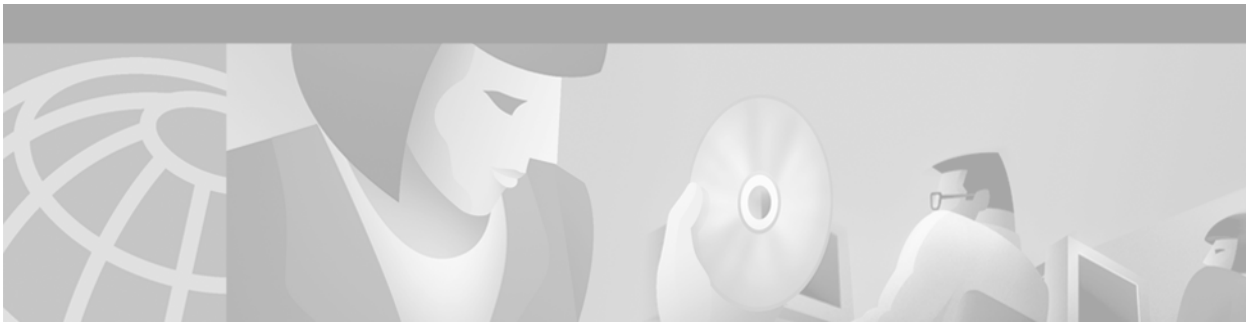

Configuring Link Fragmentation and Interleaving for Frame Relay and ATM Virtual Circuits

This chapter describes the tasks for configuring the Link Fragmentation and Interleaving (LFI) for Frame Relay and ATM Virtual Circuits (VCs) feature, and it includes example configurations.

For complete conceptual information, see the section "Link Fragmentation and Interleaving for Frame Relay and ATM VCs" in the chapter "Link Efficiency Mechanisms Overview" in this book. To locate documentation of commands that appear in this chapter, use the command reference master index or search online.

To identify the hardware platform or software image information associated with a feature, use the Feature Navigator on Cisco.com to search for information about the feature or refer to the software release notes for a specific release. For more information, see the "Identifying Supported Platforms" section in the "Using Cisco IOS Software" chapter in this book.

LFI for Frame Relay and ATM VCs Configuration Task List

To configure the LFI for Frame Relay and ATM VCs feature, perform the tasks described in the following sections. The tasks in the two sections are required; the task in the remaining section is optional.

- Configuring LFI Using Multilink Point-to-Point Protocol over Frame Relay (Required)
- Configuring LFI Using MLP over ATM (Required)
- Verifying LFI for Frame Relay and ATM (Optional)
- Monitoring and Maintaining LFI for Frame Relay and ATM (Optional)

See the end of this chapter for the section "LFI for Frame Relay and ATM VCs Configuration Examples."

Configuring LFI Using Multilink Point-to-Point Protocol over Frame Relay

To configure LFI using Multilink Point-to-Point Protocol (MLP) over Frame Relay, perform the tasks described in the following sections. The tasks in both sections are required.

- Configuring LFI Using MLP in a Virtual Template Interface (Required)
- Associating the Virtual Template Interface with a Frame Relay Permanent Virtual Circuit (Required)

Configuring LFI Using MLP in a Virtual Template Interface

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

	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 * number of cells – 10

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

fragment size = bandwidth * fragment-delay/8

Associating the Virtual Template Interface with a Frame Relay Permanent Virtual Circuit

To associate the virtual template interface with a Frame Relay permanent virtual circuit (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 data-link connection identifier (DLCI).
Step 4	`Router(config-if)# class name`	Associates a Frame Relay map class with a DLCI.

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 either the first two or the second two sections:

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

Configuring LFI Using MLP on a Virtual Template Interface

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

	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 Real-Time Protocol (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.
		Note To determine the correct form of the **interface** command, consult your ATM network module, port adapter, or router documentation

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 available bit rate (ABR) 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.

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 1	`Router(config)# interface dialer number`	Creates a dialer interface and enters interface configuration mode.
Step 2	`Router(config-if)# bandwidth kilobits`	Sets the bandwidth value for an interface.

	Command	Purpose
Step 3	`Router(config-if)# ip address ip-address mask`	Configures the IP address for the interface.
	or	
	`Router(config-if)# ip unnumbered type number`	Enables IP processing on a serial interface without assigning an explicit IP address to the interface.
Step 4	`Router(config-if)# encapsulation ppp`	Enables PPP encapsulation on the interface.
Step 5	`Router(config-if)# dialer pool number`	For a dialer interface, specifies which dialing pool to use to connect to a specific destination subnetwork.
Step 6	`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 7	`Router(config-if)# ppp authentication chap`	(Optional) Enables Challenge Authentication Protocol (CHAP) on the interface.
Step 8	`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 9	`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 10	`Router(config-if)# ppp multilink`	Enables MLP on the interface.
Step 11	`Router(config-if)# ppp multilink fragment-delay milliseconds`	Configures the maximum delay allowed for transmission of a packet fragment on an MLP bundle.
Step 12	`Router(config-if)# ppp multilink interleave`	Enables interleaving of RTP packets among the fragments of larger packets on an MLP bundle.

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`	Specifies the ATM interface type and enters interface configuration mode.
	or	
	`Router(config)# interface atm slot/port`	**Note** To determine the correct form of the **interface atm** command, consult your ATM network module, port adapter, or router documentation
Step 2	`Router(config-if)# pvc [name] vpi/vci`	Creates an ATM PVC.

	Command	Purpose
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.
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.

Verifying LFI for Frame Relay and ATM

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

Command	Purpose
`Router# show frame-relay pvc dlci`	Displays statistics about PVCs for Frame Relay interfaces.
`Router# show interfaces`	Displays interleaving statistics if any 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 commands in privileged EXEC mode, as needed:

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.

Caution 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.

LFI for Frame Relay and ATM VCs Configuration Examples

This section provides the following LFI for Frame Relay and ATM VCs 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

For information about configuring LFI for Frame Relay and ATM VCs, see "LFI for Frame Relay and ATM VCs Configuration Task List" in this chapter.

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 address 192.168.47.6 255.255.255.252
 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 address 192.168.47.17 255.255.255.252
 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
```