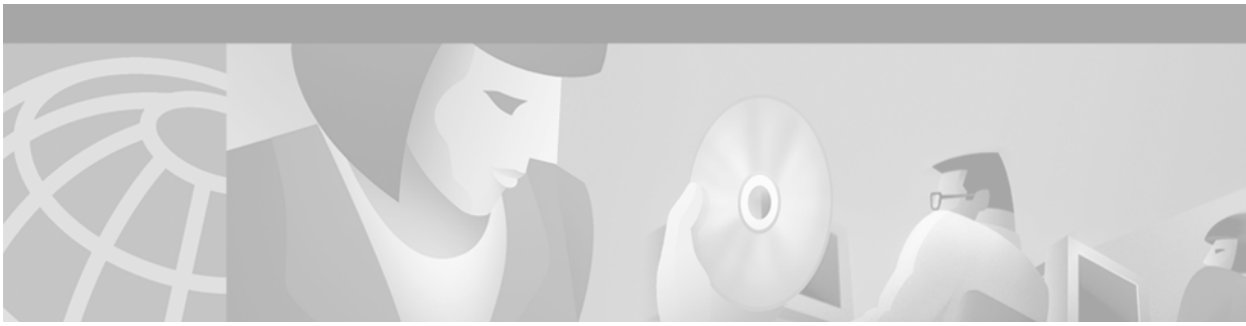

Link Fragmentation and Interleaving for Frame Relay and ATM Virtual Circuits

This feature module describes the Link Fragmentation and Interleaving for Frame Relay and ATM Virtual Circuits feature. It includes information such as the benefits of the new feature, related documents, and supported platforms.

This document contains the following sections:

Feature Overview

The Link Fragmentation and Interleaving for Frame Relay and ATM Virtual Circuits feature supports the transport of real-time (voice) and other (data) traffic on lower-speed Frame Relay and ATM virtual circuits (VCs) without causing excessive delay to the real-time traffic.

This new feature implements link fragmentation and interleaving (LFI) using multilink PPP (MLP) over Frame Relay and ATM. The feature enables delay-sensitive real-time packets and packets that are not real-time data to share the same link by fragmenting the long data packets into a sequence of smaller data packets (fragments). The fragments are interleaved with the real-time packets. On the receiving side of the link, the fragments are reassembled and the packet reconstructed. This method of fragmenting and interleaving helps guarantee the appropriate quality of service (QoS) for the real-time traffic.

Before the introduction of this new feature, MLP supported packet fragmentation and interleaving at the bundle layer; however, it did not support interleaving on Frame Relay or ATM. The Link Fragmentation and Interleaving for Frame Relay and ATM Virtual Circuits feature supports low-speed Frame Relay and ATM and also Frame Relay/ATM interworking (FRF.8).

Benefits

End-to-End Voice over IP Quality

This new feature enhances Voice over IP (VoIP) quality of service (QoS) by preventing delay, delay variation (jitter), and packet loss for voice traffic on low speed ATM-to-ATM and ATM-to-Frame Relay networks.

Interoperability with Other QoS Features

The Link Fragmentation and Interleaving for Frame Relay and ATM Virtual Circuits feature works concurrently with and on the same switching path as other QoS features, ensuring high quality and scalable VoIP deployment. This feature works with the following QoS features:

- Frame Relay traffic shaping
- Low latency queueing
- Class-based weighted fair queueing (CBWFQ)

Restrictions

The following restrictions apply to the Link Fragmentation and Interleaving for Frame Relay and ATM Virtual Circuits feature:

- Only one link per MLP bundle is supported. If more than one link is used, there is no way of knowing which link is doing the LFI.
- Only voice over IP is supported; voice over Frame Relay and voice over ATM are not supported.

Related Features and Technologies

- Frame Relay/ATM interworking (FRF.8)
- Frame Relay fragmentation (FRF.12)
- Frame Relay traffic shaping

Related Documents

- *Cisco IOS Quality of Service Solutions Configuration Guide*, Release 12.1
- *Cisco IOS Quality of Service Solutions Command Reference*, Release 12.1
- *Cisco IOS Wide-Area Networking Configuration Guide*, Release 12.1
- *Cisco IOS Wide-Area Networking Command Reference*, Release 12.1

Supported Platforms

- Cisco 2600
- Cisco 3600 Series
- Cisco 7200 Series

Supported Standards, MIBs, and RFCs

Standards

No new or modified standards are supported.

MIBs

No new or modified MIBs are supported.

For descriptions of supported MIBs and how to use MIBs, see the Cisco MIB web site on Cisco Connection Online (CCO) at http://www.cisco.com/public/sw-center/netmgmt/cmtk/mibs.shtml.

RFCs

RFC 1990, *The PPP Multilink Protocol (MP).*

Prerequisites

- Frame Relay traffic shaping must be configured on Frame Relay interfaces.
- Per-VC FIFO queueing must be configured on the Frame Relay and ATM VCs associated with MLP.
- MLP over ATM must use the following ATM network modules:
  - Multiport T1/E1 ATM Network Module with Inverse Multiplexing over ATM
  - ATM OC-3 Network Module
  - Enhanced ATM Port Adapter

Configuration Tasks

See the following sections for configuration tasks for the Link Fragmentation and Interleaving for Frame Relay and ATM Virtual Circuits feature. Each task in the list is identified as optional or required.

- Configuring LFI Using MLP over Frame Relay (Required)
- Configuring LFI Using MLP over ATM (Required)
- Verifying LFI for Frame Relay and ATM (Optional)

Configuring LFI Using MLP over Frame Relay

To configure LFI using MLP over Frame Relay, perform the tasks in the following sections:

- Configuring LFI Using MLP in a Virtual-Template Interface
- Associating the Virtual Template Interface with a Frame Relay PVC

Configuring LFI Using MLP in a Virtual-Template Interface

To configure LFI using MLP in a virtual template interface, use the following interface configuration commands:

	Command	Purpose
Step 1	`Router(config)# interface virtual-template number`	Creates a virtual template and enters interface configuration mode.
Step 2	`Router(config-if)# bandwidth kilobits`	Sets the bandwidth value for an interface.
Step 3	`Router(config-if)# service-policy output policy-name`	Attaches the specified policy map to the output interface.
Step 4	`Router(config-if)# ppp multilink`	Enables MLP on the interface.
Step 5	`Router(config-if)# ppp multilink fragment-delay milliseconds`	Configures the maximum delay allowed for transmission of a packet fragment on an MLP bundle.
Step 6	`Router(config-if)# ppp multilink interleave`	Enables interleaving of RTP packets among the fragments of larger packets on an MLP bundle.

The ideal fragment size should allow the fragments to fit into an exact multiple of ATM cells. The fragment size for MLP over ATM can be calculated using the following formula:

fragment size = 48 x number of cells – 10

Fragment size at the MLP bundle can be configured using the following formula:

fragment size = bandwidth x fragment-delay / 8

Associating the Virtual Template Interface with a Frame Relay PVC

To associate the virtual template interface with a Frame Relay PVC, use the following commands beginning in global configuration mode:

	Command	Purpose
Step 1	`Router(config)# interface type number`	Configures an interface type and enters interface configuration mode.
Step 2	`Router(config-if)# frame-relay traffic-shaping`	Enables Frame Relay traffic shaping on the interface.
Step 3	`Router(config-if)# frame-relay interface-dlci dlci [ppp virtual-template-name]`	Associates a virtual template interface with a Frame Relay DLCI.[1]
Step 4	`Router(config-if)# class name`	Associates a Frame Relay map class with a DLCI.

1. DLCI = data-link connection identifier

Configuring LFI Using MLP over ATM

LFI using MLP can be configured over ATM using a virtual template interface or a dialer interface. To configure LFI using MLP over ATM using a virtual template interface or dialer interface, perform the tasks in the following sections:

- Configuring LFI Using MLP on a Virtual Template Interface
- Associating the Virtual Template Interface with an ATM PVC
- Configuring LFI Using MLP on a Dialer Interface
- Associating the Dialer Interface with an ATM PVC

Configuring LFI Using MLP on a Virtual Template Interface

To configure LFI using MLP on a virtual template interface, use the following interface configuration commands:

	Command	Purpose
Step 5	`Router(config)# interface virtual-template number`	Creates a virtual template and enters interface configuration mode.
Step 6	`Router(config-if)# bandwidth kilobits`	Sets the bandwidth value for an interface.
Step 7	`Router(config-if)# service-policy output policy-name`	Attaches the specified policy map to the output interface.
Step 8	`Router(config-if)# ppp multilink`	Enables MLP on the interface.
Step 9	`Router(config-if)# ppp multilink fragment-delay milliseconds`	Configures the maximum delay allowed for transmission of a packet fragment on an MLP bundle.
Step 10	`Router(config-if)# ppp multilink interleave`	Enables interleaving of RTP packets among the fragments of larger packets on an MLP bundle.

The ideal fragment size for MLP over ATM should allow the fragments to fit into an exact multiple of ATM cells. The fragment size for MLP over ATM can be calculated using the following formula:

fragment size = 48 x number of cells – 10

Fragment size at the MLP bundle can be configured using the following formula:

fragment size = bandwidth x fragment-delay / 8

Note To attach a service policy to a multilink ppp bundle configured through a virtual template, make sure that the multilink ppp bundle interface is operational. If the interface is not operational, attaching the service policy fails. If a multilink ppp bundle interface is configured through a virtual template, at least two virtual access interfaces are configured, (that is, virtual-access 1 and virtual-access 2). One of these virtual access interfaces is a ppp interface and the other is a multilink ppp bundle interface.

When a service policy is attached to a virtual template, the error message "Class Based Weighted Fair Queuing not supported on interface virtual-access1" appears if the virtual-access1 interface is the ppp interface. Since the service policy is successfully attached to the multilink ppp bundle interface, this is not an error condition. To verify whether the service policy is attached correctly, use the **show interfaces** command and review the queuing policy.

Associating the Virtual Template Interface with an ATM PVC

To associate the virtual template interface with an ATM PVC, use the following commands beginning in global configuration mode:

	Command	Purpose

Step 1	`Router(config)# interface atm slot/0`
	or
	`Router(config)# interface atm slot/port`
		Specifies the ATM interface type and enters interface configuration mode.[1]

Step 2	`Router(config-if)# pvc [name] vpi/vci`	Creates an ATM PVC.
Step 3	`Router(config-if-atm-vc)# abr output-pcr output-mcr`	Selects ABR[2] QoS and configures the output peak cell rate and output minimum guaranteed cell rate for an ATM PVC.
Step 4	`Router(config-if-atm-vc)# protocol ppp virtual-template number`	Specifies that PPP is established over the ATM PVC using the configuration from the specified virtual template.

1. To determine the correct form of the **interface atm** command, consult your ATM network module, port adapter, or route documentation.
2. ABR = available bit rate

Configuring LFI Using MLP on a Dialer Interface

To configure LFI using MLP in a dialer interface, use the following commands beginning in global configuration mode:

	Command	Purpose
Step 5	`Router(config)# interface dialer number`	Creates a dialer interface and enters interface configuration mode.
Step 6	`Router(config-if)# bandwidth kilobits`	Sets the bandwidth value for an interface.

	Command	Purpose

Step 7	`Router(config-if)# ip address ip-address mask`
	or
	`Router(config-if)# ip unnumbered type number`
		Configures the IP address for the interface.
		Enables IP processing on a serial interface without assigning an explicit IP address to the interface.

Step 8	`Router(config-if)# encapsulation ppp`	Enables PPP encapsulation on the interface.
Step 9	`Router(config-if)# dialer pool number`	For a dialer interface, specifies which dialing pool to use to connect to a specific destination subnetwork.
Step 10	`Router(config-if)# service-policy output name`	Attaches a policy map to an output interface or VC to be used as the service policy for that interface or VC.
Step 11	`Router(config-if)# ppp authentication chap`	(Optional) Enables CHAP[1] on the interface.
Step 12	`Router(config-if)# ppp chap hostname name`	(Optional) Creates a pool of dialup routers that all appear to be the same host when authenticating with CHAP.
Step 13	`Router(config-if)# ppp chap password secret`	(Optional) Enables a router calling a collection of routers that do not support this command (such as routers running older Cisco IOS software images) to configure a common CHAP secret password to use in response to challenges from an unknown peer.
Step 14	`Router(config-if)# ppp multilink`	Enables MLP on the interface.
Step 15	`Router(config-if)# ppp multilink fragment-delay milliseconds`	Configures the maximum delay allowed for transmission of a packet fragment on an MLP bundle.
Step 16	`Router(config-if)# ppp multilink interleave`	Enables interleaving of RTP packets among the fragments of larger packets on an MLP bundle.

1. CHAP = Challenge Handshake Authentication Protocol

Associating the Dialer Interface with an ATM PVC

To associate a dialer interface with an ATM PVC, use the following commands beginning in global configuration mode:

	Command	Purpose

Step 1	`Router(config)# interface atm slot/0`
	or
	`Router(config)# interface atm slot/port`
		Specifies the ATM interface type and enters interface configuration mode.[1]

Step 2	`Router(config-if)# pvc [name] vpi/vci`	Creates an ATM PVC.
Step 3	`Router(config-if-atm-vc)# abr output-pcr output-mcr`	Selects ABR QoS and configures the output peak cell rate and output minimum guaranteed cell rate for an ATM PVC.

	Command	Purpose
Step 4	`Router(config-if-atm-vc)# encapsulation aal5mux ppp dialer`	Specifies that the encapsulation type will be PPP and that the PVC will be associated with a dialer interface.
Step 5	`Router(config-if-atm-vc)# dialer pool-member` *number*	Configures the interface to be a member of a dialer profile dialing pool.

1. To determine the correct form of the **interface atm** command, consult your ATM network module, port adapter, or route documentation.

Verifying LFI for Frame Relay and ATM

To display information about LFI for Frame Relay and ATM, use the following privileged EXEC commands:

Command	Purpose
`Router# show frame-relay pvc` *dlci*	Displays statistics about PVCs for Frame Relay interfaces.
`Router# show interfaces`	Displays interleaving statistics. Interleaving data is displayed only if interleaving occurs.
`Router# show ppp multilink`	Displays bundle information for the MLP bundles and their PPP links in the router.

Monitoring and Maintaining LFI for Frame Relay and ATM

To monitor LFI for Frame Relay and ATM, use the following privileged EXEC commands:

Command	Purpose
`Router# debug ppp multilink fragments`	Displays information about individual multilink fragments and important multilink events.
`Router# debug voice RTP`	Displays information about the interleaving of voice and data packets.

Note The **debug ppp multilink fragments** and **debug voice RTP** commands have memory overhead and should not be used when memory is scarce or when traffic is very high.

Configuration Examples

This section provides the following configuration examples:

- LFI over Frame Relay Using a Virtual Template Interface Configuration Example
- LFI over ATM Using a Virtual Template Interface Configuration Example
- LFI over ATM Using a Dialer Interface Configuration Example

LFI over Frame Relay Using a Virtual Template Interface Configuration Example

The following example shows the configuration of LFI using MLP over Frame Relay using a virtual template interface:

```
hostname router1
!
username cisco-1 password 7 140417081E013E
!
class-map cba
  match access-group 100
!
policy-map abc
  class cba
    priority 48
!
interface Serial5/0
 no ip address
 encapsulation frame-relay
 frame-relay traffic-shaping
!
! The following commands enable PPP on and associate "Virtual-Template1 with DLCI 16.
interface Serial5/0.1 point-to-point
 frame-relay interface-dlci 16 ppp Virtual-Template1
  class mlp
!
! The following commands configure MLP using LFI on "Virtual-Template1."
interface Virtual-Template1
 bandwidth 78
 ip unnumbered serial 5/0
 ip mroute-cache
 service-policy output abc
 ppp authentication chap
 ppp chap hostname router2
 ppp multilink
 ppp multilink fragment-delay 8
 ppp multilink interleave
!
map-class frame-relay mlp
 frame-relay cir 64000
 frame-relay bc 300
 frame-relay be 0
 no frame-relay adaptive-shaping
!
access-list 100 permit udp any any precedence critical
!
! The following commands configure Voice over IP.
dial-peer voice 5 voip
 destination-pattern 1222
 session target ipv4:131.180.80.10
 dtmf-relay cisco-rtp
 ip precedence 5
!
dial-peer voice 1 pots
 destination-pattern 1333
 port 2/1/0
```

LFI over ATM Using a Virtual Template Interface Configuration Example

The following example shows the configuration of LFI using MLP on an ATM interface. This configuration uses a virtual template interface.

```
hostname router1
!
username cisco-1 password 7 36497A4872384A
!
class-map xyz
  match access-group 100
!
policy-map xyz
  class xyz
   priority 48
!
interface ATM4/0
 no ip address
 no atm ilmi-keepalive
!
! The following commands enable PPP on and associate "Virtual-Template1 with PVC 0/32.
int atm4/0.1 point-to-point
 pvc 0/32
  abr 100 80
  protocol ppp Virtual-Template1
!
! The following commands configure MLP using LFI on "Virtual-Template1."
interface Virtual-Template1
 bandwidth 78
 ip unnumbered ATM4/0
 ip mroute-cache
 service-policy output xyz
 ppp authentication chap
 ppp chap hostname router2
 ppp multilink
 ppp multilink fragment-delay 8
 ppp multilink interleave
!
access-list 100 permit udp any any precedence critical
!
! The following commands configure Voice over IP.
dial-peer voice 5 voip
 destination-pattern 1222
 session target ipv4:131.180.80.10
 dtmf-relay cisco-rtp
 ip precedence 5
!
dial-peer voice 1 pots
 destination-pattern 1333
 port 2/1/0
```

LFI over ATM Using a Dialer Interface Configuration Example

The following example shows the configuration of LFI using MLP on an ATM interface. This configuration uses a dialer interface.

```
!
class-map xyz
  match access-group 100
!
policy-map xyz
  class xyz
   priority 48
!
! The following commands configure MLP using LFI on dialer interface 1.
interface Dialer1
 bandwidth 86
 ip address 192.168.1.18 255.255.255.252
 encapsulation ppp
 dialer pool 1
 service-policy output abc
 authentication chap
 ppp chap hostname router2
 ppp chap password 0 password
 ppp multilink
 ppp multilink fragment-delay 8
 ppp multilink interleave
!
! The following commands associate PVC 1/32 with dialer interface 1.
interface ATM4/0
 pvc 1/32
  abr 100 80
  encapsulation aal5mux ppp dialer
  dialer pool-member 1
!
access-list 100 permit udp any any precedence critical
!
! The following commands configure Voice over IP.
dial-peer voice 5 voip
 destination-pattern 1222
 session target ipv4:131.180.80.10
 dtmf-relay cisco-rtp
 ip precedence 5
!
dial-peer voice 1 pots
 destination-pattern 1333
 port 2/1/0
```

Command Reference

There are no new or modified commands for the Link Fragmentation and Interleaving for Frame Relay and ATM Virtual Circuits feature.

Glossary

CBWFQ—class-based weighted fair queueing. Extends the standard WFQ functionality to provide support for user-defined traffic classes.

class-based weighted fair queueing—See CBWFQ.

FIFO queueing—first-in, first-out queueing. FIFO involves buffering and forwarding of packets in the order of arrival. FIFO embodies no concept of priority or classes of traffic. There is only one queue, and all packets are treated equally. Packets are sent out an interface in the order in which they arrive.

FRF.8—The Frame Relay/ATM Interworking Implementation Agreement.

LFI—link fragmentation and interleaving. Method of fragmenting large packets and then queueing the fragments between small packets.

MLP—multilink PPP. Method of splitting, recombining, and sequencing datagrams across multiple logical links.

multilink PPP—See MLP.

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

VC—virtual circuit. Logical circuit created to ensure reliable communication between two network devices. A VC is defined by a VPI/VCI pair and can be either permanent (PVC) or switched (SVC).

Voice over IP—method of transporting voice traffic over an IP network. In Voice over IP, the voice signal is segmented into frames, which are then coupled in groups of two and stored in voice packets. These voice packets are transported using a method that is in compliance with ITU-T specification H.323.

weighted fair queueing—See WFQ.

WFQ—weighted fair queueing. Congestion management algorithm that identifies conversations (in the form of traffic streams), separates packets that belong to each conversation, and ensures that capacity is shared fairly among these individual conversations. WFQ is an automatic way of stabilizing network behavior during congestion and results in increased performance and reduced retransmission.