Configuring Link Fragmentation and Interleaving (LFI) With Campus ATM Switches

Contents

Introduction

This document provides a technical overview of Link Fragmentation and Interleaving (LFI) over a Frame Relay to ATM Interworking (IWF) connection (as defined by the Frame Relay Forum or FRF.8 agreement), as well as a sample configuration for using the LS1010 or Catalyst 8500 as the IWF device in the WAN cloud. LFI uses the built-in fragmentation capabilities of multilink point-to-point protocol (MLPPP) encapsulation over ATM and Frame Relay to provide an end-to-end fragmentation and interleaving solution for low-speed links with bandwidths of up to 768 kbps.

Prerequisites

Requirements

This document requires an understanding of the following:

- Typical FRF.8 environment and FRF.8 transparent and translation modes - See [Understanding Transparent and Translation Modes With FRF.8](#).
- Familiarity with LS1010 and Catalyst 8500 configuration commands and how either the [Channelized E1 Frame Relay Port Adapter](#) or the [Channelized DS3 Frame Relay Port](#)

Adapter performs interworking between a Frame Relay endpoint and an ATM endpoint.
- Serialization delay and jitter. See [VoIP over PPP Links with Quality of Service (LLQ / IP RTP Priority, LFI, cRTP)](#) and [VoIP over Frame Relay with Quality of Service (Fragmentation, Traffic Shaping, IP RTP Priority)](#).

Components Used

This document is not restricted to specific software and hardware versions.

Conventions

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

Why MLPPP over ATM and Frame Relay?

Fragmentation is a key technique for controlling serialization delay and delay variation on low-speed links carrying both real-time and non-real-time traffic. Serialization delay is the fixed delay required to clock a voice or data frame onto the network interface, and it is directly related to the clock rate on the trunk. An extra flag is needed to separate the frames for low clock speeds and small frame sizes.

LFI uses the built-in fragmentation capabilities of MLPPP to prevent delay and jitter (variations in delay) caused by variable-sized large packets being queued in between relatively small voice packets. With LFI, packets larger than a configured fragment size are encapsulated in an MLPPP header. [RFC 1990](#) defines the MLPPP header as well as the following:

- (B)eginning fragment bit is a one bit field set to 1 on the first fragment derived from a PPP packet and set to 0 for all other fragments from the same PPP packet.
- (E)nding fragment bit is a one bit field set to 1 on the last fragment and set to 0 for all other fragments.
- The sequence field is a 24-bit or 12-bit number that is incremented for every fragment transmitted. By default, the sequence field is 24 bits long, but can be negotiated to be only 12 bits with the LCP configuration option described below.

In addition to fragmentation, delay-sensitive packets must be scheduled with adequate priority between fragments of a big packet. With fragmentation, Weighted Fair Queueing (WFQ) becomes "aware" of whether a packet is part of a fragment or is unfragmented. WFQ assigns a sequence number to each arriving packet and then schedules packets based on this number.

Layer-2 fragmentation provides a superior solution to all other approaches in solving the "big-packet problem." The following table lists the advantages and disadvantages of other potential solutions.

Potential Solution	Advantages	Disadvantages
Abort transmission of the big packet and **re-queue** it behind the	- Only postpones packet transmission. - When the packet is retransmitted,	- Some physical interfaces do not support aborted

delay sensitive traffic.	the same problem can occur. If the packets are continually requeued and even dropped, bandwidth starvation may result.	transmission or introduce a performance penalty for doing so (such as resetting the entire transmit queue).
Fragment the big packet using **network-layer fragmentation** techniques.	• Both IP and CLNP support fragmentation at any router, with reassembly occurring at the destination host. • Can avoid the need to fragment the big packet with MTU discovery. • Uses a global mechanism to overcome what is essentially a local (one-hop) problem - all the downstream hops must deal with a larger number of packets to switch, even if all the subsequent links are fast. • Voids the option of TCP/IP header compression.	• Many applications do not accept fragmentation and set the "Do Not Fragment" bit in the IP header. These packets will be dropped if fragmented. Applications that are not capable of accepting fragmented packets will be rendered inoperable in this environment.
Fragment the packet using **link-layer** techniques.	• Supported with any network layer packet or bridged packet.	• Provides per-link fragmentation rather than requiring

		fragmented packets to be transported end-to-end. Only the routers attached to the slow link need to accommodate the handling and reassembly of additional packets.

The ideal fragment size for multilink point-to-point protocol over ATM (MLPPPoATM) should allow the fragments to fit into an exact multiple of ATM cells. See Link Fragmentation and Interleaving for Frame Relay and ATM Virtual Circuit for guidance on selecting fragmentation values.

MLPPPoA and MLPPPoFR Headers

A typical configuration of FRF.8 consists of the following:

- A Frame Relay endpoint
- An ATM endpoint
- An interworking (IWF) device

Each endpoint encapsulates data and voice packets in a layer-2 encapsulation header, which communicates the protocol encapsulated and transported in the frame or cell. Both Frame Relay and ATM support Network Layer Protocol ID (NLPID) encapsulation headers. The ISO/International Electrotechnical Commission (IEC) TR 9577 document defines well-known NLPID values for a select number of protocols. A value of 0xCF is assigned to PPP.

RFC 1973 defines PPP in Frame Relay and the MLPPPoFR header, while RFC 2364 defines PPP over AAL5 and the MLPPPoA header. Both headers use an NLPID value of 0xCF to identify PPP as the encapsulated protocol.

Each of these headers is illustrated in Figure 1 below.

'vanilla' PPP over AAL5 with NLPID encapsulation (non-fragmented)

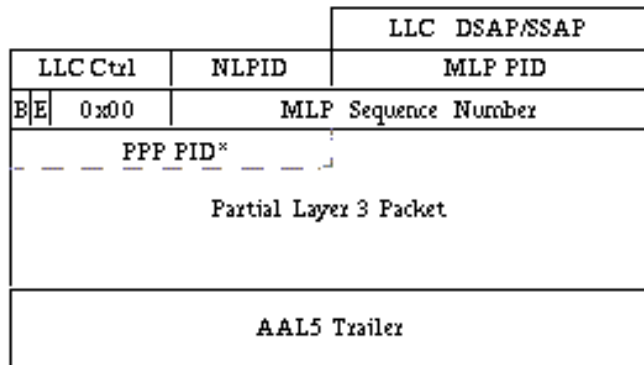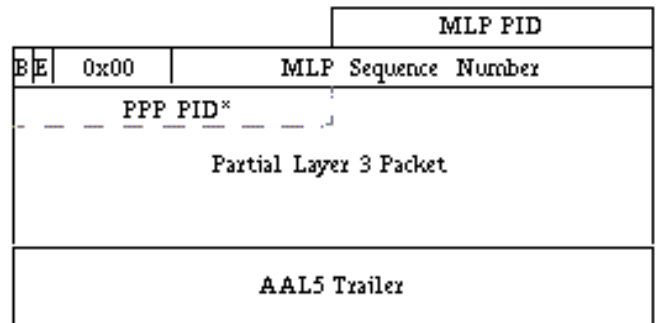| LLC Ctrl	NLPID	LLC DSAP/SSAP
		PPP PID

Layer 3 Packet

AAL5 Trailer

MLP Header

MLPPP over AAL5 with NLPID encapsulation (fragment)

LLC Ctrl	NLPID	LLC DSAP/SSAP
		MLP PID

| B | E | 0x00 | MLP Sequence Number |

PPP PID*

Partial Layer 3 Packet

AAL5 Trailer
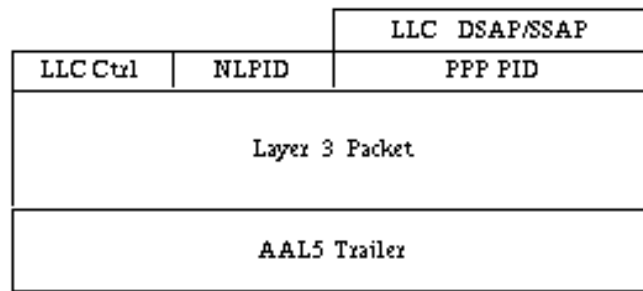
MLPPP over AAL5 with VC multiplexing (fragment)

MLP PID

| B | E | 0x00 | MLP Sequence Number |

PPP PID*

Partial Layer 3 Packet

AAL5 Trailer

Figure 1. *PPP over AAL5 header, MLPPPoA header with NLPID encapsulation, and MLPPPoA header with VC multiplexing*

Note: The MLPPPoFR header also includes a one-byte flag field of 0x7e, which is not shown in Figure 1. After the headers, byte number 5 starts the PPP or MLPPP protocol fields.

Table 1 - *FRF.8 Transparent vs. FRF.8 Translational.*

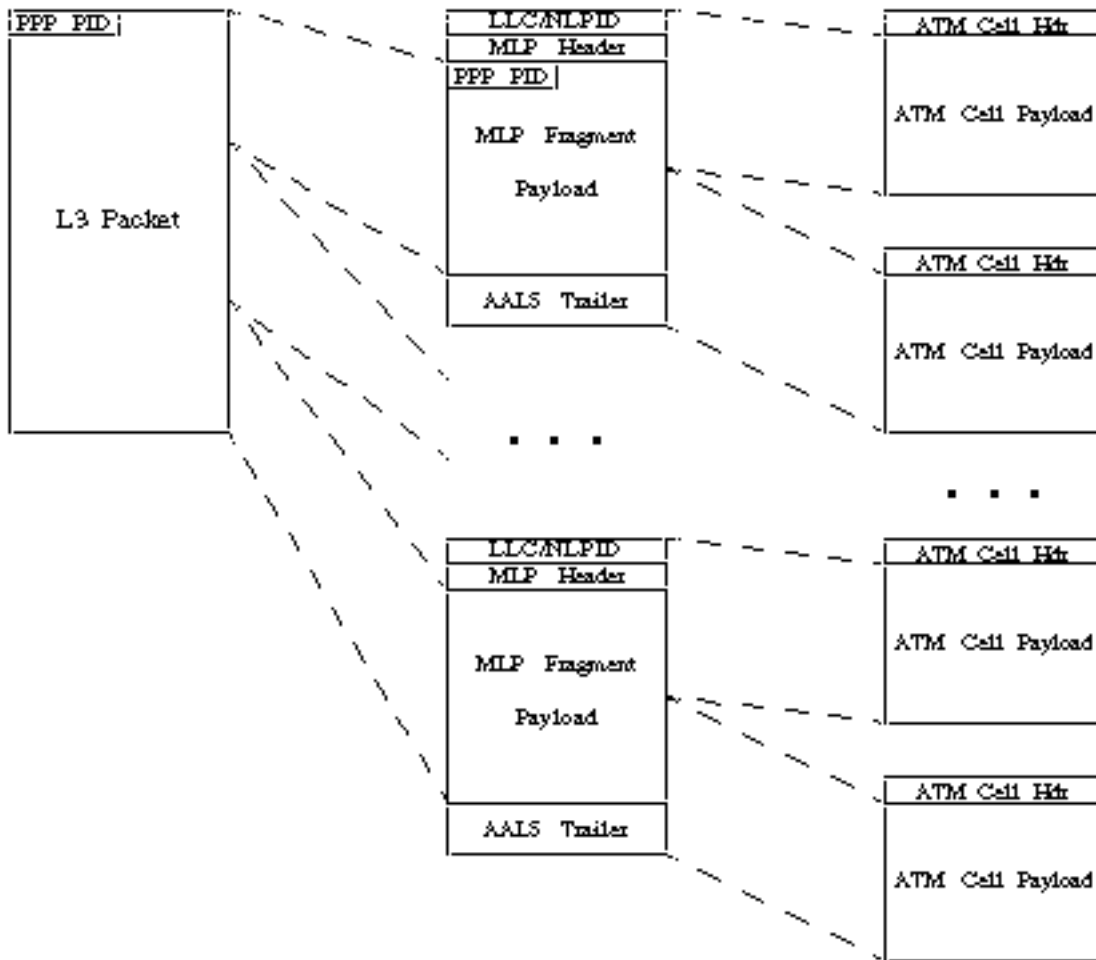| Header	VC Muxed	LLC/NLPID Encapsulation			
		FRF8 Transparent		FRF8 Translational	
		ATM RX	ATM TX	ATM RX	ATM TX
LLC DSAP (0xfe)			✔	✔	✔
LLC SSAP (0xfe)			✔	✔	✔
LLC Ctrl (0x03)		✔	✔	✔	✔
NLPID (0xcf)		✔	✔	✔	✔
MLP PID (0x003d)	✔	✔	✔	✔	✔
MLP BE/Seq # (0xX0XXXXXX)	✔	✔	✔	✔	✔

Figure 2. *How the MLPPPoATM packet is fragmented using NLPID.*
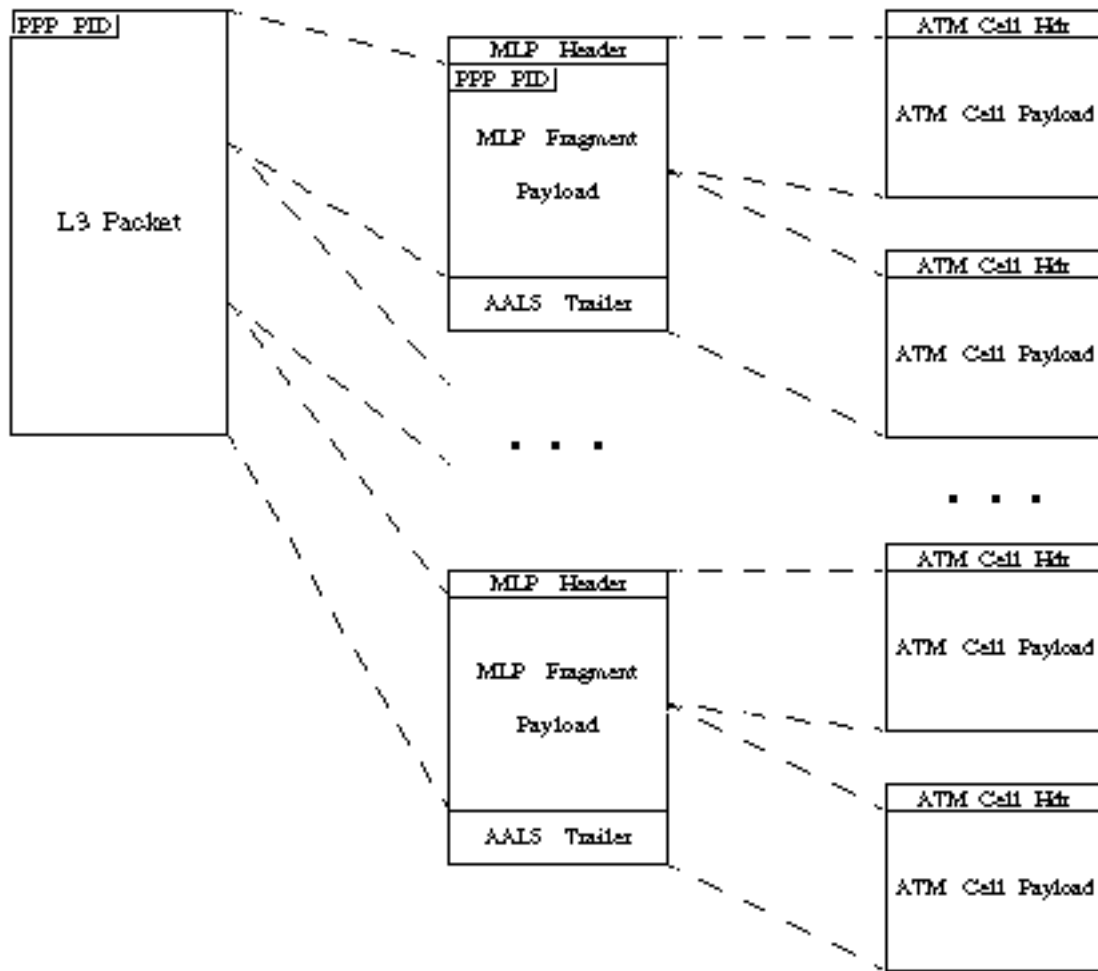
Figure 3. *How the MLPPPoATM packet is fragmented using VC Multiplexing.*

```
byte #                          byte #
        ++++++++++++                    ++++++++++++
  1     +    DLCI    +           1     +    0xFE    +
        ++++++++++++                    ++++++++++++
  2     +    DLCI    +           2     +    0xFE    +
        ++++++++++++                    ++++++++++++
  3     +    0x03    +           3     +    0x03    +
        ++++++++++++                    ++++++++++++
  4     +    0xCF    +           4     +    0xCF    +
        ++++++++++++                    ++++++++++++
  5     +          +             5     +          +
        +          +                   +          +
  .     +          +             .     +          +
  .     +          +             .     +          +
  .     +          +             .     +          +
  .     +          +             .     +          +
        +          +                   +          +
  n     +          +             n     +          +
        ++++++++++++                    ++++++++++++
```
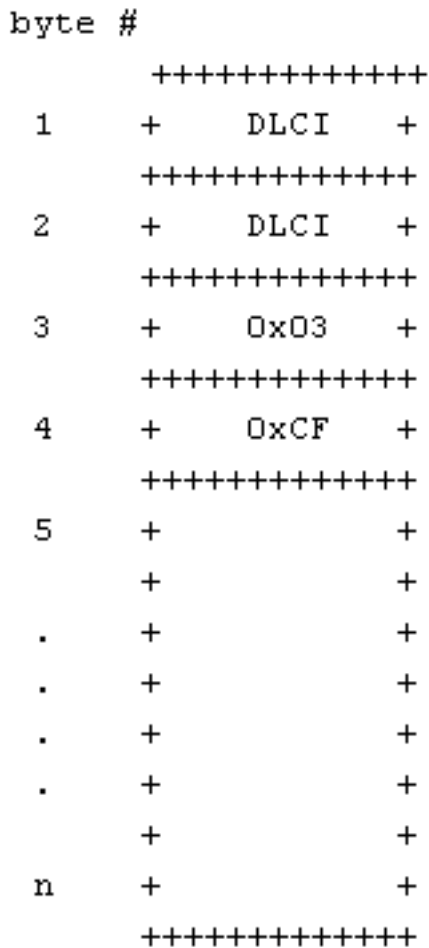
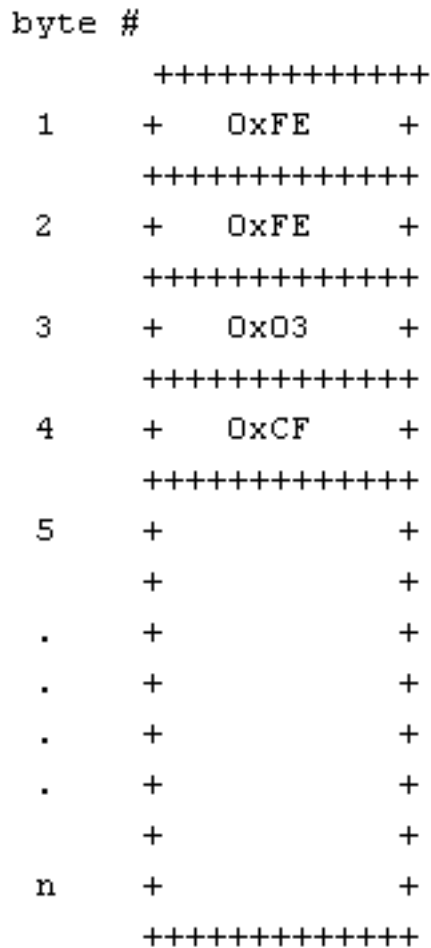**Figure 4.** *MLPoFR Header*   **Figure 5.** *MLPoATM Header*

The meaning of the byte values are shown below:

- 0xFEFE - Identifies the destination and source service access points (SAPs) in the Logical Link Control (LLC) header. A value of 0xFEFE indicates that what follows next is a short-form NLPID header, which is used with protocols having a defined NLPID value.
- 0x03 - Control field used with many encapsulations, including High Level Data Link Control (HDLC). Also indicates that the contents of the packet consist of unnumbered information.
- 0xCF - Well-known NLPID value for PPP.

FRF.8 Transparent vs Translation Modes

The FRF.8 agreement defines two operational modes for the IWF device:

- Transparent - IWF device forwards the encapsulation headers unaltered. It does not perform any protocol-header mapping, fragmentation or reassembly.
- Translation - IWF device performs protocol-header mapping between the two encapsulation headers to account for small differences between the encapsulation types.

The mode configured on the IWF device, which can be a Cisco ATM campus switch or a 7200 Series router with a PA-A3 ATM port adapter, changes the number of layer-2 header bytes on the ATM and Frame Relay segments of the interworking link. Let's look at this overhead in more detail.

The following two tables show the overhead bytes for data packets and voice over IP (VoIP) packets.

Table 2 - *Data link overhead in bytes for a data packet over an FRF.8 link.*

FRF.8 Mode	Transparent				Translation			
Traffic Direction	Frame Relay to ATM		ATM to Frame Relay		Frame Relay to ATM		ATM to Frame Relay	
Frame Relay or ATM leg of PVC	Frame Relay	ATM	ATM	Frame Relay	Frame Relay	ATM	ATM	Frame Relay
Frame Flag (0x7e)	1	0	0	1	1	0	1	0
Frame Relay header	2	0	0	2	2	0	0	2
LLC DSAP/SSAP (0xfefe)	0	0	2	2	0	2	2	0
LLC Control (0x03)	1	1	1	1	1	1	1	1
NLPID (0xcf for PPP)	1	1	1	1	1	1	1	1
MLP Protocol ID (0x003d)	2	2	2	2	2	2	2	2
MLP Sequence number	4	4	4	4	4	4	4	4
PPP Protocol ID (1st frag only)	2	2	2	2	2	2	2	2
Payload (Layer 3+)	0	0	0	0	0	0	0	0
ATM Adaptation Layer (AAL)5	0	8	8	0	0	8	8	0
Frame	2	0	0	2	2	0	0	2

Check Sequence (FCS)								
Total Overhead (bytes)	15	18	20	17	15	20	20	15

Table 3 - *Data link overhead in bytes for a VoIP packet over an FRF.8 link.*

FRF.8 Mode	Transparent				Translation				Frame Relay to Frame Relay
Traffic Direction	Frame Relay to ATM		ATM to Frame Relay		Frame Relay to ATM		ATM to Frame Relay		
Frame Relay or ATM leg of PVC	Frame Relay	ATM	ATM	Frame Relay	Frame Relay	ATM	ATM	Frame Relay	
Frame Flag (0x7e)	1	0	0	1	1	0	0	1	1
Frame Relay Header	2	0	0	2	2	0	0	2	2
LLC DSAP/SSAP (0xfefe)	0	0	2	2	0	2	2	0	0
LLC Control (0x03)	1	1	1	1	1	1	1	1	1
NLPID (0xcf for PPP)	1	1	1	1	1	1	1	1	1
PPP ID	2	2	2	2	2	2	2	2	0
Payload (IP+User Datagram Protocol (UDP)+RTP+Voice)	0	0	0	0	0	0	0	0	0
AAL5	0	8	8	0	0	8	8	0	0
FCS	2	0	0	2	2	0	0	2	2

Total Overhead (bytes)	9	12	14	11	9	14	14	9	7

In reviewing the tables above, note the following:

- Packets smaller than the specified fragmentation size are encapsulated only in a PPP header and not in an MLPPP header. Similarly, packets larger than the specified fragmentation size are encapsulated in both a PPP header and an MLPPP header. Thus, VoIP packets have up to eight bytes less of overhead.
- Only the first Multilink PPP (MLP) fragment includes a PPP Protocol ID field. Thus, the first fragment carries two extra byes of overhead.
- In transparent mode, the encapsulation headers are passed unchanged through the IWF device. Thus, the overhead varies *in each direction and on each segment*. Specifically, an MLPPPoA header starts with a short-form NLPID header of 0xFEFE. In transparent mode, this header is passed unchanged by the IWF device from the ATM segment to the Frame Relay segment. However, in the Frame Relay to ATM direction, no such header exists in transparent mode on either segment.
- In translation mode, the IWF device changes the encapsulation headers. Thus, the overhead is the same *on each segment in either direction*. Specifically, in the ATM to Frame Relay direction, the ATM endpoint encapsulates the packet in an MLPPPoA header. The IWF device removes the NLPID header before passing the remaining frame to the Frame Relay segment. In the Frame Relay to ATM direction, the IWF device again manipulates the frame and prepends an NLPID header before passing the segmented frame to the ATM endpoint.
- When designing FRF links with MLP, be sure to account for the correct number of data link overhead bytes. Such overhead influences the amount of bandwidth consumed by each VoIP call. It also plays a role in determining the optimum MLP fragment size. Optimizing the fragment size to fit an integral number of ATM cells is critical, particularly on slow-speed PVCs where a significant amount of bandwidth can be wasted on padding the last cell to an even multiple of 48 bytes.

For clarity purposes, let's walk through the steps of the packet encapsulation process when a packet goes in the Frame Relay to ATM direction with transparent mode:

1. The Frame Relay endpoint encapsulates the packet in an MLPPPoFR header.
2. The IWF device removes the two-byte Frame Relay header with the Data Link Connection Identifier (DLCI). It then forwards the remaining packet to the IWF's ATM interface, which segments the packet into cells and forwards it across the ATM segment.
3. The ATM endpoint examines the header of the received packet. If the first two bytes of the received packet are 0x03CF, the ATM endpoint considers the packet to be a valid MLPPPoA packet.
4. The MLPPP functions on the ATM endpoint perform further processing.

Look at the packet encapsulation process when a packet goes in the ATM to the Frame Relay direction with transparent mode:

1. The ATM endpoint encapsulates the packet in an MLPPPoA header. It then segments the packets into cells and forwards them out the ATM segment.
2. The IWF receives the packet, forwards it to its Frame Relay interface, and prepends a two-byte Frame Relay header.
3. The Frame Relay endpoint examines the header of the received packet. If the first four bytes

after the two-byte Frame Relay header are 0xfefe03cf, the IWF treats the packet as a legal MLPPPoFR packet.

4. The MLPPP functions on the Frame Relay endpoint perform further processing.

The following illustrations show the format of MLPPPoA and MLPPPoFR packets.
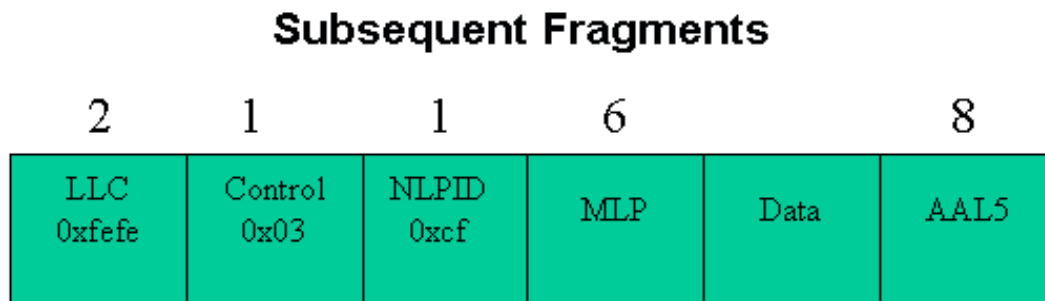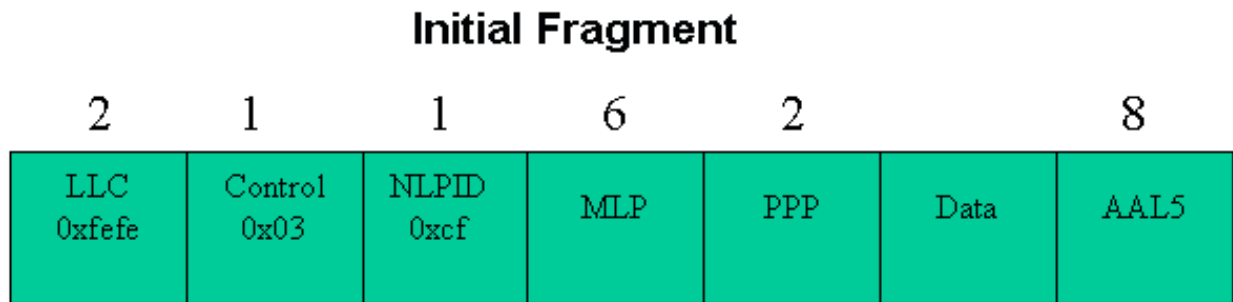
Initial Fragment

2	1	1	6	2		8
LLC 0xfefe	Control 0x03	NLPID 0xcf	MLP	PPP	Data	AAL5

Subsequent Fragments

2	1	1	6		8
LLC 0xfefe	Control 0x03	NLPID 0xcf	MLP	Data	AAL5

Figure 6. *MLPPPoA overhead. Only the first fragment carries a PPP header.*

Initial Fragment

1	2	1	1	6	2		2
Flag 0x7e	Frame Relay Header	Control 0x03	NLPID 0xcf	MLP	PPP	Data	FCS

Subsequent Fragments

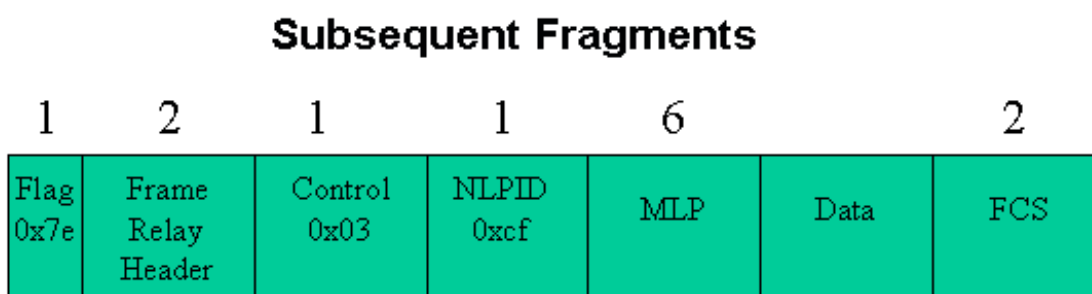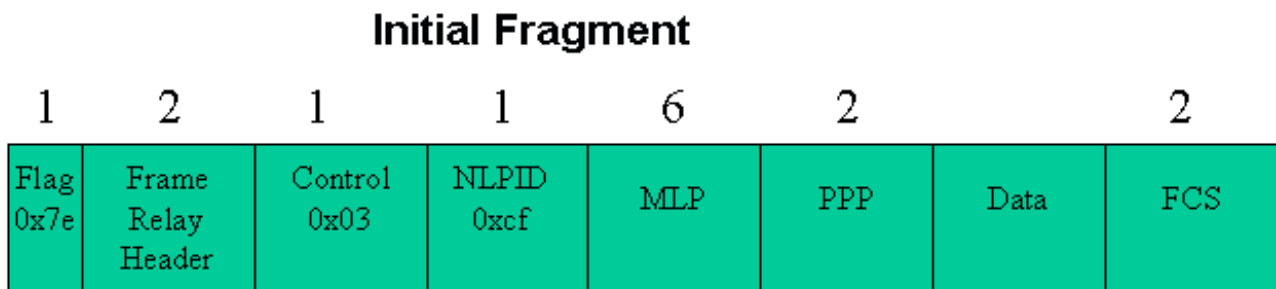| 1	2	1	1	6		2
Flag 0x7e	Frame Relay Header	Control 0x03	NLPID 0xcf	MLP	Data	FCS

Figure 7. *MLPPPoFR overhead. Only the first fragment carries a PPP header.*

VoIP Bandwidth Requirements

When provisioning bandwidth for VoIP, the data link overhead has to be included in the bandwidth calculations. Table 4 shows the per-call bandwidth requirements for VoIP depending on the codec and the use of compressed Real-time Transport Protocol (RTP). The calculations in Table 4 assume a best-case scenario for RTP header compression (cRTP), in other words, no UDP checksum or transmission errors. Headers are then consistently compressed from 40 bytes to two bytes.

Table 4 - *Per VoIP call bandwidth requirements (kbps).*

FRF.8 Mode	Transparent				Translation				Frame Relay to Frame Relay
Traffic Direction	Frame Relay to ATM		ATM to Frame Relay		Frame Relay to ATM		ATM to Frame Relay		
Frame Relay or ATM leg of PVC	Frame Relay	ATM	ATM	Frame Relay	Frame Relay	ATM	ATM	Frame Relay	
G729 - 20 ms Samples - No cRTP	27.6	42.4	42.4	28.4	27.6	42.4	42.4	27.6	26.8
G729 - 20 ms Samples - cRTP	12.4	21.2	21.2	13.2	12.4	21.2	21.2	12.4	11.6
G729 - 30 ms Samples - No cRTP	20.9	28.0	28.0	21.4	20.9	28.0	28.0	20.9	20.3
G729 - 30 ms Samples - cRTP	10.8	14.0	14.0	11.4	10.8	14.0	14.0	10.8	10.3
G711 - 20 ms Samples - No cRTP	83.6	106.0	106.0	84.4	83.6	106.0	106.0	83.6	82.8
G711 - 20 ms Samples - cRTP	68.4	84.8	84.8	69.2	68.4	84.8	84.8	68.4	67.6

G711 - 30 ms Samples - No cRTP	76.3	97.9	97.9	76.8	76.3	97.9	97.9	76.3	75.8
G711 - 30ms Samples - cRTP	66.3	84.0	84.0	66.8	66.3	84.0	84.0	66.3	65.7

Since overhead varies on each leg of the PVC, we recommend designing for a worst-case scenario. For example, consider the case of a G.279 call with 20 msec sampling and cRTP across a transparent PVC. On the Frame Relay leg, the bandwidth requirement is 12.4 kbps in one direction and 13.2 kbps in the other. Thus, we recommend provisioning based on 3.2 kbps per call.

For comparison purposes, the table also shows the VoIP bandwidth requirement on an end-to-end Frame Relay PVC configured with FRF.12 fragmentation. As noted in the table, PPP consumes between 0.5 kbps and 0.8 kbps of additional bandwidth per call to support the additional encapsulation header bytes. Thus, we recommend using FRF.12 with end-to-end Frame Relay VCs.

Compressed RTP (cRTP) over ATM requires Cisco IOS® Software Release 12.2(2)T. When cRTP is enabled with MLPoFR and MLPoATM, TCP/IP header compression is automatically enabled and cannot be disabled. This restriction results from RFC 2509, which does not allow PPP negotiation of RTP header compression without also negotiating TCP header compression.

Translation and Transparent Support on Cisco Devices

Originally, LFI required that IWF devices use transparent mode. More recently, the Frame Relay Forum introduced FRF.8.1 to support translation mode. Cisco introduced support for FRF.8.1 and translation mode in the following versions of Cisco IOS Software:

- 12.0(18)W5(23) for the LS1010 and Catalyst 8500 Series with a 4CE1 FR-PAM (CSCdt39211)
- 12.2(3)T and 12.2(2) on Cisco IOS routers with ATM interfaces, such as the PA-A3 (CSCdt70724)

Some service providers do not yet support PPP translation on their FRF.8 devices. Whenever this is the case, the provider must configure their PVCs for transparent mode.

Hardware and Software

This configuration uses the following hardware and software:

- ATM endpoint - PA-A3-OC3 in a 7200 Series router running Cisco IOS Software Release 12.2(8)T. (Note: LFI is supported on the PA-A3-OC3 and PA-A3-T3 only. It is not supported on the IMA and ATM OC-12 port adapters.)
- IWF device - LS1010 with Channelized T3 port adapter module and Cisco IOS Software Release 12.1(8)EY.
- Frame Relay endpoint - PA-MC-T3 in a 7200 Series router running Cisco IOS Software

Release 12.2(8)T.

Topology Diagram
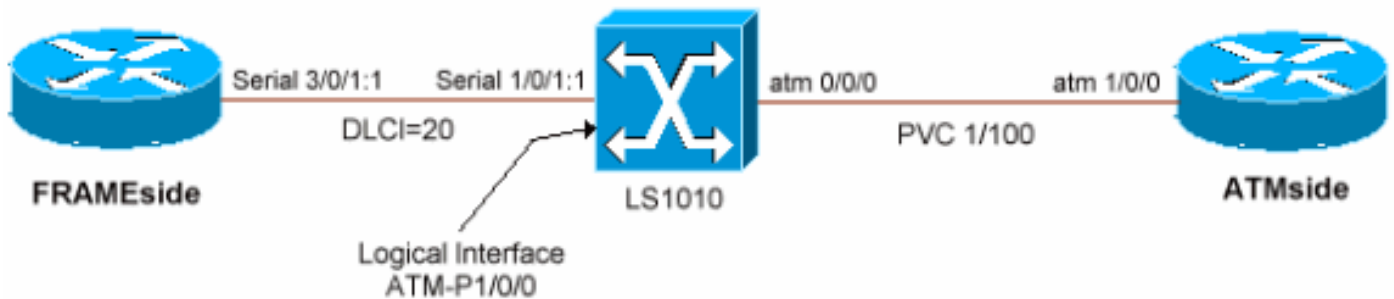
Configurations

This section shows how to configure the LFI feature over an FRF.8 link in transparent mode. It uses a virtual template on the two router endpoints, from which the MLP bundle's virtual access interface is cloned. LFI supports dialer interfaces and virtual templates for specifying the protocol-layer parameters of MLPPP. Cisco IOS Software Release 12.2(8)T increases to 200 the number of unique virtual templates that can be configured per router. Previous versions support only up to 25 virtual templates per router. This limitation can be a scaling issue on an ATM distribution router if every PVC is required to have a unique IP address. As a workaround, use IP as unnumbered or replace virtual templates with dialer interfaces on numbered links.

Cisco IOS Release 12.1(5)T introduced support for LFI over only one member link per MLPPP bundle. Thus, this configuration uses only a single VC at each endpoint. Support for multiple VCs per bundle is planned for an upcoming release of Cisco IOS.

Frame Relay Endpoint
1. The channelized T3 port adapter requires that you create a channel-group and specify the timeslots. By default, no interfaces exist. ``` FRAMEside#show ip int brief Interface   IP-Address   OK? Method Status Protocol FastEthernet0/0 172.16.142.231 YES NVRAM up up Loopback1   191.1.1.1   YES NVRAM up up ```

2. Use the **show diag** command to determine the installed port adapter. In this example, the T3 PA is in slot 3. Current versions of Cisco IOS now display the field replaceable (FRU) part number to order in case of a hardware failure.
 	
``
FRAMEside#show diag 3
Slot 3:
   CT3 single wide Port adapter, 1 port
   Port adapter is analyzed
``

```
     Port adapter insertion time 13:16:35 ago
     EEPROM contents at hardware discovery:
     Hardware revision 1.0          Board revision
  A0
     Serial number    23414844      Part number
  73-3037-01
     FRU Part Number:  PA-MC-T3= (SW)

     Test history     0x0           RMA number
  00-00-00
     EEPROM format version 1
     EEPROM contents (hex):
       0x20: 01 A0 01 00 01 65 48 3C 49 0B DD 01 00
  00 00 00
       0x30: 50 00 00 00 00 10 30 00 FF FF FF FF FF
  FF FF FF
```

3. Executing the **show controller t3** command
   displays physical-layer alarms and statistics.

```
  FRAMEside#show controller t3 3/0
  T3 3/0 is up.  Hardware is CT3 single wide port
  adapter
    CT3 H/W Version : 1.0.1, CT3 ROM Version : 1.1,
  CT3 F/W Version : 2.4.0
    FREEDM version: 1, reset 0 resurrect 0
    Applique type is Channelized T3
    No alarms detected.
    FEAC code received: No code is being received
    Framing is M23, Line Code is B3ZS, Clock Source
  is Internal
    Rx throttle total 0, equipment customer loopback
    Data in current interval (75 seconds elapsed):
       2 Line Code Violations, 1 P-bit Coding
  Violation
       0 C-bit Coding Violation, 1 P-bit Err Secs
       0 P-bit Severely Err Secs, 0 Severely Err
  Framing Secs
       0 Unavailable Secs, 1 Line Errored Secs
       0 C-bit Errored Secs, 0 C-bit Severely Errored
  Secs
    [output omitted]
```

4. Select a T1 from within T3 controller-configuration
   mode, create a channel-group, and assign timeslots
   to the group.

```
  FRAMEside(config)#controller t3 3/0
  b13-8-7204(config-controller)#?
  Controller configuration commands:
    cablelength  cable length in feet (0-450)
    clock        Specify the clock source for a T3
  link
    default      Set a command to its defaults
    description  Controller specific description
    equipment    Specify the equipment type for
  loopback mode
    exit         Exit from controller configuration
  mode
    framing      Specify the type of Framing on a T3
  link
```

```
   help        Description of the interactive help
system
   idle        Specify the idle pattern for all
channels on a T3 interface
   loopback    Put the entire T3 line into loopback
   mdl         Maintenance Data Link Configuration
   no          Negate a command or set its defaults
   shutdown    Shut down a DS3 link (send DS3 Idle)
   t1          Create a T1 channel

b13-8-7204(config-controller)#t1 ?
   <1-28>  T1 Channel number <1-28>

b13-8-7204(config-controller)#t1 1 channel-group ?
   <0-23>  Channel group number

b13-8-7204(config-controller)#t1 1 channel-group 1
?
   timeslots  List of timeslots in the channel group

b13-8-7204(config-controller)#t1 1 channel-group 1
timeslots ?
   <1-24>  List of timeslots which comprise the
channel

b13-8-7204(config-controller)#t1 1 channel-group 1
timeslots 1-2
b13-8-7204(config-controller)#
13:22:28: %LINK-3-UPDOWN: Interface Serial3/0/1:1,
changed state to down
13:22:29: %LINEPROTO-5-UPDOWN: Line protocol on
Interface Serial3/0/1:1, changed state to down
13:22:46: %LINK-3-UPDOWN: Interface Serial3/0/1:1,
changed state to up
13:22:47: %LINEPROTO-5-UPDOWN: Line protocol on
Interface Serial3/0/1:1, changed state to up
13:23:07: %LINEPROTO-5-UPDOWN: Line protocol on
Interface Serial3/0/1:1, changed state to down
```

Note: If the attached remote interface is not
similarly configured, the link layer of the new
channelized interface comes up, but the line
protocol stays down.

5. Interface serial 3/0/1:1 identifies the new
   channelized interface. Configure the interface for
   Frame Relay encapsulation and then enable Frame
   Relay Traffic Shaping (FRTS) on the main interface.

```
FRAMEside(config)#int serial 3/0/1:1
FRAMEside(config-if)#encapsulation frame-relay ietf
FRAMEside(config-if)#frame-relay traffic-shaping
!--- FRTS must be enabled for MLPoFR.
```

6. Configure a Frame Relay map-class to apply traffic-
   shaping parameters to the Frame Relay VC (which
   will be created below).

```
FRAMEside(config)#map-class frame-relay mlp
FRAMEside(config-map-class)#frame-relay cir ?
   <1-45000000>  Applied to both Incoming/Outgoing
CIR, Bits per second
```

```
    in            Incoming CIR
    out           Outgoing CIR


  FRAMEside(config-map-class)#frame-relay cir 128000
  FRAMEside(config-map-class)#frame-relay mincir
  128000
  FRAMEside(config-map-class)#frame-relay bc ?
    <300-16000000>  Applied to both Incoming/Outgoing
  Bc, Bits
    in            Incoming Bc
    out            Outgoing Bc
    <cr>
  FRAMEside(config-map-class)#frame-relay bc 1280
```
!--- Configure a burst committed (Bc) value of
1/100th of the CIR or 1280 bps. `FRAMEside(config-`
```
  map-class)#frame-relay be 0
```
!--- Configure an excess burst (Be) value of 0.
```
  FRAMEside(config-map-class)#no frame-relay
  adaptive-shaping
```

7. Create a QoS service policy. Use the same parameters as the ATM side. See below for reference.
```
  FRAMEside#show policy-map example
    Policy Map example
      Class voice
        Weighted Fair Queueing
            Strict Priority
            Bandwidth 110 (kbps) Burst 2750 (Bytes)
      Class class-default
        Weighted Fair Queueing
            Flow based Fair Queueing
            Bandwidth 0 (kbps) Max Threshold 64
  (packets)
```

8. Create a virtual template interface and apply MLPPP parameters. Also apply the QoS service-policy to the VC.
```
  FRAMEside(config)#interface Virtual-Template1
  FRAMEside(config-if)#ip address 1.1.1.2
  255.255.255.0
  FRAMEside(config-if)#service-policy output example
  FRAMEside(config-if)#ppp multilink
  FRAMEside(config-if)#ppp multilink fragment-delay
  10
  FRAMEside(config-if)#ppp multilink interleave
  FRAMEside(config-if)#end
```


9. Create a subinterface and assign the Frame Relay Data Link Connection Identifier (DLCI) number. Then apply PPP encapsulation, the virtual template, and the map-class.
```
  FRAMEside(config)#int serial 3/0/1:1.1 point
  FRAMEside(config-subif)#frame-relay interface-dlci
  ?
    <16-1007>  Define a switched or locally
  terminated DLCI

  FRAMEside(config-subif)#frame-relay interface-dlci
```

```
   20 ppp ?
    Virtual-Template  Virtual Template interface

 FRAMEside(config-subif)#frame-relay interface-dlci
 20 ppp Virtual-Template 1
 FRAMEside(config-fr-dlci)#class mlp
```

10. Use the **show frame-relay pvc** command to confirm your virtual-template and map-class parameters on the VC.

```
 FRAMEside#show frame-relay pvc 20

 PVC Statistics for interface Serial3/0/1:1 (Frame
 Relay DTE)

 DLCI = 20, DLCI USAGE = LOCAL, PVC STATUS =
 INACTIVE, INTERFACE = Serial3/0/1:1.1

   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
   5 minute input rate 0 bits/sec, 0 packets/sec
   5 minute output rate 0 bits/sec, 0 packets/sec
   pvc create time 00:03:24, last time pvc status
 changed 00:03:24
 Bound to Virtual-Access1 (down, cloned from
 Virtual-Template1)
   cir 128000    bc 1280      be 0          byte
 limit 160    interval 10
   mincir 128000    byte increment 160   Adaptive
 Shaping none
   pkts 0       bytes 0       pkts delayed 0
 bytes delayed 0
   shaping inactive
   traffic shaping drops 0
   Queueing strategy: fifo
   Output queue 0/40, 0 drop, 0 dequeued
```

11. Use the **show controller serial 3/0/1:1** to confirm that the Frame Relay link is in an up status and clear of physical-layer alarms. Each channelized interface is assigned a "VC" number. In the following output, channel-group 1 (3/0/1:1) is assigned a VC number of 0.

```
 FRAMEside#show controller serial 3/0/1:1
 CT3 SW Controller 3/0
   ROM ver 0x10001, h/w ver 1.0.1, f/w ver 2.4.0,
 FREEDM rev 1
 !--- FREEDM is the HDLC controller on the
 channelized T3 port adapter. It extracts data from
 the 24 timeslots of a T1, validates the CRC, and
 checks for any other frame errors. T3 linestate is
 Up, T1 linestate 0x00000002, num_active_idb 1
```

```
   Buffer pool size 640, particle size 512, cache
size 640, cache end 128/127
   Rx desctable 0xF1A5A20, shadow 0x628C6AFC, size
512, spin 128
```
*!--- When it initializes, the interface driver
builds a control structure known as the receive
ring. The receive ring consists of a list of 512
packet buffer descriptors. As packets arrive,
FREEDM DMAs the data into the buffer to which a
descriptor points.* rx queue 0xF1B8000, cache
0xF1B8000, fq base 0xF1B8800 rdq base 0xF1B8000,
host_rxrdqr 0xF1B8004, host_rxfqw 0xF1B8804 Tx
desctable 0xF1A7A60, shadow 0x628B6AD0, size 4096,
spin 256 *!--- When it initializes, the interface
driver also creates the transmit queue or transmit
ring. In the case of the channelized T3 PA, the
driver creates a queue of 4096 entries and sets
all fields in the descriptors to NULL or empty.* tx
queue 0xF1C0000, cache 0xF1C0000 host_txrdqw 1802,
fq base 0xF1C4000, host_txfqr 0xF1C5C20 dynamic
txlimit threshold 4096 TPD cache 0x628C7A54, size
4096, cache end 4096/4094, underrun 0 RPD cache
0x628C7328, size 448, cache end 0 Freedm fifo
0x628AA7B0, head ptr 0x628AA7C8, tail ptr
0x628AB7A8, reset 0 PCI bus 6, PCI shared memory
block 0xF1A454C, PLX mailbox addr 0x3D820040
FREEDM devbase 0x3D800000, PLX devbase 0x3D820000
Rx overruns 0, Tx underruns 0, **tx rdq count 0**
*!--- The "tx rdq count" indicates the number of
outstanding transmit packets in FREEDM's "transmit
ready" queue. This queue holds a packet before it
reaches the transmit ring.* Tx bad vc 0 FREEDM err:
cas 0, hdl 0, hdl_blk 0, ind_prov 0, tavail 0,
tmac busy 0, rmac b usy 0 rxrdq_wt 0x2, rxrdq_rd
0x1, rxsfq_wt 0x201, rxsfq_rd 0x206 **VC 0 (1:1) is
enabled, T1 1 is enabled/Up**, rx throttle 0
Interface Serial3/0/1:1 is up (idb status
0x84208080)
```
   xmitdelay 0, max pak size 1608, maxmtu 1500, max
buf size 1524
   started 8, throttled 0, unthrottled 0,
in_throttle FALSE
   VC config: map 0xC0000000, timeslots 2, subrate
0xFF, crc size 2, non-inverted data
     freedm fifo num 3, start 0x628AA7B0, end
0x628AA7C0, configured = TRUE
   Rx pkts 0, bytes 0, runt 0, giant 0, drops 0
     crc 0, frame 0, overrun 0, abort 1, no buf 0
   Tx pkts 194313, bytes 2549490, underrun 0, drops
0, tpd udr 0
     tx enqueued 0, tx count 0/36/0, no buf 0
     tx limited = FALSE
```
*!--- The "tx count x/y/z" counter includes the
following information: !--- "x" = Number of
transmit ring entries in use. !--- "y" = Maximum
number of packets allowed on the transmit queue.
!--- "z" = Number of times that the transmit limit
has been exceeded.*

LS1010 Configuration

1. Use the **show hardware** command to confirm that
   your LS1010 is equipped with a channelized Frame

Relay port adapter module (PAM).

```
LS1010#show hardware
LS1010 named LS1010, Date: 07:36:40 UTC Mon May 13
2002
Feature Card's FPGA Download Version: 11
Slot Ctrlr-Type     Part No.   Rev  Ser No  Mfg Date
RMA No.   Hw Vrs  Tst EEP
---- ------------   ---------- -- -------- ---------
-------- ------- --- ---
0/0  155MM PAM     73-1496-03 A0 02829507 May 07 96
00-00-00   3.1     0   2
1/0  1CT3 FR-PAM   73-2972-03 A0 12344261 May 17 99
00-00-00   3.0     0   2
2/0  ATM Swi/Proc 73-1402-03 B0 03824638 Sep 14 96
00-00-00   3.1     0   2
2/1  FeatureCard1 73-1405-03 B0 03824581 Sep 14 96
00-00-00   3.2     0   2
```

2. Use the **show ip int brief** command to identify the controller interface.

```
LS1010#show ip int brief
Interface       IP-Address       OK? Method Status
Protocol
ATM0/0/0        unassigned       YES unset  up
up
ATM0/0/1        unassigned       YES unset  down
down
ATM0/0/2        unassigned       YES unset  down
down
ATM0/0/3        unassigned       YES unset  down
down
ATM-P1/0/0      unassigned       YES unset  up
up
T3 1/0/0        unassigned       YES unset  up
up
```

3. Create a channelized interface and select the same timeslots as the serial port adapter (PA).

```
LS1010(config)#controller t3 1/0/0
LS1010(config-controller)#channel-group 1 t1 ?
  <1-28>  T1 line number <1-28>

LS1010(config-controller)#channel-group 1 t1 1
timeslots ?
  <1-24>  List of timeslots which comprise the
channel

LS1010(config-controller)#channel-group 1 t1 1
timeslot 1-2
LS1010(config-controller)#

2w1d: %LINK-3-UPDOWN: Interface Serial1/0/0:1,
changed state to up
2w1d: %LINEPROTO-5-UPDOWN: Line protocol on
Interface Serial1/0/0:1, changed state to up
```

4. Configure Frame Relay encapsulation on the new serial interface. In addition, change the Local Management Interface (LMI) type from NNI to DCE.

```
LS1010(config)#int serial 1/0/0:1
LS1010(config-if)#encap frame ?
  ietf  Use RFC1490 encapsulation

LS1010(config-if)#encap frame ietf
LS1010(config-if)#frame-relay intf-type dce
```

5. Use the **show interface serial** command to confirm Frame Relay encapsulation.

```
LS1010#show int serial 1/0/0:1
Serial1/0/0:1 is up, line protocol is up
  Hardware is FRPAM-SERIAL
  MTU 4096 bytes, BW 128 Kbit, DLY 0 usec,
    reliability 139/255, txload 1/255, rxload
1/255
  Encapsulation FRAME-RELAY IETF, loopback not set
  Keepalive set (10 sec)
  LMI enq sent  32, LMI stat recvd 0, LMI upd recvd
0
  LMI enq recvd 40, LMI stat sent  40, LMI upd sent
0, DCE LMI up
  LMI DLCI 1023  LMI type is CISCO  frame relay DCE
!--- By default, the serial PAM and the serial PA
use LMI type Cisco. The serial PAM should show DCE
LMI status of "up", and the serial PA should show
DTE LMI status of "up". Broadcast queue 0/64,
broadcasts sent/dropped 0/0, interface broadcasts 0
Last input 00:00:03, output 00:00:05, output hang
never Last clearing of "show interface" counters
00:06:40 Input queue: 0/75/0/0
(size/max/drops/flushes); Total output drops: 0
Queueing strategy: fifo Output queue :0/40
(size/max) 5 minute input rate 0 bits/sec, 0
packets/sec 5 minute output rate 0 bits/sec, 0
packets/sec 44 packets input, 667 bytes, 0 no
buffer Received 0 broadcasts, 0 runts, 0 giants, 0
throttles 5 input errors, 0 CRC, 0 frame, 0
overrun, 0 ignored, 0 abort 71 packets output, 923
bytes, 0 underruns 0 output errors, 0 collisions, 0
interface resets 0 output buffer failures, 0 output
buffers swapped out 0 carrier transitions
Timeslots(s) Used: 1-2 on T1 1
    Frames Received with:
     DE set: 0, FECN set :0, BECN set: 0
    Frames Tagged :
     DE: 0, FECN: 0 BECN: 0
    Frames Discarded Due to Alignment Error: 0
    Frames Discarded Due to Illegal Length: 0
    Frames Received with unknown DLCI: 5
    Frames with illegal Header : 0
    Transmit Frames with FECN set :0,  BECN Set :0
    Transmit Frames Tagged FECN : 0 BECN : 0
    Transmit Frames Discarded due to No buffers : 0
    Default Upc Action : tag-drop
    Default Bc (in Bits) : 32768


LS1010#show frame lmi

LMI Statistics for interface Serial1/0/0:1 (Frame
Relay DCE) LMI TYPE = CISCO<
  Invalid Unnumbered info 0              Invalid
Prot Disc 0
```

```
   Invalid dummy Call Ref 0              Invalid Msg
Type 0
   Invalid Status Message 0             Invalid
Lock Shift 0
   Invalid Information ID 0             Invalid
Report IE Len 0
   Invalid Report Request 0            Invalid
Keep IE Len 0
   Num Status Enq. Rcvd 120            Num Status
msgs Sent 120
   Num Update Status Sent 0            Num St Enq.
Timeouts 0
```

6. Before you configure the PVC, ensure that the ATM
   interface is up/up.
   ```
   LS1010#show int atm 0/0/0
   ATM0/0/0 is up, line protocol is up
     Hardware is oc3suni
     MTU 4470 bytes, sub MTU 4470, BW 155520 Kbit, DLY
   0 usec,
        reliability 255/255, txload 1/255, rxload
   1/255
     Encapsulation ATM, loopback not set
     Last input 00:00:00, output 00:00:00, output hang
   never
     Last clearing of "show interface" counters never
     Input queue: 0/75/0/0 (size/max/drops/flushes);
   Total output drops: 0
     Queueing strategy: fifo
     Output queue :0/40 (size/max)
     5 minute input rate 0 bits/sec, 0 packets/sec
     5 minute output rate 1000 bits/sec, 2 packets/sec
        253672 packets input, 13444616 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
        2601118 packets output, 137859254 bytes, 0
   underruns
        0 output errors, 0 collisions, 0 interface
   resets
        0 output buffer failures, 0 output buffers
   swapped out
   ```
7. In addition to the two physical interfaces, the
   LS1010 uses a logical interface to link the ATM side
   and the Frame Relay side. The logical interface is
   identified as "atm-p1" on the ATM pseudo interface.
   ```
   LS1010#show int atm-p1/0/0
   ATM-P1/0/0 is up, line protocol is up
     Hardware is ATM-PSEUDO
     MTU 4470 bytes, sub MTU 4470, BW 45000 Kbit, DLY
   0 usec,
        reliability 0/255, txload 1/255, rxload 1/255
     Encapsulation ATM, loopback not set
     Keepalive not supported
     Encapsulation(s):
     2000 maximum active VCs, 0 current VCCs
     VC idle disconnect time: 300 seconds
     Last input never, output never, output hang never
     Last clearing of "show interface" counters never
   ```

```
     Input queue: 0/75/0/0 (size/max/drops/flushes);
   Total output drops: 0
     Queueing strategy: fifo
     Output queue :0/40 (size/max)
     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
   throttles
        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, 0 interface
   resets
        0 output buffer failures, 0 output buffers
   swapped out
```

8. In serial interface configuration mode, configure the interworking PVC.

   ```
   interface Serial1/0/0:1
    no ip address
    encapsulation frame-relay IETF
    no arp frame-relay
    frame-relay intf-type dce
    frame-relay pvc 20 service transparent  interface
   ATM0/0/0 1 100
   ```

9. Confirm your configuration with the show vc interface atm command.

   ```
   LS1010#show vc int atm 0/0/0
   Interface       Conn-Id  Type   X-Interface     X-
   Conn-Id  Encap  Status
   ATM0/0/0        0/5      PVC    ATM0            0/39
   QSAAL    UP
   ATM0/0/0        0/16     PVC    ATM0            0/35
   ILMI     UP
   ATM0/0/0        1/100    PVC    Serial1/0/0:1   20
   UP
   ```

ATM Endpoint

1. Ensure that you are using an enhanced ATM PA or PA-A3. Use the **show interface atm** command to confirm.

   ```
   ATMside#show int atm 1/0/0
   ATM1/0/0 is up, line protocol is up
     Hardware is cyBus ENHANCED ATM PA
     MTU 4470 bytes, sub MTU 4470, BW 149760 Kbit, DLY
   80 usec,
        reliability 255/255, txload 1/255, rxload
   1/255
     Encapsulation ATM, loopback not set
     Encapsulation(s): AAL5
     4095 maximum active VCs, 0 current VCCs
   [output omitted]
   ```

2. Configure the ATM-layer parameters of the permanent virtual circuit (PVC). In this configuration, we are using a point-to-point subinterface with a

sustained cell rate (SCR) of 150 kbps. This value was selected to be about 15% higher than the Frame Relay endpoint's CIR of 128 kbps. The additional 15% helps to ensure that the VC delivers an equivalent bandwidth to actual user traffic on both sides of the connection while accommodating the extra overhead of the ATM side. (See also [Configuring Traffic Shaping on Frame Relay to ATM Service Interworking (FRF.8) PVCs](#).)

```
ATMside(config)#int atm 1/0/0.1 point
ATMside(config-subif)#pvc 1/100
ATMside(config-if-atm-vc)#vbr-nrt 300 150 ?
  <1-65535>  Maximum Burst Size(MBS) in Cells
  <cr>

ATMside(config-if-atm-vc)#vbr-nrt 300 150
ATMside(config-if-atm-vc)#end
ATMside(config-if-atm-vc)#tx-ring-limit 4
!--- Tune down the transmit ring to push most
queueing to the layer-3 queues, where our service
policy will apply.
```

3. Confirm that your VC appears in the VC table. Execute the **show atm vc** command. Note that the router assigns a default maximum burst size (MBS) of 94 since we did not enter an explicit value.

```
ATMside#show atm vc
            VCD /                                Peak
Avg/Min Burst
Interface Name VPI  VCI Type Encaps SC  kbps kbps
Cells Sts
1/0/0.1    1    1    100 PVC  SNAP   VBR 300  150
94    UP
```

4. Create a QoS service policy. In the policy shown below, we created four classes, including the router-created class-default class. Create a class-map for the voice over IP (VoIP) packets.

```
ATMside(config)#class-map voice
ATMside(config-cmap)#match ip rtp ?
  <2000-65535>  Lower bound of UDP destination port

ATMside(config-cmap)#match ip rtp 16384 ?
  <0-16383>  Range of UDP ports

ATMside(config-cmap)#match ip rtp 16384 16383

!--- Cisco IOS H.323 devices use this UDP port
range to transmit VoIP packets.
```

Create a class-map for the voice signaling packets. This example uses H.323 Fast Connect. (See also the "LLQ Configuration Guidelines" section of [VoIP over PPP Links with Quality of Service (LLQ / IP RTP Priority, LFI, cRTP](#).)

```
class-map voice-signaling
  match access-group 103
!
```

```
access-list 103 permit tcp any eq 1720 any
access-list 103 permit tcp any any eq 1720
```

Create a named policy-map and assign QoS actions to each class. This example assigns priority queueing to the VoIP user packets with the **priority** command and a minimum bandwidth guarantee to call-signaling packets with the **bandwidth** command. All other traffic goes to the class-default class, which separates the traffic into IP-layer flows and provides fair-queueing among the flows.

```
policy-map example
  class call-control
    bandwidth percent 10
  class voice
     priority 110
  class class-default
    fair-queue
```

Confirm your configuration.
```
ATMside#show policy-map example
  Policy Map example
    Class call-control
      bandwidth percent 10
    Class voice
      priority 110
    Class class-default
      fair-queue
```

5. Create a virtual template and apply the QoS service policy to it.

```
interface Virtual-Template1
  bandwidth 150
  ip address 1.1.1.1 255.255.255.0
  service-policy output example
  ppp multilink
  ppp multilink fragment-delay 10
  ppp multilink interleave
```

!--- You select a fragment size indirectly by specifying the maximum tolerable serialization delay. The recommended maximum per-hop serialization delay for voice environments is 10 milliseconds (ms). LFI also requires ppp multilink interleave.

6. Apply the virtual template and multilink-PPP encapsulation to the ATM PVC.
```
ATMside(config)#int atm 1/0/0.1
ATMside(config-subif)#pvc 1/100
ATMside(config-if-atm-vc)#protocol ppp ?
  Virtual-Template  Virtual Template interface
  dialer            pvc is part of dialer profile

ATMside(config-if-atm-vc)#protocol ppp Virtual-
Template 1
```

7. Confirm your settings on the ATM PVC.
   ```
   ATMside#show run int atm 1/0/0.1
   Building configuration...

   Current configuration : 127 bytes
   !
   interface ATM1/0/0.1 point-to-point
    pvc 1/100
      vbr-nrt 300 150
      tx-ring-limit 4
      protocol ppp Virtual-Template1
    !
   end
   ```

8. The router creates a virtual-access interface automatically. If you do not have MLPPP configured on the Frame Relay endpoint, the status of the virtual-access interface is up/down.
   ```
   ATMside#show int virtual-access 1
   Virtual-Access1 is up, line protocol is down
     Hardware is Virtual Access interface
     Internet address is 1.1.1.1/24
     MTU 1500 bytes, BW 150 Kbit, DLY 100000 usec,
        reliability 255/255, txload 1/255, rxload
   1/255
     Encapsulation PPP, loopback not set
     DTR is pulsed for 5 seconds on reset
     LCP Listen, multilink Closed
     Closed: LEXCP, BRIDGECP, IPCP, CCP, CDPCP, LLC2,
   BACP, IPV6CP
     Bound to ATM1/0/0.1 VCD: 1, VPI: 1, VCI: 100
     Cloned from virtual-template: 1
   ```

show and debug Commands

ATM Endpoint

Use the following commands on the ATM endpoint to confirm that LFI is working correctly. Before issuing debug commands, please see Important Information on Debug Commands.

- **show ppp multilink** - LFI uses two virtual-access interfaces -- one for PPP and one for the MLP bundle. Use the **show ppp multilink** to differentiate between the two.
  ```
  ATMside#show ppp multilink
  Virtual-Access2, bundle name is FRAMEside
  !--- The bundle interface is assigned to VA 2. Bundle up for 01:11:55 Bundle is Distributed
  0 lost fragments, 0 reordered, 0 unassigned 0 discarded, 0 lost received, 1/255 load 0x1E
  received sequence, 0xA sent sequence Member links: 1 (max not set, min not set) Virtual-
  Access1, since 01:11:55, last rcvd seq 00001D 187 weight !--- The PPP interface is assigned
  to VA 1.
  ```
- **show interface virtual-access 1** - Confirm that the virtual-access interface is up/up and incrementing the input and output packets counters.
  ```
  ATMside#show int virtual-access 1
  Virtual-Access1 is up, line protocol is up
    Hardware is Virtual Access interface
  ```

```
    Internet address is 1.1.1.1/24
    MTU 1500 bytes, BW 150 Kbit, DLY 100000 usec,
       reliability 255/255, txload 1/255, rxload 1/255
    Encapsulation PPP, loopback not set
    DTR is pulsed for 5 seconds on reset
    LCP Open, multilink Open
    Bound to ATM1/0/0.1 VCD: 1, VPI: 1, VCI: 100
    Cloned from virtual-template: 1
    Last input 01:11:30, output never, output hang never
    Last clearing of "show interface" counters 2w1d
    Input queue: 0/75/0/0 (size/max/drops/flushes); Total output drops: 0
    Queueing strategy: fifo
    Output queue :0/40 (size/max)
    5 minute input rate 0 bits/sec, 0 packets/sec
    5 minute output rate 0 bits/sec, 0 packets/sec
       878 packets input, 13094 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
       255073 packets output, 6624300 bytes, 0 underruns
       0 output errors, 0 collisions, 0 interface resets
       0 output buffer failures, 0 output buffers swapped out
       0 carrier transitions
```

- **show policy-map int virtual-access 2** - Confirm that the QoS service policy is bound to the MLPPP bundle interface.

```
ATMside#show policy-map int virtual-access 2
 Virtual-Access2

  Service-policy output: example

    queue stats for all priority classes:
      queue size 0, queue limit 27
      packets output 0, packet drops 0
      tail/random drops 0, no buffer drops 0, other drops 0

    Class-map: call-control (match-all)
      0 packets, 0 bytes
      5 minute offered rate 0 bps, drop rate 0 bps
      Match: access-group 103
      queue size 0, queue limit 3
      packets output 0, packet drops 0
      tail/random drops 0, no buffer drops 0, other drops 0
      Bandwidth: 10%, kbps 15

    Class-map: voice (match-all)
      0 packets, 0 bytes
      5 minute offered rate 0 bps, drop rate 0 bps
      Match: ip rtp 16384 16383
      Priority: kbps 110, burst bytes 4470, b/w exceed drops: 0

    Class-map: class-default (match-any)
      0 packets, 0 bytes
      5 minute offered rate 0 bps, drop rate 0 bps
      Match: any
      queue size 0, queue limit 5
      packets output 0, packet drops 0
      tail/random drops 0, no buffer drops 0, other drops 0
      Fair-queue: per-flow queue limit 2
```

- **debug ppp packet and debug atm packet** - Use these commands if all interfaces are up/up, but you are not able to ping end to end. In addition, you can use these commands to capture

PPP keepalives, as illustrated below.

```
2w1d: Vi1 LCP-FS: I ECHOREQ [Open] id 31 len 12 magic 0x52FE6F51
2w1d: ATM1/0/0.1(O):
VCD:0x1 VPI:0x1 VCI:0x64 DM:0x0 SAP:FEFE CTL:03  Length:0x16
2w1d: CFC0 210A 1F00 0CB1 2342 E300 0532 953F
2w1d:
2w1d: Vi1 LCP-FS: O ECHOREP [Open] id 31 len 12 magic 0xB12342E3
```
!--- This side received an Echo Request and responded with an outbound Echo Reply. 2w1d: Vi1 LCP: O ECHOREQ [Open] id 32 len 12 magic 0xB12342E3 2w1d: ATM1/0/0.1(O): VCD:0x1 VPI:0x1 VCI:0x64 DM:0x0 SAP:FEFE CTL:03 Length:0x16 2w1d: CFC0 2109 2000 0CB1 2342 E300 049A A915 2w1d: Vi1 LCP-FS: I ECHOREP [Open] id 32 len 12 magic 0x52FE6F51 2w1d: Vi1 LCP-FS: Received id 32, sent id 32, line up *!--- This side transmitted an Echo Request and received an inbound Echo Reply.*

Frame Relay Endpoint

Use the following commands on the Frame Relay endpoint to confirm that LFI is working correctly. Before issuing debug commands, please see [Important Information on Debug Commands](#).

- **show ppp multilink** - LFI uses two virtual-access interfaces -- one for PPP and one for the MLP bundle. Use the **show ppp multilink** to differentiate between the two.
  ```
  FRAMEside#show ppp multilink

  Virtual-Access2, bundle name is ATMside
    Bundle up for 01:15:16
    0 lost fragments, 0 reordered, 0 unassigned
    0 discarded, 0 lost received, 1/255 load
    0x19 received sequence, 0x4B sent sequence
    Member links: 1 (max not set, min not set)
      Virtual-Access1, since 01:15:16, last rcvd seq 000018  59464 weight
  ```

- **show policy-map interface virtual-access** - Confirm that the QoS service policy is bound to the MLPPP bundle interface.
  ```
  FRAMEside#show policy-map int virtual-access 2
   Virtual-Access2
    Service-policy output: example

      Class-map: voice (match-all)
        0 packets, 0 bytes
        5 minute offered rate 0 bps, drop rate 0 bps
        Match: ip rtp 16384 16383
        Weighted Fair Queueing
          Strict Priority
          Output Queue: Conversation 264
          Bandwidth 110 (kbps) Burst 2750 (Bytes)
          (pkts matched/bytes matched) 0/0
          (total drops/bytes drops) 0/0

      Class-map: class-default (match-any)
        27 packets, 2578 bytes
        5 minute offered rate 0 bps, drop rate 0 bps
        Match: any
        Weighted Fair Queueing
          Flow Based Fair Queueing
          Maximum Number of Hashed Queues 256
          (total queued/total drops/no-buffer drops) 0/0/0
  ```

- **debug frame packet** and **debug ppp packet** - Use these commands if all interfaces are

up/up, but you are not able to ping end-to-end.

```
FRAMEside#debug frame packet
Frame Relay packet debugging is on
FRAMEside#
FRAMEside#ping 1.1.1.1
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 1.1.1.1, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 36/36/40 ms
FRAMEside#
2w1d: Serial3/0/1:1.1(o): dlci 20(0x441), NLPID 0x3CF(MULTILINK), datagramsize 52
2w1d: Serial3/0/1:1.1(o): dlci 20(0x441), NLPID 0x3CF(MULTILINK), datagramsize 52
2w1d: Serial3/0/1:1.1(o): dlci 20(0x441), NLPID 0x3CF(MULTILINK), datagramsize 28
2w1d: Serial3/0/1:1.1(o): dlci 20(0x441), NLPID 0x3CF(MULTILINK), datagramsize 52
2w1d: Serial3/0/1:1.1(o): dlci 20(0x441), NLPID 0x3CF(MULTILINK), datagramsize 52
2w1d: Serial3/0/1:1.1(o): dlci 20(0x441), NLPID 0x3CF(MULTILINK), datagramsize 28
2w1d: Serial3/0/1:1.1(o): dlci 20(0x441), NLPID 0x3CF(MULTILINK), datagramsize 52
2w1d: Serial3/0/1:1.1(o): dlci 20(0x441), NLPID 0x3CF(MULTILINK), datagramsize 52
2w1d: Serial3/0/1:1.1(o): dlci 20(0x441), NLPID 0x3CF(MULTILINK), datagramsize 28
2w1d: Serial3/0/1:1.1(o): dlci 20(0x441), NLPID 0x3CF(MULTILINK), datagramsize 52
2w1d: Serial3/0/1:1.1(o): dlci 20(0x441), NLPID 0x3CF(MULTILINK), datagramsize 52
```

Queueing and LFI

MLPPPoA and MLPPPoFR clone two virtual-access interfaces from the dialer interface or virtual template. One such interface represents the PPP link, and the other represents the MLP bundle interface. Use the **show ppp multilink** command to determine the specific interface used for each function. As of this writing, only one VC per bundle is supported, and thus only one virtual-access interface should appear in the bundle-member list in the **show ppp multilink** output.

In addition to the two virtual-access interfaces, each PVC is associated with a main interface and a subinterface. Each of these interfaces provides some form of queueing. However, only the virtual-access interface representing the bundle interface supports fancy queueing via an applied QoS service policy. The other three interfaces must have FIFO queueing. When applying a service-policy to a virtual-template, the router displays the following message:

```
cr7200(config)#interface virtual-template 1
cr7200(config)#service-policy output Gromit
Class Base Weighted Fair Queueing not supported on interface Virtual-Access1
```

Note: Class Based Weighted Fair Queueing supported on MLPPP bundle interface only.

These messages are normal. The first message is advising that a service-policy is not supported on the PPP virtual-access interface. The second message confirms that the service-policy is applied to the MLP bundle virtual-access interface. To confirm the queueing mechanism on the MLP bundle interface, use the commands **show interface virtual-access**, **show queue virtual-access**, and **show policy-map interface virtual-access**.

MLPPPoFR requires that Frame Relay Traffic Shaping (FRTS) be enabled on the physical interface. FRTS activates per-VC queues. On platforms such as the 7200, 3600, and 2600 Series, FRTS is configured with the following two commands:

- **frame-relay traffic-shaping** on the main interface

- **map-class** with any shaping commands.

Current versions of Cisco IOS prints the following warning message if MLPPoFR is applied without FRTS.

```
"MLPoFR not configured properly on Link x Bundle y"
```

If you see this warning message, ensure that FRTS has been configured on physical interface and that the QoS service policy has been attached to the virtual template. To verify the configuration, use the **show running-config serial interface** and **show running-config virtual-template** commands. When MLPPPoFR is configured, the interface queueing mechanism changes to dual FIFO, as illustrated below. The high-priority queue handles voice packets and control packets, such as Local Management Interface (LMI), and the low-priority queue handles fragmented packets, presumably data or non-voice packets.

```
Router#show int serial 6/0:0
    Serial6/0:0 is up, line protocol is down
      Hardware is Multichannel T1
      MTU 1500 bytes, BW 64 Kbit, DLY 20000 usec,
          reliability 255/255, txload 1/255, rxload 1/255
      Encapsulation FRAME-RELAY, crc 16, Data non-inverted
      Keepalive set (10 sec)
      LMI enq sent  236, LMI stat recvd 0, LMI upd recvd 0, DTE LMI down
      LMI enq recvd 353, LMI stat sent  0, LMI upd sent  0
      LMI DLCI 1023  LMI type is CISCO  frame relay DTE
      Broadcast queue 0/64, broadcasts sent/dropped 0/0, interface broadcasts 0
      Last input 00:00:02, output 00:00:02, output hang never
      Last clearing of "show interface" counters 00:39:22
      Queueing strategy: dual fifo
      Output queue: high size/max/dropped 0/256/0 !--- high-priority queue Output
queue 0/128, 0 drops; input queue 0/75, 0 drops !--- low-priority queue 5 minute
input rate 0 bits/sec, 0 packets/sec 5 minute output rate 0 bits/sec, 0 packets/sec
353 packets input, 4628 bytes, 0 no buffer Received 0 broadcasts, 0 runts, 0 giants,
0 throttles 0 input errors, 0 CRC, 0 frame, 0 overrun, 0 ignored, 0 abort 353 packets
output, 4628 bytes, 0 underruns 0 output errors, 0 collisions, 0 interface resets 0
output buffer failures, 0 output buffers swapped out 0 carrier transitions no alarm
present Timeslot(s) Used:12, subrate: 64Kb/s, transmit delay is 0 flags
```

LFI uses two layers of queueing -- MLPPP bundle level, which supports fancy queueing, and PVC level, which only supports FIFO queueing. The bundle interface maintains its own queue. All MLP packets go through the MLP bundle and virtual access layers first before the Frame Relay or ATM layer. LFI monitors the size of the member links' hardware queues and dequeues packets to the hardware queues when they fall below a threshold, which originally was a value of two. Otherwise, the packets are queued in the MLP bundle queue.

Troubleshooting and Known Issues

The following table lists known issues with LFI over FRF links and focuses on the troubleshooting steps to take to isolate your symptoms to a resolved bug.

Symptom	Troubleshooting Steps	Resolved Bugs
Reduced	Ping with various-sized	CSCdt59038 - With 1500-byte packets and

throughput on ATM leg or Frame Relay leg	packets from 100 bytes to the Ethernet MTU. • Do large packets experience timeouts?	fragmentation set to 100 bytes, there are 15 fragmented packets. The delay was caused by multiple levels of queueing. CSCdu18344 - With FRTS, packets are dequeued slower than expected. The MLPPP bundle dequeue function checks the queue size of the traffic shaper queue. FRTS was too slow in clearing this queue.

Out-of-order packets	• Execute the **show ppp multilink** command. Look for incrementing values for "lost fragments", "discarded", and "lost received" counters.
	`Virtual-Access4,`
	`bundle name is`
	`xyz`
	`Bundle up for`
	`03:56:11`
	`2524 lost`
	`fragments, 3786`
	`reordered,`
	`0 unassigned`
	`1262 discarded,`
	`1262 lost`
	`received,`
	`1/255 load`
	`0x42EA1 received`
	`sequence, 0xCF7`
	`sent sequence`
	`Member links: 1`
	`(max not set,`
	`min`
	`not set)`
	`Virtual-Access1,`
	`since`
	`03:59:02, last`
	`rcvd seq 042EA0`
	`400`
	`weight`
	
	• Enable **debug ppp multi events** and look for "Lost fragment" and
		CSCdv89201 - When the physical ATM interface is congested, MLP fragments are dropped or received out of order at the remote end. This problem affects only ATM network modules on the 2600 and 3600 Series. It results from how the interface driver was incorrectly switching packets in the fast path (such as with fast switching or Cisco Express Forwarding). Specifically, the second fragment of the current packet was sent after the first fragment of the next packet

	"Out of sync with peer" messages. ``` *Mar 17 09:14:08.216: Vi4 MLP: Lost fragment 3FED9 in 'dhartr21' (all links have rcvd higher seq#) *Mar 17 09:14:08.232: Vi4 MLP: Received lost fragment seq 3FED9, expecting 3FEDC in 'dhartr21' *Mar 17 09:14:08.232: Vi4 MLP: Out of sync with peer, resyncing to last rcvd seq# (03FED9) *Mar 17 09:14:08.236: Vi4 MLP: Unusual jump in seq number, from 03FEDC to 03FEDA ```	
Loss of end-to-end connectivity when 3600 Series performs IWF in transparent mode	• Change the mode to translational and test again.	CSCdw11409 - Ensures that CEF looks in the correct byte location to begin processing the encapsulation headers of MLPPP packets

Related Information

- **Configuring Link Fragmentation and Interleaving for Frame Relay and ATM Virtual Circuits**
- **Designing and Deploying Multilink PPP over Frame Relay and ATM**

- [RFC2364, PPP Over AAL5, July 1998](#)
- [RFC1973, PPP in Frame Relay, June 1996](#)
- [RFC1717, The PPP Multilink Protocol (MP), Nov. 1994](#)
- [Frame Relay / ATM PVC Service Interworking Implementation Agreement FRF.8](#)
- [More ATM Information](#)
- [Tools & Resources - Cisco Systems](#)
- [Technical Support & Documentation - Cisco Systems](#)