Data Throughput in the Balance; Optimizing Your Data Transfer Rate over Multilink Frame Relay Using Fragmentation

By
Roger J. Ruby, Sr. Product Manager
Gilbert Sauvé, Sr. Sales Engineer
Quick Eagle Networks

1.0 Introduction

When one thinks of efficiently accomplishing a large task, an old saying comes to mind: how do you eat an elephant? Answer: one bite at a time. This adage has never been truer than when dealing with data transfer rates over a multi-link bundled transport mechanism, such as multilink frame relay (MFR). Long data frames, interspersed with short frames, can rob the end user of the full logical link bandwidth supported on their MFR bundle. The Frame Relay Forum has provided an elegant solution to this issue with the FRF.12 Implementation Agreement. This white paper will cover the techniques of frame relay fragmentation and how these methods can yield the maximum data throughput possible over an MFR logical bundle of NxT1 or E1 links. The following topics will be discussed:

- Review of Frame Relay/Multilink Frame Relay Technologies
- The basic “mechanics” of Fragmentation as defined by FRF.12
- An MS Windows-based application example using FR fragmentation

2.0 Frame Relay / Multilink Frame Relay Technologies Review

Frame relay (FR) technology has been enjoying a tremendous run rate over the past several years. Starting off as replacement for older X.25 packet data technology, FR allows a customer to have virtual point-to-point or point to multi-point connections at a fraction of the cost of a dedicated facility. This is particularly beneficial when considerable geographical distances separate the customers’ sites. Frame Relay is a layer 2 protocol. With Frame Relay a customer shares the FR backbone bandwidth with other subscribers. Each customer is given DLCI (Data Link Control Identifier) assignments and is assigned a committed amount of bandwidth or Information Rate (CIR) with an opportunity to “burst” above that rate, as required, for each DLCI. Essentially, frame relay service providers statistically multiplex their subscribers onto shared facilities between POPs.

Bandwidth 'Rules'

When frame relay services were first rolled out, 56Kbpsec access lines from the customer to the frame relay POP were considered more than adequate. Then fractional and full T1/E1 tariffs became more favorable, thereby allowing the customer to support CIR's approaching several hundred Kbits per second.

A multi-link solution, for frame relay, is becoming part of the 'next wave' of service delivery solutions for the enterprise. CIR's in the megabits-per-second can now be offered in FR SLA's (Service Level Agreements). Multi-link frame relay service is a standards-based solution that implements a CPE MFR device that can inter-operate with the major providers of frame relay infra-structures.

MFR FRF.16 forms a logical high-speed bundle from multiple T1's or E1's. Used at a corporate headquarter site, this high-speed bundle effortlessly carries the aggregate traffic from/to remote corporate locations. In addition, the MFR protocol manages the adding or subtracting of links in the bundle, as local loop conditions or customer requirements dictate. The switch-end and CPE-end exchanging MFR Link Protocol Control messages accomplish this.

Refer to Figure 1, on the following page, which illustrates an MFR FRF.16 implementation. Note that the FR switches need to be equipped with FRF.16 software support. The main advantage to this approach is that it has lower latency because all of the T1's/E1's are treated as a single MFR bundle and will, therefore, be terminated at the nearest POP. This will be very beneficial for those applications, running over FR, that cannot tolerate excessive latency, such as voice over frame.

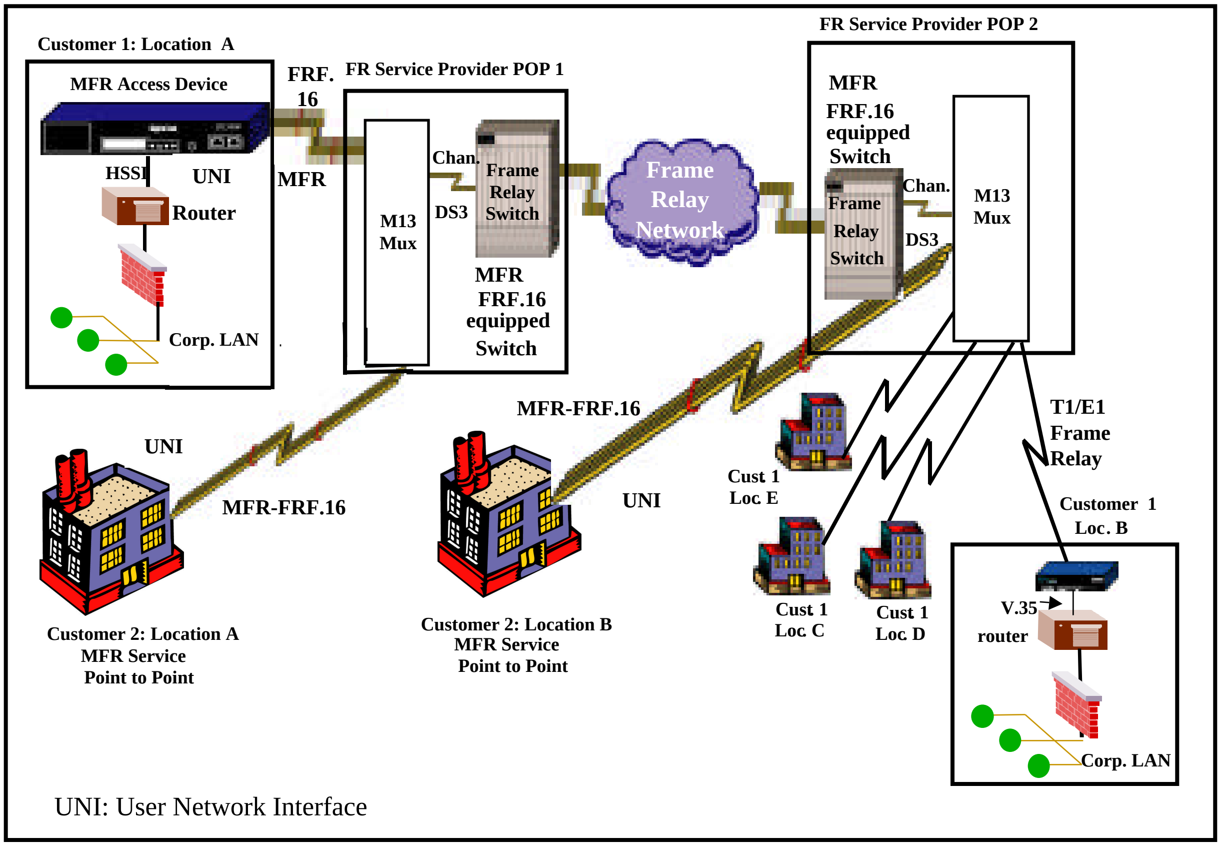

Figure 1: Typical MFR FRF.16 topology

Note: that the connection into the MFR device, on the CPE side is serial-port based since a router device will be required “behind” the MFR device..

3.0 Why Fragmentation

Fragmentation, as defined by the Frame Relay Forum’s FRF.12 Implementation Agreement, sends data frames over an MFR “bundle” after being partitioned into near equal lengths so that the individual links can be evenly loaded with data. To more fully utilize the aggregate bandwidth capacity of the MFR bundle, it is necessary to fragment long frames that share the same UNI with shorter frames so that the shorter frames are not excessively delayed. Fragmentation facilitates interleaving delay-sensitive traffic on one Virtual Circuit (VC) with fragments of a long frame on another VC utilizing the same physical interfaceA secondary benefit to fragmentation is improved voice packet delay and no additional jitter because the resulting stream will closely match the stream flowing into the MFR DSU at the other end.

So what’s the big deal?

The MFR algorithm places data frames onto each T1/E1 link in a “round-robin” fashion. Referring to Figure 2, below, notice that frames of varying lengths are being packed, rather unevenly, on the 8 links that comprise the multilink bundle. This means that the shorter frames will arrive at the UNI earlier than the longer frames. The receiving end then has to hold the shorter frames in buffers until the longer frames arrive before they are passed onto either the CPE router, at the customer site, or the FR backbone on the carrier/service provider side. Short frames sent after a long frame, in a sequence of frames on a PVC (Private Virtual Circuit), may be received before the longer frame. As MFR guarantees the integrity of the sequencing, the shorter frame will be held until the longer preceding frame has been forwarded out of the MFR unit. This phenomenon will introduce more jitter as an inter-frame delay is introduced. The net result is that it takes longer to complete the entire data session because of the inconsistent wait times at each end of MFR bundle.

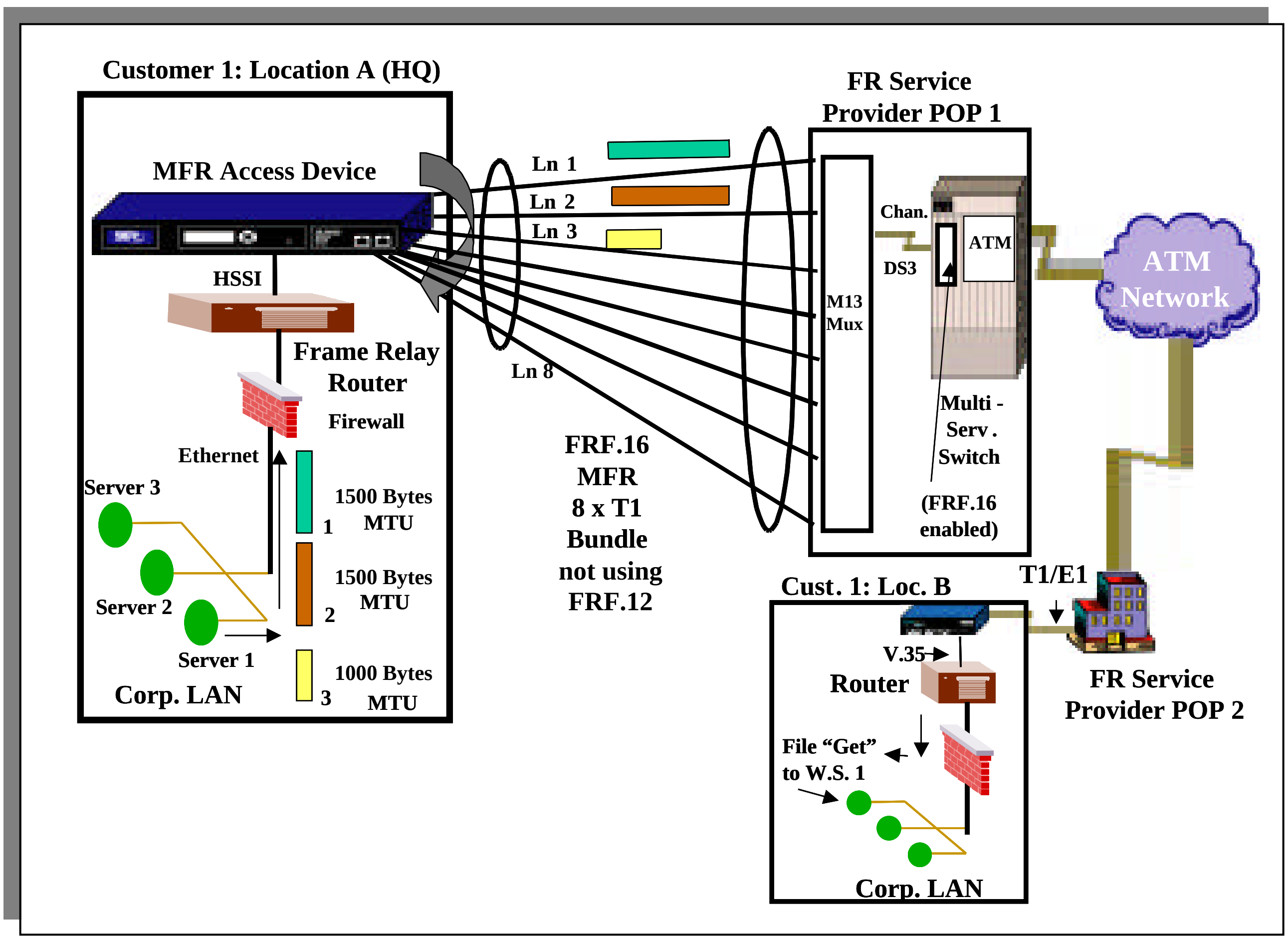

Figure 2: FRF.16 Round-Robin Link Loading *without* FRF.12

4.0 The “Mechanics” of Fragmentation

In some implementations, fragmentation is realized by an ***optional*** fragmentation-processing step being added ***prior*** to the multi-link function. The MFR device or interface may have a threshold setting configured that will be the “trigger-point” at which fragmentation takes place along with the setting of the fragment size . For example, with the “trigger-point” at 512 bytes and a fragment size of 256 bytes, a 512 byte frame would result in 2*256 byte fragments.

One 256-byte fragment could, therefore, be placed on two of the four T1/E1 lines. An additional FR fragmentation “header” is placed onto each fragment, with a fragment sequence number, so that the entire frame can be re-assembled, in the proper order, at the receiving end. See Figure 3, below, for the structure of a UNI FR fragment as defined in FRF12.

	8	7	6	5	4	3	2	1
Fragmentation header	B	E	C	Seq. # high 4 bits				1
	Sequence # low 8 bits							
Frame Relay header	DLCI high six bits						C/R	0
	DLCI low 4 bits				F	B	DE	1
	Fragment Payload							
	FCS (two octets)							

Figure 3: UNI Frame Fragment Structure

4.0 An MS Windows™ Application Example

When an enterprise connects to an N x T1 service they expect to receive NxT1 throughput. This is not always the case if the inverse multiplexing equipment being deployed does not use a fragmentation algorithm. To demonstrate the advantages of fragmentation lets look at two FTP scenarios, the first **without** fragmentation and the second **with** fragmentation.

Example 1- FTP, using TCP/IP, without FRF.12 Fragmentation:
-Windows 98 default TCP Window size of 4000 Bytes
- Windows 98 default MTU (Max. Transmission Unit) of 1500 bytes
- 8 XT1 circuits between the CPE MFR device and Frame Relay Switch.

Note: Refer back to Figure 2

In this first scenario, server # 1, Location A sends 4000 bytes and waits for the acknowledgment to come back, from the Workstation (W.S. #1) located at Location B. A second round of 4000bytes then follows. Because the Ethernet MTU is set to 1500 bytes, the server sends out 3 frames, two of 1500 bytes and one of 1000 bytes. Given that fragmentation is disabled or not available, the three frames received for transmission by the CPE equipment will be put in the least busy output queues of the T1 ports. So each frame will traverse a separate T1 port using a total of 3 T1 ports. The time it will take to get the 4000 bytes across the bundle will be (1500 bytes*8 bit/byte)/1.5Mbps=7.8ms for an effective speed of (4000 Bytes * 8bits /bytes)/7.8ms=4.1Mbps. Here, we are assuming that the TCP ACK time is negligible. Therefore, there will never be more than 3 X T1's used in this scenario even if there are 8 T1 ports between the units. This means that your throughput will never exceed the aggregated rate of 3 T1s.

Example 2- FTP, using TCP/IP with Fragmentation Enabled
Fragmentation is set for 125 Bytes per fragment:

- Windows 98 default TCP window size of 4000Bytes
- Windows 98 default Ethernet MTU of 1500 bytes
- 8 X T1 circuits between CPE MFR device and Frame Relay Switch
  Note: Refer to Figure 4

As in example 1, server # 1 will send out 4000 bytes and waits for the acknowledgment before sending out another round of 4000 bytes. These, again, will be sent out as 2 * 1500 byte frames plus one 1000 byte frame. But now, as the first frame is received at the serial port of the CPE MFR device, it is "fragmented" into 12 * 125 byte frames and then queued up sequentially on the least busy T1 ports. The same occurs for the remaining two frames, and we end up with 4000 bytes/125 byte/frame=32 frames where we will have 4 * 125 bytes sent over each of the 8 T1 links. The time it takes to send out these 4 * 125 byte fragments over a T1 link is (4*125 bytes * 8 bits/byte) / 1.5Mbps=2.7mS. Again, given that the "Ack" time is negligible, the throughput of the bundle will be: (4000 bytes * 8 bits/byte) / 2.7mS = approx. 12.0 Mbps.

With fragmentation enabled, the full capacity of the bundle is used even if the default window size and MTU of Windows 98™ has not been optimized for transmission over an 8 X T1 link MFR bundle.

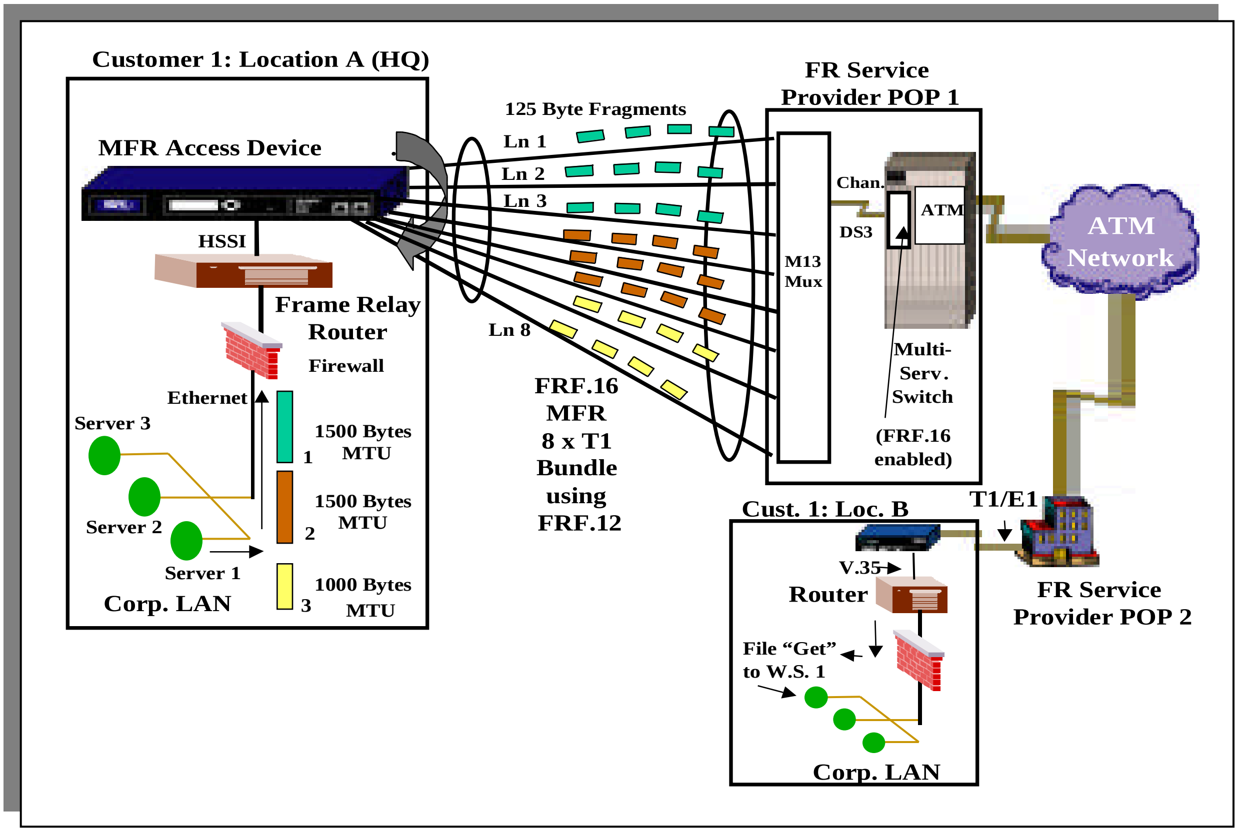

Figure 4: FRF.16 Round-Robin Link Loading *with* FRF.12

5.0 Summary

Fragmentation is required for an application not having an MTU large enough to put a frame on every T1 in the bundle. The MS Windows IP stack is set at about 4000 byes. This means the fragmentation should be strongly considered for bundles of 3 links or more but fragmentation may have little or no impact on MFR bundles of 2 links.

Several Frame Relay Forum (FRF) vendor and service provider companies are staunch supporters of standards-based Frame Relay CPE and POP solutions. FRF.12-based solutions optimizes a customer’s monthly investment in facility line charges when deploying a multilink solution to the serving POP. The frame relay service provider can easily configure both fragmentation threshold and fragment size at time of service deployment for both the CPE and the switch end. Typically, a 125 to 140 byte fragment size will provide, in many cases, an optimal setting for keeping those NxT1/E1 “pipes” filled with data even under the heaviest data load conditions. This equates to faster throughput and maximum return on investment.

For detailed technical information on FRF.12 and FRF.16, as well as other Frame Relay Forum Implementation Agreements, access the following FRF web link:

http://www.frforum.com/5000/5000index.html