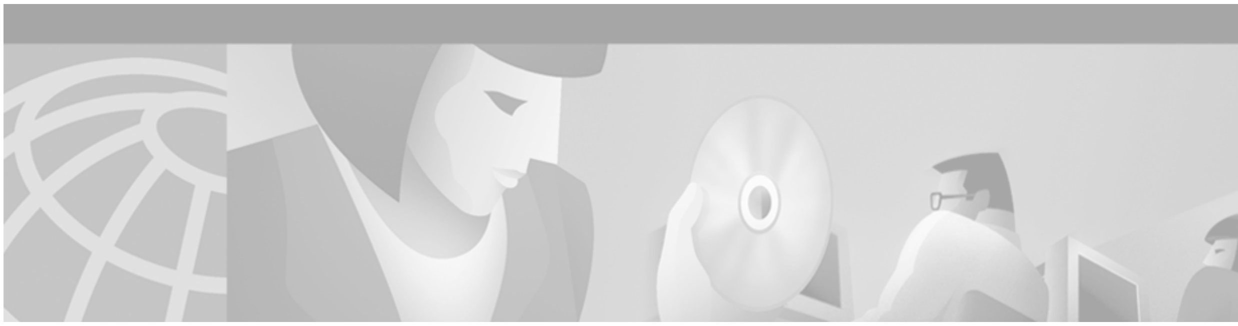

Frame Relay Fragmentation with Hardware Compression

This feature module describes the Frame Relay Fragmentation with Hardware Compression feature and includes the following sections:

Feature Overview

The Frame Relay Fragmentation with Hardware Compression feature introduces the following functionality:

- Frame Relay Fragmentation and Hardware Compression Interoperability
- Hardware Compression and Header Compression Interoperability
- Hardware Compression and Software Compression Interoperability

Frame Relay Fragmentation and Hardware Compression Interoperability

Before the Frame Relay Fragmentation with Hardware Compression feature was introduced, Frame Relay fragmentation worked with software compression, but not with hardware compression.

The introduction of this new feature enables FRF.12, FRF.11 Annex C, and Cisco proprietary fragmentation to work with hardware compression on interfaces and virtual circuits (VCs) using Cisco proprietary or Internet Engineering Task Force (IETF) encapsulation types.

Note When payload compression and Frame Relay fragmentation are used at the same time, payload compression is always performed before fragmentation.

Hardware Compression and Header Compression Interoperability

Before this feature was introduced, hardware compression and header compression could not be used on the same VC or interface because header compression worked only with Cisco proprietary encapsulation, and hardware compression worked only with IETF-compliant encapsulation.

Now a new, proprietary hardware and software compression protocol called *data-stream compression* can be used on the same VC or interface as header compression. Data-stream compression is functionally equivalent to FRF.9 compression and must be used with Cisco proprietary encapsulation. Frame Relay fragmentation can also be enabled.

Note On IETF-encapsulated VCs and interfaces, FRF.9 hardware and software compression will work with Frame Relay fragmentation, but *not* with header compression.

Hardware Compression and Software Compression Interoperability

The Frame Relay Fragmentation with Hardware Compression feature provides hardware and software compression interoperability when hardware compression is configured on one side of the link and software compression is configured on the other side.

Benefits

The Frame Relay Fragmentation with Hardware Compression feature makes hardware compression available to networks that transmit voice and to other networks that use fragmentation. Hardware compression, though functionally equivalent to software compression, provides the benefit of offloading computationally intensive compression algorithms from the CPUs of routers, freeing up bandwidth for other functionality and features.

Restrictions

- Voice over Frame Relay and Voice over IP packets will not be payload-compressed when Frame Relay fragmentation is configured.
- On VCs using IETF encapsulation, FRF.9 hardware and software compression will work with Frame Relay fragmentation but will not work with header compression.

Related Documents

- *Cisco IOS Wide-Area Networking Configuration Guide*, Release 12.1
- *Cisco IOS Wide-Area Networking Command Reference*, Release 12.1
- *FRF.12 Support on Additional Platforms*, Cisco IOS Release 12.1(2)T
- *Voice over Frame Relay Using FRF.11 and FRF.12*, Cisco IOS Release 12.0(4)T

Supported Platforms

- Cisco 2600
- Cisco 3600 series
- Cisco 7200 series

Supported Standards, MIBs, and RFCs

Standards

- FRF.9, *Data Compression Over Frame Relay Implementation Agreement*
- FRF.12, *Frame Relay Fragmentation Implementation Agreement*

MIBs

No new or modified MIBs are supported by this feature.

To obtain lists of MIBs supported by platform and Cisco IOS release and to download MIB modules, go to the Cisco MIB web site on Cisco Connection Online (CCO) at http://www.cisco.com/public/sw-center/netmgmt/cmtk/mibs.shtml.

RFCs

No new or modified RFCs are supported by this feature.

Prerequisites

Frame Relay fragmentation must be used with the following hardware compression modules:

- Cisco 2600 AIM-COMPR2
- Cisco 3620 and 3640 NM-COMPR
- Cisco 3660 AIM-COMPR4
- Cisco 7200 SA-COMPR

Configuration Tasks

See the following sections for configuration tasks for the Frame Relay Fragmentation with Hardware Compression feature. Each task in the list is identified as optional or required.

- Configuring Frame Relay Fragmentation with Hardware Compression (Required)
- Configuring Hardware Compression and Header Compression on a Point-to-Point Subinterface (Optional)
- Configuring Hardware Compression and Header Compression on a Multipoint Subinterface (Optional)
- Verifying Frame Relay Fragmentation with Hardware Compression (Optional)

Configuring Frame Relay Fragmentation with Hardware Compression

No new configuration tasks are required to configure Frame Relay fragmentation with hardware compression. See the section "Configuration Examples" for an example of Frame Relay fragmentation with Hardware Compression configured on the same interface.

Configuring Hardware Compression and Header Compression on a Point-to-Point Subinterface

To configure data-stream hardware compression and TCP or Real-Time Transport Protocol (RTP) header compression on a point-to-point subinterface, use the following commands beginning in global configuration mode. Note that when you specify data-stream hardware compression, Cisco proprietary encapsulation is automatically enabled.

	Command	Purpose
Step 1	`Router(config)#` **`interface`** *`type number`* **`point-to-point`**	Configures a subinterface type and enters subinterface configuration mode.
Step 2	`Router(config-subif)#` **`ip address`** *`address mask`*	Sets the IP address for an interface.
Step 3	`Router(config-subif)#` **`frame-relay interface-dlci`** *`dlci`*	Assigns a DLCI[1] to a specified Frame Relay subinterface on the router or access server.
Step 4	`Router(config-subif)#` **`frame-relay payload-compress data-stream stac`** `[`*`hardware-options`*`]`	Enables hardware compression on an interface or subinterface that uses Cisco proprietary encapsulation.
Step 5	`Router(config-subif)#` **`frame-relay ip tcp header-compression`** `[`**`passive`**`]`	Configures an interface to ensure that the associated PVCs[2] carry outgoing TCP headers in compressed form.
	or	
	`Router(config-subif)#` **`frame-relay ip rtp header-compression`** `[`**`passive`**`]`	Enables RTP header compression on the physical interface.

1. DLCI = data-link connection identifier
2. PVC = permanent virtual circuit

Configuring Hardware Compression and Header Compression on a Multipoint Subinterface

To configure data-stream hardware compression and TCP or RTP header compression on a multipoint subinterface, use the following commands beginning in global configuration mode. Note that when you specify data-stream hardware compression, Cisco proprietary encapsulation is automatically enabled.

	Command	Purpose
Step 1	`Router(config)# interface type number multipoint`	Configures a subinterface type and enters subinterface configuration mode.
Step 2	`Router(config-subif)# frame-relay interface-dlci dlci`	Assigns a DLCI to a specified Frame Relay subinterface on the router or access server.
Step 3	`Router(config-subif)# frame-relay map protocol protocol-address dlci [payload-compress data-stream stac [hardware-options]]`	Defines the mapping between a destination protocol address and the DLCI used to connect to the destination address on an interface that uses Cisco proprietary encapsulation.
Step 4	`Router(config-subif)# frame-relay ip tcp header-compression [passive]`	Configures an interface to ensure that the associated PVCs carry outgoing TCP headers in compressed form.
	or	
	`Router(config-subif)# frame-relay ip rtp header-compression [passive]`	Enables RTP header compression on the physical interface.

Verifying Frame Relay Fragmentation with Hardware Compression

To verify that Frame Relay fragmentation is working with hardware compression, use one or more of the following privileged EXEC commands:

Command	Purpose
`Router# show compress`	Displays compression statistics.
`Router# show frame-relay pvc dlci`	Displays statistics about PVCs for Frame Relay interfaces.
`Router# show traffic-shape queue`	Displays information about the elements queued at a particular time at the VC DLCI level.

Monitoring and Maintaining Frame Relay Fragmentation with Hardware Compression

To monitor Frame Relay fragmentation with hardware compression, use the same commands listed in the section "Verifying Frame Relay Fragmentation with Hardware Compression."

Configuration Examples

This section provides the following configuration examples:

- Frame Relay Fragmentation with Hardware Compression Configuration Example
- Hardware Compression with Header Compression on a Point-to-Point Subinterface Configuration Example

- Hardware Compression with Header Compression on a Multipoint Subinterface Configuration Example
- Hardware Compression with Header Compression and Frame Relay Fragmentation Configuration Example

Frame Relay Fragmentation with Hardware Compression Configuration Example

In the following example, FRF.12 fragmentation and FRF.9 hardware compression are configured on multipoint interface 3/1 and point-to-point interface 3/1.1:

```
interface serial3/1
 ip address 10.1.0.1 255.255.255.0
 encapsulation frame-relay
 frame-relay traffic-shaping
 frame-relay class frag
 frame-relay map ip 10.1.0.2 110 broadcast ietf payload-compress frf9 stac
!
interface serial3/1.1 point-to-point
 ip address 10.2.0.1 255.255.255.0
 frame-relay interface-dlci 120 ietf
 frame-relay payload-compress frf9 stac
!
map-class frame-relay frag
 frame-relay cir 64000
 frame-relay bc 640
 frame-relay fragment 100
```

Hardware Compression with Header Compression on a Point-to-Point Subinterface Configuration Example

The following example shows the configuration of data-stream hardware compression and TCP header compression on point-to-point interface 1/0.1:

```
interface serial1/0
  encapsulation frame-relay
  frame-relay traffic-shaping
 !
 interface serial1/0.1 point-to-point
  ip address 10.0.0.1 255.0.0.0
  frame-relay interface-dlci 100
  frame-relay payload-compress data-stream stac
  frame-relay ip tcp header-compression
```

Hardware Compression with Header Compression on a Multipoint Subinterface Configuration Example

The following example shows the configuration of data-stream hardware compression and TCP header compression on multipoint interface 3/1:

```
interface serial3/1
 ip address 10.1.0.1 255.255.255.0
 encapsulation frame-relay
 frame-relay traffic-shaping
 frame-relay map ip 10.1.0.2 110 broadcase cisco payload-compress data-stream stac
 frame-relay ip tcp header-compression
```

Hardware Compression with Header Compression and Frame Relay Fragmentation Configuration Example

The following example shows the configuration of data-stream hardware compression, RTP header compression, and FRF.12 fragmentation on point-to-point interface 1/0.1:

```
interface serial1/0
  encapsulation frame-relay
  frame-relay traffic-shaping
 !
 interface serial1/0.1 point-to-point
  ip address 10.0.0.1 255.0.0.0
  frame-relay interface-dlci 100
  frame-relay class frag
  frame-relay payload-compress data-stream stac
  frame-relay ip rtp header-compression
 !
 map-class frame-relay frag
  frame-relay cir 64000
  frame-relay bc 640
  frame-relay be 0
  frame-relay fragment 100
  frame-relay ip rtp priority 16000 16000 20
```

Command Reference

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

- **frame-relay map**
- **frame-relay payload-compress**
- **show frame-relay pvc**
- **show traffic-shape queue**

frame-relay map

To define the mapping between a destination protocol address and the data-link connection identifier (DLCI) used to connect to the destination address, use the **frame-relay map** interface configuration command. To delete the map entry, use the **no** form of this command.

frame-relay map *protocol protocol-address dlci* [**broadcast**] [**ietf** | **cisco**] [**payload-compress** {**packet-by-packet** | **frf9 stac** [*hardware-options*] | **data-stream stac** [*hardware-options*]}]

no frame-relay map *protocol protocol-address*

Syntax Description

protocol	Supported protocol, bridging, or logical link control keywords: **appletalk**, **decnet**, **dlsw, ip**, **ipx**, **llc2**, **rsrb**, **vines,** and **xns**.
protocol-address	Destination protocol address.
dlci	DLCI number used to connect to the specified protocol address on the interface.
broadcast	(Optional) Forwards broadcasts to this address when multicast is not enabled (see the **frame-relay multicast-dlci** command for more information about multicasts). This keyword also simplifies the configuration of OSPF[1] (see the "Usage Guidelines" section for more detail).
ietf	(Optional) IETF[2] form of Frame Relay encapsulation. Used when the router or access server is connected to the equipment of another vendor across a Frame Relay network.
cisco	(Optional) Cisco encapsulation method.
payload-compress packet-by-packet	(Optional) Enables packet-by-packet payload compression using the Stacker method.
payload-compress frf9 stac	(Optional) Enables FRF.9 compression using the Stacker method: • If the router contains a CSA,[3] compression is performed in the CSA hardware (hardware compression). • If the CSA is not available, compression is performed in the software installed on the VIP2[4] (distributed compression). • If the VIP2 is not available, compression is performed in the main processor of the router (software compression).
payload-compress data-stream stac	(Optional) Enables data-stream compression using the Stacker method: • If the router contains a CSA, compression is performed in the CSA hardware (hardware compression). • If the CSA is not available, compression is performed in the main processor of the router (software compression).

hardware-options	• (Optional) **distributed**. Specifies that compression is implemented in the software that is installed in a VIP2. If the VIP2 is not available, compression is performed in the main processor of the router (software compression). This option applies only to the Cisco 7500 series routers. This option is not supported with data-stream compression. • (Optional) **software**. Specifies that compression is implemented in the Cisco IOS software installed in the main processor of the router. • (Optional) **csa** *csa_number*. Specifies the CSA to use for a particular interface. This option applies only to Cisco 7200 series routers.

1. OSPF = open shortest path first
2. IETF = Internet Engineering Task Force
3. CSA = compression service adapter
4. VIP2 = second generation Versatile Interface Processor

Defaults

No mapping is defined.

Command Modes

Interface configuration

Command History

Release	Modification
10.0	This command was introduced.
11.3	The **payload-compress frf9 stac** keyword was added.
12.1(5)T	The **payload-compress data-stream stac** keyword was added.

Usage Guidelines

Many DLCIs can be known by a router or access server and can send data to many different places, but they are all multiplexed over one physical link. The Frame Relay map defines the logical connection between a specific protocol and address pair and the correct DLCI.

The optional **ietf** and **cisco** keywords allow flexibility in the configuration. If no keywords are specified, the map inherits the attributes set with the **encapsulation frame-relay** command. You can also use the encapsulation options to specify that, for example, all interfaces use IETF encapsulation except one, which needs the original Cisco encapsulation method and can be configured through use of the **cisco** keyword with the **frame-relay map** command.

Data-stream compression is supported on interfaces and virtual circuits (VCs) using Cisco proprietary encapsulation. When the **data-stream stac** keyword is specified, Cisco encapsulation is automatically enabled. FRF.9 compression is supported on IETF-encapsulated VCs and interfaces. When the **frf9 stac** keyword is specified, IETF encapsulation is automatically enabled.

Packet-by-packet compression is Cisco-proprietary and will not interoperate with routers of other manufacturers.

You can disable payload compression by entering the **no frame-relay map payload** command and then entering the **frame-relay map** command again with one of the other encapsulation keywords (**ietf** or **cisco**).

Use the **frame-relay map** command to enable or disable payload compression on multipoint interfaces. Use the **frame-relay payload-compress** command to enable or disable payload compression on point-to-point interfaces.

We recommend that you shut down the interface before changing encapsulation types. Although this is not required, shutting down the interface ensures that the interface is reset for the new encapsulation.

The **broadcast** keyword provides two functions: it forwards broadcasts when multicasting is not enabled, and it simplifies the configuration of OSPF for nonbroadcast networks that will use Frame Relay.

The **broadcast** keyword might also be required for some routing protocols—for example, AppleTalk—that depend on regular routing table updates, especially when the router at the remote end is waiting for a routing update packet to arrive before adding the route.

By requiring selection of a designated router, OSPF treats a nonbroadcast, multiaccess network such as Frame Relay in much the same way as it treats a broadcast network. In previous releases, selection of a designated router required manual assignment in the OSPF configuration using the **neighbor interface** router command. When the **frame-relay map** command (with the **broadcast** keyword) and the **ip ospf network** command (with the **broadcast** keyword) are configured, there is no need to configure any neighbors manually. OSPF will now automatically run over the Frame Relay network as a broadcast network. (See the **ip ospf network** interface command for more detail.)

Note The OSPF broadcast mechanism assumes that IP class D addresses are never used for regular traffic over Frame Relay.

Examples

IP Address Mapping Example

The following example maps the destination IP address 172.16.123.1 to DLCI 100:

```
interface serial 0
 frame-relay map ip 172.16.123.1 100 broadcast
```

OSPF will use DLCI 100 to broadcast updates.

FRF.9 Compression Example

The following example shows FRF.9 compression configuration using the **frame-relay map** command:

```
interface serial2/0/1
 ip address 172.16.1.4 255.255.255.0
 no ip route-cache
 encapsulation frame-relay ietf
 no keepalive
 shutdown
 frame-relay map ip 172.16.1.1 105 ietf payload-compress frf9 stac
!
```

Data-Stream Compression Example

The following example shows data-stream compression configuration using the **frame-relay map** command:

```
interface serial0/0
 frame-relay map ip 10.0.0.1 100 payload-compress data-stream stac
```

Related Commands

Command	Description
encapsulation frame-relay	Enables Frame Relay encapsulation.
frame-relay payload-compress	Enables Stacker payload compression on a specified point-to-point interface or subinterface.
ip ospf network	Configures the OSPF network type to a type other than the default for a given medium.

frame-relay payload-compress

To enable Stacker payload compression on a specified point-to-point interface or subinterface, use the **frame-relay payload-compress** interface configuration command. To disable payload compression on a specified point-to-point interface or subinterface, use the **no** form of this command.

frame-relay payload-compress {**packet-by-packet** | **frf9 stac** [*hardware-options*] | **data-stream stac** [*hardware-options*]}

no frame-relay payload-compress {**packet-by-packet** | **frf9 stac** | **data-stream stac**}

Syntax Description

packet-by-packet	Packet-by-packet payload compression using the Stacker method.
frf9 stac	Enables FRF.9 compression using the Stacker method. • If the router contains a CSA,[1] compression is performed in the CSA hardware (hardware compression). • If the CSA is not available, compression is performed in the software installed on the VIP2[2] (distributed compression). • If the VIP2 is not available, compression is performed in the main processor of the router (software compression).
hardware-options	• (Optional) **distributed**. Specifies that compression is implemented in the software that is installed in a VIP2. If the VIP2 is not available, compression is performed in the main processor of the router (software compression). This option applies only to the Cisco 7500 series routers. This option is not supported with data-stream compression. • (Optional) **software**. Specifies that compression is implemented in the Cisco IOS software installed in the main processor of the router. • (Optional) **csa** *csa_number*. Specifies the CSA to use for a particular interface. This option applies only to Cisco 7200 series routers.
data-stream stac	Enables data-stream compression using the Stacker method. • If the router contains a CSA, compression is performed in the CSA hardware (hardware compression). • If the CSA is not available, compression is performed in the main processor of the router (software compression).

1. CSA = compression service adapter
2. VIP2 = second generation Versatile Interface Processor

Defaults

Disabled

Command Modes

Interface configuration

Command History

Release	Modification
11.0	This command was introduced.
11.2	The **packet-by-packet** keyword was added.
11.3	The **frf9 stac** keyword was added.
12.1(5)T	The **data-stream stac** keyword was added.

Usage Guidelines

Use the **frame-relay payload-compress** command to enable or disable payload compression on a point-to-point interface or subinterface. Use the **frame-relay map** command to enable or disable payload compression on a multipoint interface or subinterface.

We recommend that you shut down the interface before changing encapsulation types. Although shutting down the interface is not required, it ensures that the interface is reset for the new encapsulation.

Data-stream hardware compression is supported on interfaces and virtual circuits (VCs) using Cisco proprietary encapsulation. When the **data-stream stac** keyword is specified, Cisco encapsulation is automatically enabled. FRF.9 compression is supported on VCs and interfaces that using Internet Engineering Task Force (IETF) encapsulation type. When the **frf9 stac** keyword is specified, IETF encapsulation is automatically enabled.

Examples

FRF.9 Compression Example

The following example configures FRF.9 compression for subinterfaces:

```
interface serial2/0/0
 no ip address
 no ip route-cache
 encapsulation frame-relay
 ip route-cache distributed
 no keepalive
 shutdown
!
interface serial2/0/0.500 point-to-point
 ip address 172.16.1.4 255.255.255.0
 no cdp enable
 frame-relay interface-dlci 500 ietf
 frame-relay payload-compress frf9 stac
```

Data-Stream Compression Example

The following example shows the configuration of data-stream compression using the **frame-relay payload-compress** command:

```
interface serial1/0
 encapsulation frame-relay
 frame-relay traffic-shaping
!
interface serial1/0.1 point-to-point
 ip address 10.0.0.1 255.0.0.0
 frame-relay interface-dlci 100
 frame-relay payload-compress data-stream stac
```

Related Commands

Command	Description
frame-relay map	Defines mapping between a destination protocol address and the DLCI used to connect to the destination address.

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*] [*dlci*]

Syntax Description

interface	(Optional) Indicates a specific interface for which PVC information will be displayed.
interface	(Optional) Interface number containing the data-link connection identifiers (DLCIs) for which you wish to display PVC information.
dlci	(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 Frame Relay PVC Interface Priority Queueing.
12.1(5)T	This command was modified to display the number of packets in the post-hardware-compression queue.

Usage Guidelines

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

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

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 *dlci* argument.

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

Traffic Shaping

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

If a Local Management Interface (LMI) status report indicates that a PVC is not active, 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 examples 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 Fragmentation and Hardware Compression Example

The following is sample output from the **show frame-relay pvc** command for a PVC that is configured with Cisco-proprietary fragmentation and hardware compression:

```
Router# show frame-relay pvc 110

PVC Statistics for interface Serial0/0 (Frame Relay DTE)

DLCI = 110, DLCI USAGE = LOCAL, PVC STATUS = STATIC, INTERFACE = Serial0/0

  input pkts 409           output pkts 409          in bytes 3752
  out bytes 4560           dropped pkts 1           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 3d00h, last time pvc status changed 2d22h
  Service type VoFR-cisco
   Voice Queueing Stats: 0/100/0 (size/max/dropped)
  Post h/w compression queue: 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
  configured voice bandwidth 16000, used voice bandwidth 0
  fragment type VoFR-cisco         fragment size 100
  cir 64000     bc   640       be 0         limit 80     interval 10
  mincir 32000     byte increment 80    BECN response no
  frags 428       bytes 4810      frags delayed 24        bytes delayed 770
  shaping inactive
  traffic shaping drops 0
  ip rtp priority parameters 16000 32000 20000
```

Frame Relay PVC Priority Queueing Example

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

```
Router# 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 64-K 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.

```
Router# show frame-relay pvc 100

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

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

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

```
 Class priority-data
  Weighted Fair Queueing
      Output Queue: Conversation 74
        Bandwidth 40 (%) Packets Matched 0 Max Threshold 64 (packets)
        (pkts discards/bytes discards/tail drops) 0/0/0
 Class class-default
  Weighted Fair Queueing
      Flow Based Fair Queueing
      Maximum Number of Hashed Queues 64  Max Threshold 20 (packets)
  Output queue size 0/max total 600/drops 0
  fragment type end-to-end         fragment size 50
  cir 64000     bc   640       be 0         limit 80     interval 10
  mincir 64000     byte increment 80    BECN response no
  frags 0         bytes 0         frags delayed 0         bytes delayed 0
  shaping inactive
  traffic shaping drops 0
```

PPP over Frame Relay Example

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)
```

Voice over Frame Relay Example

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 from 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
```

FRF.12 Fragmentation Example

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

Multipoint Subinterfaces Transporting Data

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
```

PVC Transporting Voice and Data

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

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

Table 1 show frame-relay pvc Field Descriptions

Field	Description
DLCI	One of the 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.
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.
priority	Priority assigned to the PVC.
Service type	Type of service performed by this PVC. Can be VoFR or VoFR-cisco.
Post h/w compression queue	Number of packets in the post-hardware-compression queue when hardware compression and Frame Relay fragmentation are configured.
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[1] 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.

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

Field	Description
Max Threshold	Maximum queue size for this class when WRED is not used.
pkts discards	Number of packets discarded for this class.
bytes discards	Number of bytes discarded for this class.
tail drops	Number of packets discarded for this class because the queue was full.
mean queue depth	Average queue depth based on the actual queue depth on the interface and the exponential weighting constant. It is a moving average. The minimum and maximum thresholds are compared against this value to determine drop decisions.
drops:	WRED parameters.
class	IP Precedence value.
random	Number of packets randomly dropped when the mean queue depth is between the minimum threshold value and the maximum threshold value for the specified IP Precedence value.
tail	Number of packets dropped when the mean queue depth is greater than the maximum threshold value for the specified IP Precedence value.
min-th	Minimum WRED threshold in number of packets.
max-th	Maximum WRED threshold in number of packets.
mark-prob	Fraction of packets dropped when the average queue depth is at the maximum threshold.
Maximum Number of Hashed Queues	(Applies to class default only) Number of queues available for unclassified flows.
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.

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

Field	Description
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 with the **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.
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.

1. WFQ = weighted fair queueing

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 voice over Frame Relay DLCIs.
show interfaces serial	Displays information about a serial interface.

Command	Description
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 traffic-shape queue

To display information about the elements queued at a particular time at the virtual circuit data-link connection identifier (DLCI) level, use the **show traffic-shape queue** EXEC command.

show traffic-shape queue [*interface* [*dlci*]]

Syntax Description

interface	(Optional) Interface containing the DLCI(s) for which you wish to display information about queued elements.
dlci	(Optional) Specific DLCI for which you wish to display information about queued elements.

Command Modes

EXEC

Command History

Release	Modification
11.2	This command was introduced.
12.0(3)XG	The *dlci* argument was added.
12.0(4)T	The *dlci* argument was added.
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(5)T	This command was modified to display the number of packets in the post-hardware-compression queue.

Usage Guidelines

When no parameters are specified with this command, the output displays information for all interfaces and DLCIs containing queued elements. When an interface and DLCI are specified, information is displayed about the queued elements for that DLCI only.

Examples

The following is sample output from the **show traffic-shape queue** command when hardware compression and Frame Relay fragmentation are configured:

```
Router# show traffic-shape queue
Post h/w compression queue: 0
Traffic queued in shaping queue on Serial0/0 dlci 110
  Queueing strategy: weighted fair
  Queueing Stats: 0/600/64/0 (size/max total/threshold/drops)
     Conversations  0/1/16 (active/max active/max total)
     Reserved Conversations 0/2 (allocated/max allocated)
```

The following is sample output from the **show traffic-shape queue** command when weighted fair queueing is configured on the map class associated with DLCI 16:

```
Router# show traffic-shape queue serial1/1 dlci 16

Traffic queued in shaping queue on Serial1.1 dlci 16
  Queueing strategy: weighted fair
  Queueing Stats: 1/600/64/0 (size/max total/threshold/drops)
     Conversations  0/16 (active/max total)
     Reserved Conversations 0/2 (active/allocated)
  (depth/weight/discards) 1/4096/0
  Conversation 5, linktype: ip, length: 608

source: 172.21.59.21, destination: 255.255.255.255, id: 0x0006, ttl: 255,
  TOS: 0 prot: 17, source port 68, destination port 67
```

The following is sample output from the **show traffic-shape queue** command when priority queueing is configured on the map class associated with DLCI 16:

```
Router# show traffic-shape queue serial1/1 dlci 16

Traffic queued in shaping queue on Serial1.1 dlci 16
  Queueing strategy: priority-group 4
  Queueing Stats: low/1/80/0 (queue/size/max total/drops)

Packet 1, linktype: cdp, length: 334, flags: 0x10000008
```

The following is sample output from the **show traffic-shape queue** command when first-come, first-served queueing is configured on the map class associated with DLCI 16:

```
Router# show traffic-shape queue Serial1/1 dlci 16

Traffic queued in shaping queue on Serial1.1 dlci 16
  Queueing strategy: fcfs
  Queueing Stats: 1/60/0 (size/max total/drops)

Packet 1, linktype: cdp, length: 334, flags: 0x10000008
```

The following is sample output from the **show traffic-shape queue** command displaying statistics for the special queue for voice traffic that is created automatically when the **frame-relay voice bandwidth** command is entered:

```
Router# show traffic-shape queue serial 1 dlci 45

 Voice queue attached to traffic shaping queue on Serial1 dlci 45
 ~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
    Voice Queueing Stats: 0/100/0 (size/max/dropped)
    ~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
 Traffic queued in shaping queue on Serial1 dlci 45
   Queueing strategy: weighted fair
   Queueing Stats: 0/600/64/0 (size/max total/threshold/drops)
      Conversations  0/16 (active/max total)
      Reserved Conversations 0/2 (active/allocated)
```

Table 2 describes the significant fields shown in these displays.

Table 2 show traffic-shape queue Field Descriptions

Field	Description
Post h/w compression queue	Number of packets in the post-hardware-compression queue when hardware compression and Frame Relay fragmentation are configured.
Queueing strategy	When Frame Relay traffic shaping is configured, the queueing type can be weighted fair, custom-queue, priority-group, or fcfs (first-come, first-served), depending on what is configured on the Frame Relay map class for this DLCI. The default is fcfs for Frame Relay traffic shaping. When generic traffic shaping is configured, the only queueing type available is weighted fair queueing.
Queueing Stats	Statistics for the configured queueing strategy, as follows: • size—Current size of the queue. • max total—Maximum number of packets of all types that can be queued in all queues. • threshold—For weighted fair queueing, the number of packets in the queue after which new packets for high-bandwidth conversations will be dropped. • drops—Number of packets discarded during this interval.
Conversations active	Number of currently active conversations.
Conversations max total	Maximum allowed number of concurrent conversations.
Reserved Conversations active	Number of currently active conversations reserved for voice.
Reserved Conversations allocated	Maximum configured number of conversations reserved.
depth	Number of packets currently queued.
weight	Number used to classify and prioritize the packet.
discards	Number of packets discarded from queues.
Packet	Number of queued packets.
linktype	Protocol type of the queued packet (cdp = Cisco Discovery Protocol).
length	Number of bytes in the queued packet.
flags	Number of flag characters in the queued packet.
source	Source IP address.
destination	Destination IP address.
id	Packet ID.
ttl	Time-to-live count.
TOS	IP type of service.
prot	Layer 4 protocol number. Refer to RFC 943 for a list of protocol numbers (17 = User Datagram Protocol (UDP)).
source port	Port number of source port.
destination port	Port number of destination port.

Related Commands

Command	Description
show frame-relay fragment	Displays Frame Relay fragmentation details.
show frame-relay pvc	Displays statistics about PVCs for Frame Relay interfaces.
show frame-relay vofr	Displays details about FRF.11 subchannels being used on Voice over Frame Relay DLCIs.
show traffic-shape	Displays the current traffic-shaping configuration.
show traffic-shape statistics	Displays the current traffic-shaping statistics.

Glossary

compression—The running of a data set through an algorithm that reduces the space required to store or the bandwidth required to transmit the data set.

data-link connection identifier—See DLCI.

DLCI—data-link connection identifier. Value that specifies a PVC or SVC in a Frame Relay network. In the basic Frame Relay specification, DLCIs are locally significant (connected devices might use different values to specify the same connection).

fragmentation—Process of breaking a packet into smaller units when transmitting over a network medium that cannot support the original size of the packet.

Real-Time Transport Protocol—See RTP.

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

TCP—Transmission Control Protocol. Connection-oriented transport layer protocol that provides reliable full-duplex data transmission. TCP is part of the TCP/IP protocol stack.

Transmission Control Protocol—See TCP.

VC—virtual circuit. Logical circuit created to ensure reliable communication between two network devices. A virtual circuit is defined by a VPI/VCI pair and can be either permanent (PVC) or switched (SVC). Virtual circuits are used in Frame Relay and X.25. In ATM, a virtual circuit is called a virtual channel.

virtual circuit—VC