37      40      1/10
30 second input rate 1000 bits/sec, 2 packets/sec
30 second output rate 119000 bits/sec, 126 packets/sec
  1346 packets input, 83808 bytes, 0 no buffer
  Received 12 broadcasts, 0 runts, 0 giants, 0 throttles
  0 input errors, 0 CRC, 0 frame, 0 overrun, 0 ignored, 0 abort
  84543 packets output, 9977642 bytes, 0 underruns
  0 output errors, 0 collisions, 6 interface resets
  0 output buffer failures, 0 output buffers swapped out
  0 carrier transitions
  DCD=up DSR=up DTR=up RTS=up CTS=up
```

DWFQ Example

The following is sample output from the **show interfaces** command when distributed weighted fair queueing (DWFQ) is enabled on an interface. Notice that the queueing strategy is listed as “VIP-based fair queueing.”

```
Router# show interfaces fastethernet 1/1/0

Fast Ethernet 1/1/0 is up, line protocol is up
  Hardware is cyBus Fast Ethernet Interface, address is 0007.f618.4448 (bia 00e0)
  Description: pkt input i/f for WRL tests (to pagent)
  Internet address is 80.0.2.70/24
  MTU 1500 bytes, BW 100000 Kbit, DLY 100 usec, rely 255/255, load 1/255
  Encapsulation ARPA, loopback not set, keepalive not set, fdx, 100BaseTX/FX
  ARP type: ARPA, ARP Timeout 04:00:00
  Last input never, output 01:11:01, output hang never
  Last clearing of "show interface" counters 01:12:31
  Queueing strategy: VIP-based fair queueing
  Output queue 0/40, 0 drops; input queue 0/75, 0 drops
  30 second input rate 0 bits/sec, 0 packets/sec
  30 second output rate 0 bits/sec, 0 packets/sec
    0 packets input, 0 bytes, 0 no buffer
  Received 0 broadcasts, 0 runts, 0 giants
    0 input errors, 0 CRC, 0 frame, 0 overrun, 0 ignored, 0 abort
    0 watchdog, 0 multicast
    0 input packets with dribble condition detected
    1 packets output, 60 bytes, 0 underruns
    0 output errors, 0 collisions, 0 interface resets
    0 babbles, 0 late collision, 0 deferred
    0 lost carrier, 0 no carrier
    0 output buffers copied, 0 interrupts, 0 failures
```

DNIS Binding Example

The following is sample output when the **show interfaces** command is used on an unbound dialer interface:

```
Router# show interfaces dialer0

Dialer0 is up (spoofing), line protocol is up (spoofing)
  Hardware is Unknown
  Internet address is 21.1.1.2/8
  MTU 1500 bytes, BW 64 Kbit, DLY 20000 usec, rely 255/255, load 3/255
  Encapsulation PPP, loopback not set
  DTR is pulsed for 1 seconds on reset
  Last input 00:00:34, output never, output hang never
  Last clearing of "show interface" counters 00:05:09
  Queueing strategy: fifo
  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 1000 bits/sec, 0 packets/sec
    18 packets input, 2579 bytes
    14 packets output, 5328 bytes
```

When the **show interfaces** command is issued on a bound dialer interface, you will get an additional report indicating the binding relationship. The output looks as follows:

```
Router# show interfaces dialer0

Dialer0 is up, line protocol is up
  Hardware is Unknown
  Internet address is 21.1.1.2/8
  MTU 1500 bytes, BW 64 Kbit, DLY 20000 usec, rely 255/255, load 1/255
  Encapsulation PPP, loopback not set
  DTR is pulsed for 1 seconds on reset
  Interface is bound to BRI0:1
  Last input 00:00:38, output never, output hang never
  Last clearing of "show interface" counters 00:05:36

  Queueing strategy: fifo
  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
    38 packets input, 4659 bytes
    34 packets output, 9952 bytes
Bound to:
BRI0:1 is up, line protocol is up
  Hardware is BRI
  MTU 1500 bytes, BW 64 Kbit, DLY 20000 usec, rely 255/255, load 1/255
  Encapsulation PPP, loopback not set, keepalive not set
  Interface is bound to Dialer0 (Encapsulation PPP)
  LCP Open, multilink Open
  Last input 00:00:39, output 00:00:11, output hang never
  Last clearing of "show interface" counters never
  Queueing strategy: fifo
  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
    78 packets input, 9317 bytes, 0 no buffer
    Received 65 broadcasts, 0 runts, 0 giants, 0 throttles
    0 input errors, 0 CRC, 0 frame, 0 overrun, 0 ignored, 0 abort
    93 packets output, 9864 bytes, 0 underruns
    0 output errors, 0 collisions, 7 interface resets
    0 output buffer failures, 0 output buffers swapped out
    4 carrier transitions
```

At the end of dialer0 output, the **show interfaces** command is executed on each physical interface bound to it.

In the following sample output, the physical interface is the B1 channel of the bri0 link. This example also illustrates that the output under the B channel keeps all hardware counts that are not displayed under any logical or virtual access interface. The line in the report that states "Interface is bound to

Dialer0 (Encapsulation LAPB)” indicates that this B interface is bound to Dialer0 and the encapsulation running over this connection is Link Access Procedure, Balanced (LAPB), not PPP, which is the encapsulation configured on the D interface and inherited by the B channel.

```
Router# show interfaces bri0:1
```

```
BRI0:1 is up, line protocol is up
Hardware is BRI
MTU 1500 bytes, BW 64 Kbit, DLY 20000 usec, rely 255/255, load 1/255
Encapsulation PPP, loopback not set, keepalive not set
Interface is bound to Dialer0 (Encapsulation LAPB)
LCP Open, multilink Open
Last input 00:00:31, output 00:00:03, output hang never
Last clearing of "show interface" counters never
Queueing strategy: fifo
Output queue 0/40, 0 drops; input queue 0/75, 0 drops
5 minute input rate 0 bits/sec, 1 packets/sec
5 minute output rate 0 bits/sec, 1 packets/sec
  110 packets input, 13994 bytes, 0 no buffer
  Received 91 broadcasts, 0 runts, 0 giants, 0 throttles
  0 input errors, 0 CRC, 0 frame, 0 overrun, 0 ignored, 0 abort
  135 packets output, 14175 bytes, 0 underruns
  0 output errors, 0 collisions, 12 interface resets
  0 output buffer failures, 0 output buffers swapped out
  8 carrier transitions
```

Any protocol configuration and states should be displayed from the Dialer0 interface.

show queueing

To list all or selected configured queueing strategies, use the **show queueing** privileged EXEC command.

```
show queueing [custom | fair | priority | random-detect [interface atm_subinterface
                [vc [[vpi/] vci]]]]
```

Syntax Description		
custom	(Optional)	Status of the custom queueing list configuration.
fair	(Optional)	Status of the fair queueing configuration.
priority	(Optional)	Status of the priority queueing list configuration.
random-detect	(Optional)	Status of the weighted random early detection (WRED) and distributed weighted random early detection (DWRED) configuration, including configuration of flow-based WRED.
interface <i>atm_subinterface</i>	(Optional)	Displays the WRED parameters of every virtual circuit (VC) with WRED enabled on the specified ATM subinterface.
vc	(Optional)	Displays the WRED parameters associated with a specific VC. If desired, both the virtual path identifier (VPI) and virtual circuit identifier (VCI) values, or just the VCI value, can be specified.
<i>vpi/</i>	(Optional)	Specifies the VPI. If the <i>vpi</i> argument is omitted, 0 is used as the VPI value for locating the permanent virtual circuit (PVC). If the <i>vpi</i> argument is specified, the / separator is required.
<i>vci</i>	(Optional)	Specifies the VCI.

Defaults If no keyword is entered, this command shows the configuration of all interfaces.

Command Modes Privileged EXEC

Command History	Release	Modification
	10.3	This command was introduced.
	12.0(4)T	The red keyword was changed to random-detect .
	12.1(2)T	This command was modified to include information about the Frame Relay PVC Interface Priority Queueing (FR PIPQ) feature.

Examples**FR PIPQ Example**

The following sample output shows that FR PIPQ (referred to as “DLCI priority queue”) is configured on serial interface 0. The output also shows the size of the four DLCI priority queues.

```
Router# show queueing
```

```
Current fair queue configuration:
```

Interface	Discard threshold	Dynamic queue count	Reserved queue count
Serial3/1	64	256	0
Serial3/3	64	256	0

```
Current DLCI priority queue configuration:
```

Interface	High limit	Medium limit	Normal limit	Low limit
Serial0	20	40	60	80

```
Current priority queue configuration:
```

```
List  Queue  Args
1      low   protocol ipx
1      normal protocol vines
1      normal protocol appletalk
1      normal protocol ip
1      normal protocol decnet
1      normal protocol decnet_node
1      normal protocol decnet_rout
1      normal protocol decnet_rout
1      medium protocol xns
1      high  protocol clns
1      normal protocol bridge
1      normal protocol arp
```

```
Current custom queue configuration:
```

```
Current random-detect configuration:
```

Weighted Fair Queueing Example

The following is sample output from the **show queueing** command. Two active conversations are in serial interface 0. Weighted fair queueing (WFQ) ensures that both of these IP data streams—both using TCP—receive equal bandwidth on the interface while they have messages in the pipeline, even though more FTP data is in the queue than remote copy protocol (RCP) data.

```
Router# show queueing
```

```
Current fair queue configuration:
```

Interface	Discard threshold	Dynamic queue count	Reserved queue count
Serial0	64	256	0
Serial1	64	256	0
Serial2	64	256	0
Serial3	64	256	0

```
Current priority queue configuration:
```

List	Queue	Args
1	high	protocol cdp
2	medium	interface Ethernet1

```
Current custom queue configuration:
```

```
Current random-detect configuration:
```

```
Serial5
  Queueing strategy:random early detection (WRED)
  Exp-weight-constant:9 (1/512)
  Mean queue depth:40
```

Class	Random drop	Tail drop	Minimum threshold	Maximum threshold	Mark probability
0	1401	9066	20	40	1/10
1	0	0	22	40	1/10
2	0	0	24	40	1/10
3	0	0	26	40	1/10
4	0	0	28	40	1/10
5	0	0	31	40	1/10
6	0	0	33	40	1/10
7	0	0	35	40	1/10
rsvp	0	0	37	40	1/10

Custom Queueing Example

The following is sample output from the **show queueing custom** command:

```
Router# show queueing custom
```

```
Current custom queue configuration:
```

List	Queue	Args
3	10	default
3	3	interface Tunnel3
3	3	protocol ip
3	3	byte-count 444 limit 3

Flow-Based WRED Example

The following is sample output from the **show queueing random-detect** command. The output shows that the interface is configured for flow-based WRED to ensure fair packet drop among flows. The **random-detect flow average-depth-factor** command was used to configure a scaling factor of 8 for this interface. The scaling factor is used to scale the number of buffers available per flow and to

determine the number of packets allowed in the output queue of each active flow before the queue is susceptible to packet drop. The maximum flow count for this interface was set to 16 by the **random-detect flow count** command.

Router# **show queueing random-detect**

```

Current random-detect configuration:
Serial1
  Queueing strategy:random early detection (WRED)
  Exp-weight-constant:9 (1/512)
  Mean queue depth:29
  Max flow count:16      Average depth factor:8
  Flows (active/max active/max):39/40/16

Class  Random      Tail   Minimum  Maximum  Mark
      drop      drop  threshold threshold probability
0         31         0       20       40      1/10
1         33         0       22       40      1/10
2         18         0       24       40      1/10
3         14         0       26       40      1/10
4         10         0       28       40      1/10
5          0         0       31       40      1/10
6          0         0       33       40      1/10
7          0         0       35       40      1/10
rsvp      0         0       37       40      1/10
    
```

DWRED Example

The following is sample output from the **show queueing random-detect** command for DWRED:

Current random-detect configuration:

```

FastEthernet2/0/0
  Queueing strategy:fifo
  Packet drop strategy:VIP-based random early detection (DWRED)
  Exp-weight-constant:9 (1/512)
  Mean queue depth:0
  Queue size:0      Maximum available buffers:6308
  Output packets:5 WRED drops:0 No buffer:0
    
```

```

Class  Random      Tail   Minimum  Maximum  Mark      Output
      drop      drop  threshold threshold probability Packets
0         0         0     109     218     1/10      5
1         0         0     122     218     1/10      0
2         0         0     135     218     1/10      0
3         0         0     148     218     1/10      0
4         0         0     161     218     1/10      0
5         0         0     174     218     1/10      0
6         0         0     187     218     1/10      0
7         0         0     200     218     1/10      0
    
```

Table 6 describes the fields shown in the preceding displays.

Table 6 *show queueing Field Descriptions*

Field	Description
Discard threshold	Number of messages allowed in each queue.
Dynamic queue count	Number of dynamic queues used for best-effort conversations.
Reserved queue count	Number of reservable queues used for reserved conversations.
High limit	High DLCI priority queue size in maximum number of packets.

Table 6 show queueing Field Descriptions (continued)

Field	Description
Medium limit	Medium DLCI priority queue size in maximum number of packets.
Normal limit	Normal DLCI priority queue size in maximum number of packets.
Low limit	Low DLCI priority queue size in maximum number of packets.
List	Custom queueing—Number of the queue list. Priority queueing—Number of the priority list.
Queue	Custom queueing—Number of the queue. Priority queueing—Priority queue level (high , medium , normal , or low).
Args	Packet matching criteria for that queue.
Exp-weight-constant	Exponential weight factor.
Mean queue depth	Average queue depth. It is calculated 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.
Class	IP Precedence value.
Random drop	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 drop	Number of packets dropped when the mean queue depth is greater than the maximum threshold value for the specified IP Precedence value.
Minimum threshold	Minimum WRED threshold in number of packets.
Maximum threshold	Maximum WRED threshold in number of packets.
Mark probability	Fraction of packets dropped when the average queue depth is at the maximum threshold.

Related Commands

Command	Description
custom-queue-list	Assigns a custom queue list to an interface.
fair-queue (WFQ)	Enables WFQ for an interface.
frame-relay interface-queue priority	Enables FR PIPQ on a Frame Relay interface, and assigns priority to a PVC within a Frame Relay map class.
exponential-weighting-constant	Configures the exponential weight factor for the average queue size calculation for a WRED parameter group.
precedence	Configures a WRED group for a particular IP Precedence.
priority-group	Assigns the specified priority list to an interface.
priority-list interface	Establishes queueing priorities on packets entering from a given interface.
priority-list queue-limit	Specifies the maximum number of packets that can be waiting in each of the priority queues.
queue-list interface	Establishes queueing priorities on packets entering on an interface.

Command	Description
queue-list queue byte-count	Specifies how many bytes the system allows to be delivered from a given queue during a particular cycle.
random-detect (interface)	Enables WRED or DWRED.
random-detect flow average-depth-factor	Sets the multiplier to be used in determining the average depth factor for a flow when flow-based WRED is enabled.
random-detect flow count	Sets the flow count for flow-based WRED.
show interfaces	Displays the statistical information specific to a serial interface.
show queue	Displays the contents of packets inside a queue for a particular interface or VC.
show queueing interface	Displays the queueing statistics of an interface or VC.

Debug Commands

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

debug priority

To debug priority output queueing, use the **debug priority** privileged EXEC command. To disable debugging output, use the **no** form of this command.

debug priority

no debug priority

Syntax Description This command has no arguments or keywords.

Defaults No default behavior or values.

Command Modes Privileged EXEC

Command History	Release	Modification
	12.1(2)T	This command was modified to include information about PVC priority.

Examples The following example shows how to debug priority output queueing:

```
Router# debug priority

Priority output queueing debugging is on
Router#
```

The following is sample output from the **debug priority** command when the Frame Relay PVC Interface Priority Queueing (FR PIPQ) feature is configured on serial interface 0:

```
Router# debug priority

00:49:05:PQ:Serial0 dlci 100 -> high
00:49:05:PQ:Serial0 output (Pk size/Q 24/0)
00:49:05:PQ:Serial0 dlci 100 -> high
00:49:05:PQ:Serial0 output (Pk size/Q 24/0)
00:49:05:PQ:Serial0 dlci 100 -> high
00:49:05:PQ:Serial0 output (Pk size/Q 24/0)
00:49:05:PQ:Serial0 dlci 200 -> medium
00:49:05:PQ:Serial0 output (Pk size/Q 24/1)
00:49:05:PQ:Serial0 dlci 300 -> normal
00:49:05:PQ:Serial0 output (Pk size/Q 24/2)
00:49:05:PQ:Serial0 dlci 400 -> low
00:49:05:PQ:Serial0 output (Pk size/Q 24/3)
```

Glossary

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 nonreal-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.

PIPQ—Permanent virtual circuit (PVC) interface priority queueing. An interface-level priority queueing scheme in which prioritization is based on destination PVC rather than packet contents.

quality of service—Measure of performance for a transmission system that reflects its transmission quality and service availability.

VoFR—Voice over Frame Relay. Enables a router to carry voice traffic over a Frame Relay network. When voice traffic is sent 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.