



TECHNICAL SUPPORT NOTE

Frame-Relay Rate Limiting using B_c & B_e and FRF.12 Fragmentation

Featuring NetVanta 3200 & 3205

Introduction

Frame-Relay is a data link layer (Layer 2) technology that was developed to support high-speed data transfer over a WAN network. Information is divided into frames and each frame has an address that the network uses to determine the destination. The frames could travel through a series of switches within the Frame-Relay network before they arrive at their destination. Frame-Relay networks are typically depicted as clouds, because the network is not made up of physically connected endpoints. Instead, a logical path known as a Virtual Circuit (VC) is defined within the network. Virtual Circuits are differentiated from each other using identifiers known as DLCIs (Data Link Connection Identifier). Frame-Relay circuits are also assigned a Committed Information Rate (CIR) that is enforced by the Service Provider for a guaranteed minimum amount of bandwidth. Frame-Relay circuits can be either point-to-point or point-to-multipoint. Adtran Frame-Relay was developed to facilitate point-to-point connectivity as shown in Figure 1.

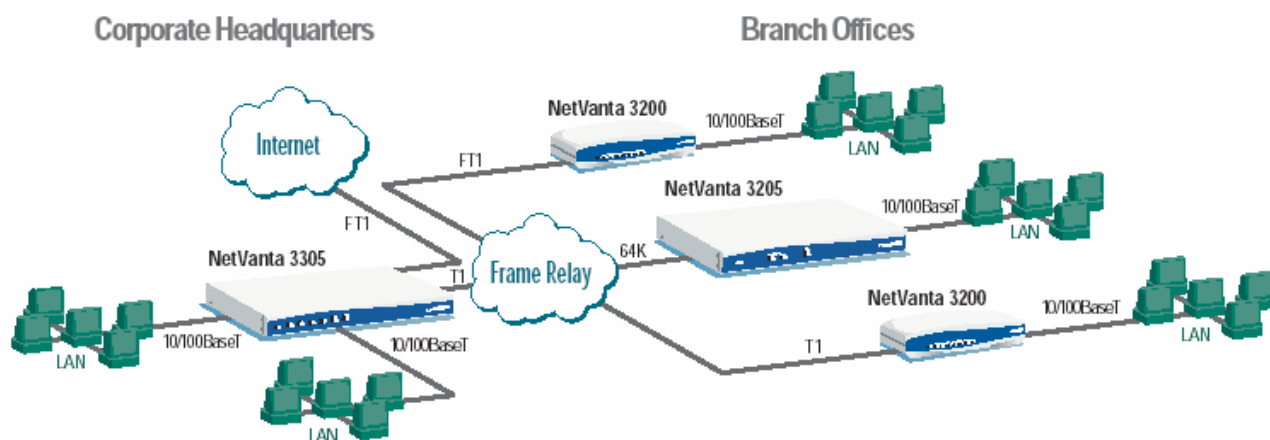


Figure 1

Rate Limiting using B_c & B_e

Rate limiting is turned off by default in Adtran routers. Rate limiting gives the Administrator more control over the virtual circuit by forcing the router to adhere to certain guidelines for transmitting packets into the Frame-Relay cloud. Rate Limiting can also be useful if there are multiple virtual circuits terminating at the router and you want to give specific virtual circuits more bandwidth privileges than others. You must set the B_c & B_e values if you want to use FRF.12 fragmentation. Your Internet Service Provider can provide you with the committed burst (B_c) and excessive burst (B_e) values of your Frame-Relay circuit. B_c is commonly set to the CIR (Committed Information Rate) of the PVC. The sum of B_c & B_e should not exceed the port speed at either side of the PVC.

Configuring B_c

The command syntax is as follows:

frame-relay bc <committed burst value>

Use the **frame-relay bc** command to set the bc (committed burst) value for a frame relay sublink. The value is in bits. Use the **no** form of this command to return to default.

Syntax Description

<committed burst value> Enter the committed burst value (in bits) for the sublink.

Functional Notes

The time interval is always one second, so this can also be considered bits per second. Shaping is performed on a sliding one-second window to make maximum use of configured bandwidth. Note that when both **bc** and **be** are non-zero, shaping is performed on the virtual circuit. The circuit is limited to the sum of **bc** and **be**, and it is recommended that the sum always be greater than 8000.

Usage Examples

The following example configures sublink fr 1.1 with a committed burst value of 128000 bits:

```
(config)# interface fr 1.1  
(config-fr 1.1)# frame-relay bc 128000
```

Configuring B_e

The command syntax is as follows:

frame-relay be <excessive burst value>

Use the **frame-relay be** command to set the be (excessive burst) value for a frame relay sublink. The value is in bits. Use the **no** form of this command to return to default.

Syntax Description

<committed burst value> Enter the excessive burst value (in bits) for the sublink.

Functional Notes

The time interval is always one second, so this can also be considered bits per second. Shaping is performed on a sliding one-second window to make maximum use of configured bandwidth. Note that when both **bc** and **be** are non-zero, shaping is performed on the virtual circuit. The circuit is limited to the sum of **bc** and **be**, and it is recommended that the sum always be greater than 8000.

Usage Examples

The following example configures sublink fr 1.1 with an excessive burst value of 64000 bits:

```
(config)# interface fr 1.1  
(config-fr 1.1)# frame-relay be 64000
```

For virtual circuits carrying voice packets, set B_e to a low value to ensure the best possible voice quality. The router only bursts when there are tokens in the token bucket. The token bucket does not accrue tokens unless the amount of traffic being sent out is less than the CIR.

FRF.12 Fragmentation

FRF.12 defines fragmentation of data frames when voice frames are present. Fragmentation is normally not necessary on high-speed links (768Kbps or greater). FRF.12 fragmentation allows the router to break long frames into a sequence of shorter frames, which are then reassembled into the original frame by the receiving device. Fragmentation is necessary to control delay and delay variation when voice traffic is carried across the same interfaces as data. The size of data fragments is configured based upon the bandwidth of the link at the slower side of the Frame-Relay network. In order to support traffic on lower-speed links, long frames should be fragmented to ensure that the shorter frames are not excessively delayed. Fragmentation enables interleaving voice and data traffic.

FRF.12 Fragmentation Size

The question that always arises when performing fragmentation: "How big should the fragments be?" The answer should be based on the bandwidth of the slower side of the Frame-Relay link. It is rare for all Frame-Relay links to be provisioned with the same bandwidth. Usually, a central site will have the highest bandwidth, with each remote branch having a bandwidth based on the type and volume of traffic expected to traverse the link. It is imperative that the fragment size is based upon the slower side of the Frame-Relay link. Figure 2 shows the recommended fragmentation sizes.

Recommended FRF.12 Fragmentation Size

Link Speed (in Kbps)	Fragment Size (in Bytes)
56	70
64	80
128	160
256	320
512	640
768	1000
1536	N/A

Figure 2

The faster the link, the less the need for fragmentation exists. It is not recommended that fragmentation be implemented on links with more than 768Kbps. The fragmentation size is set on a per-virtual circuit basis; therefore, the size should be altered for each virtual circuit based on the bandwidth available to the slower side of the Frame-Relay link.

Configuring FRF.12 Fragmentation

The command syntax is as follows:

frame-relay fragment <threshold>

Use the **frame-relay fragment** command to set the FRF.12 fragmentation threshold. Use the **no** form of this command to erase the configured threshold.

Syntax Description

<threshold> Valid fragmentation thresholds are greater than or equal to 64 and less than or equal to 1600.

Functional Notes

For frame-relay fragmentation to take effect, rate-limiting must be enabled by setting the committed burst rate and excessive burst rate.

Usage Examples

enables FRF.12 fragmentation on a sublink:

```
(config)#interface frame-relay 1.1  
(config-fr 1.1)#frame-relay bc 256000  
(config-fr 1.1)#frame-relay be 1  
(config-fr 1.1)#frame-relay fragmentation 320
```

disables FRF.12 fragmentation on a sublink:

```
(config)#interface frame-relay 1.1  
(config-fr 1.1)#no frame-relay fragment
```

Verifying the settings

show frame-relay fragment [frame-relay <port.sublink>]

Use the **show frame-relay fragment** command to display FRF.12 statistics for frame-relay sublinks enabling FRF.12 fragmentation.

Syntax Description

frame-relay <port.sublink> Displays detailed FRF.12 statistics for the specified frame-relay sublink (if FRF.12 is **Optional* enabled on that sublink).

Usage Examples

The following are sample outputs from various **show frame-relay fragment** commands:

> **enable**

show frame-relay fragment

```
interface dlcI frag_size rx_frag tx_frag dropped_frag
```

```
fr 1.1 17 100 46 48 0
```

```
fr 1.2 18 200 42 21 0
```

> **enable**

show frame-relay fragment frame-relay 1.1

DLCI = 17 FRAGMENT SIZE = 100

rx frag. pkts 46 tx frag. pkts 48

rx frag. bytes 4598 tx frag. bytes 4724

rx non-frag. pkts 18 tx non-frag. pkts 28

rx non-frag. bytes 1228 tx non-frag. bytes 1960

rx assembled pkts 23 tx pre-fragment pkts 34

rx assembled bytes 5478 tx pre-fragment bytes 6324

dropped reassembling pkts 0 dropped fragmenting pkts 0

rx out-of-sequence fragments 0

rx unexpected beginning fragment 0

show interfaces fr 1.1*

fr 1.1 is Active

Ip address is 63.97.45.57, mask is 255.255.255.248

Interface-dlcI is 100

MTU is 1500 bytes, BW is 96000 Kbit (limited)

Average utilization is 53%

*Note: If the user has configured a **B_c** and **B_e** value on the virtual circuit, the bandwidth (**BW**) displayed is the sum of those values (Bc + Be). If not, the value for **BW** is the speed of the interface. The **Average utilization** displayed is the average utilization of the displayed bandwidth. If the bandwidth number is the Bc + Be value, the (**limited**) text appears (as shown above).

If you experience any problems using your ADTRAN product, please contact [ADTRAN Technical Support](#)

DISCLAIMER

ADTRAN provides the foregoing application description solely for the reader's consideration and study, and without any representation or suggestion that the foregoing application is or may be free from claims of third parties for infringement of intellectual property rights, including but not limited to, direct and contributory infringement as well as for active inducement to infringe. In addition, the reader's attention is drawn to the following disclaimer with regard to the reader's use of the foregoing material in products and/or systems. That is:

ADTRAN SPECIFICALLY DISCLAIMS ALL WARRANTIES, EXPRESSED OR IMPLIED, INCLUDING BUT NOT LIMITED TO, MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. IN NO EVENT SHALL ADTRAN BE LIABLE FOR ANY LOSS OR DAMAGE, AND FOR PERSONAL INJURY, INCLUDING BUT NOT LIMITED TO, COMPENSATORY, SPECIAL, INCIDENTAL, CONSEQUENTIAL, OR OTHER DAMAGES.