Link Fragmentation and Interleaving for Frame Relay and ATM Virtual Circuits

First Published:
Last Updated:

The Link Fragmentation and Interleaving for Frame Relay and ATM Virtual Circuits feature supports the transport of real-time (voice) and non-real-time (data) traffic on lower-speed Frame Relay and ATM permanent virtual circuits (PVCs) without causing excessive delay of real-time traffic. (This feature does not support switched virtual circuits.)

This 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 non-real-time packets to share the same link by fragmenting the long data packets into a sequence of smaller data packets (fragments). The fragments are then interleaved with the real-time packets. On the receiving side of the link, the fragments are reassembled, and the packets are reconstructed. This method of fragmenting and interleaving helps guarantee the appropriate quality of service (QoS) for the real-time traffic.

Without this feature, MLP supported packet fragmentation and interleaving at the bundle layer; however, it did not support interleaving on Frame Relay or ATM. This feature supports low-speed Frame Relay and ATM as well as Frame Relay/ATM interworking (FRF.8) and Frame Relay fragmentation (FRF.12).

History for the LFI for Frame Relay and ATM Virtual Circuits Feature

Release	Modification
12.1(5)T	This feature was introduced.
12.2(28)SB	This feature was integrated into Cisco IOS Release 12.2(28)SB.

Finding Support Information for Platforms and Cisco IOS Software Images

Use Cisco Feature Navigator to find information about platform support and Cisco IOS software image support. Access Cisco Feature Navigator at http://www.cisco.com/go/fn. You must have an account on Cisco.com. If you do not have an account or have forgotten your username or password, click **Cancel** at the login dialog box and follow the instructions that appear.

CISCO SYSTEMS

Corporate Headquarters:
Cisco Systems, Inc., 170 West Tasman Drive, San Jose, CA 95134-1706 USA

Contents

- Prerequisites for LFI for Frame Relay and ATM PVCs, page 2
- Restrictions for LFI for Frame Relay and ATM PVCs, page 2
- Information About LFI for Frame Relay and ATM PVCs, page 2
- How to Configure LFI for Frame Relay and ATM PVCs, page 3
- Configuration Examples for LFI for Frame Relay and ATM PVCs, page 19
- Additional References, page 24
- Command Reference, page 25
- Glossary, page 27

Prerequisites for LFI for Frame Relay and ATM PVCs

The following prerequisites apply to the LFI for Frame Relay and ATM Virtual Circuits feature:

- For Frame Relay interfaces, Frame Relay traffic shaping must be configured.
- For Frame Relay and ATM PVCs associated with MLP, per-PVC FIFO queuing must be configured.
- MLP over ATM requires an ATM network module such as one of the following:
  - Multiport T1/E1 ATM Network Module with Inverse Multiplexing over ATM
  - ATM OC-3 Network Module
  - PA-A3 and PA-A6 Enhanced ATM Port Adapters
  - G.shdsl WAN Interface Cards
  - ADSL WAN Interface Cards

Restrictions for LFI for Frame Relay and ATM PVCs

The following restrictions apply to the LFI for Frame Relay and ATM Virtual Circuits feature:

- Only one link per MLP bundle is supported.
- Voice over Frame Relay and Voice over ATM are not supported (only Voice over IP is supported).

Information About LFI for Frame Relay and ATM PVCs

To configure the LFI for Frame Relay and ATM Virtual Circuits feature, you should understand the following concepts:

- Benefits of LFI for Frame Relay and ATM Virtual Circuits, page 3
- Memory Impact of LFI for Frame Relay and ATM Virtual Circuits, page 3
- Performance Impact of LFI for Frame Relay and ATM Virtual Circuits, page 3

Benefits of LFI for Frame Relay and ATM Virtual Circuits

End-to-End Voice over IP Quality

This feature enhances Voice over IP (VoIP) 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 LFI for Frame Relay and ATM Virtual Circuits feature works concurrently with (and on the same switching path as) other QoS features, which ensures high quality and scalable VoIP deployment. This feature works with the following QoS features:

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

Memory Impact of LFI for Frame Relay and ATM Virtual Circuits

This feature does not significantly increase memory requirements except when you have configured more than 200 ATM PVCs and use a separate virtual template (VT) for each PVC. If you want to use more than 200 ATM PVCs, you should create only one virtual template to be associated with all PVCs—this method decreases the memory requirement by about one third for one link per bundle and by greater than one third when multiple PVCs are bundled.

Performance Impact of LFI for Frame Relay and ATM Virtual Circuits

This feature does not significantly increase CPU usage. Also, this feature does not affect data forwarding performance (even with a large number of LFI sessions).

How to Configure LFI for Frame Relay and ATM PVCs

This section describes how to configure the LFI for Frame Relay and ATM Virtual Circuits feature and consists of the following configuration tasks:

- Configuring LFI Using MLP over Frame Relay, page 4 (required)
- Configuring LFI Using MLP on ATM Virtual Template Interfaces, page 6 (required)
- Configuring LFI Using MLP over ATM on Dialer Interfaces, page 12 (required)
- Verifying LFI for Frame Relay and ATM, page 17 (required)
- Troubleshooting LFI for Frame Relay and ATM, page 18 (optional)

Configuring LFI Using MLP over Frame Relay

This section shows how to configure LFI using MLP over Frame Relay. For each Frame Relay PVC, you configure LFI using MLP on a virtual template interface, and then you associate that virtual template interface with the PVC.

This section consists of the following configuration tasks:

- Configuring LFI Using MLP on a Virtual Template Interface, page 4 (required)
- Associating the Virtual Template Interface with a Frame Relay PVC, page 5 (required)

Configuring LFI Using MLP on a Virtual Template Interface

With this configuration method, you create a single virtual template that is used for each MLP LFI session.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **interface virtual-template** *number*
4. **bandwidth** *kilobits*
5. **service-policy output** *policy-name*
6. **ppp multilink**
7. **ppp multilink fragment-delay** *milliseconds*
8. **ppp multilink interleave**
9. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enters privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**interface virtual-template** *number* **Example:** `Router(config)# interface virtual-template 1`	Creates a virtual template and enters interface configuration mode.

	Command or Action	Purpose
Step 4	**bandwidth** *kilobits* **Example:** `Router(config-if)# bandwidth 78`	Sets the bandwidth value for an interface.
Step 5	**service-policy output** *policy-name* **Example:** `Router(config-if)# service-policy output abc`	Attaches the specified policy map to the output interface.
Step 6	**ppp multilink** **Example:** `Router(config-if)# ppp multilink`	Enables MLP on the interface.
Step 7	**ppp multilink fragment-delay** *milliseconds* **Example:** `Router(config-if)# ppp multilink fragment-delay 8`	Configures the maximum delay allowed for transmission of a packet fragment on an MLP bundle.
Step 8	**ppp multilink interleave** **Example:** `Router(config-if)# ppp multilink interleave`	Enables interleaving of Real-Time Transport Protocol (RTP) packets among the fragments of larger packets on an MLP bundle.
Step 9	**end** **Example:** `Router(config-if)# end`	Ends the configuration session and returns to privileged EXEC mode.

Associating the Virtual Template Interface with a Frame Relay PVC

To associate the virtual template interface with a Frame Relay PVC, use the following steps.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **interface** *type number*
4. **frame-relay traffic-shaping**
5. **frame-relay interface-dlci** *dlci* [**ppp** *virtual-template-name*]
6. **class** *name*
7. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enters privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**interface** *type number* **Example:** `Router(config)# interface Serial 5/0`	Configures an interface type and enters interface configuration mode.
Step 4	**frame-relay traffic-shaping** **Example:** `Router(config-if)# frame-relay traffic-shaping`	Enables Frame Relay traffic shaping on the interface.
Step 5	**frame-relay interface-dlci** *dlci* [**ppp** *virtual-template-name*] **Example:** `Router(config-if)# frame-relay interface-dlci 16 ppp virtual-template 1`	Associates a virtual template interface with a Frame Relay data-link connection identifier (DLCI).
Step 6	**class** *name* **Example:** `Router(config-if)# class mlp`	Associates a Frame Relay map class with a DLCI.
Step 7	**end** **Example:** `Router(config-if)# end`	Ends the configuration session and returns to privileged EXEC mode.

Configuring LFI Using MLP on ATM Virtual Template Interfaces

You can configure LFI using MLP over ATM by using one of two methods:

- Create one virtual template interface for *all* PVCs.
- Create one virtual template interface for *each* PVC.

The advantage of using the first method is that you need only one virtual template. This method overcomes the limitation of 200 virtual templates per router by letting you use the **ppp multilink group** command in ATM permanent virtual circuit configuration mode.

This section shows how to configure LFI using MLP over ATM using both of these methods. If you attempt to use both methods, the first method overrides the second method.

This section consists of the following configuration tasks:

- Prerequisites for Configuring LFI Using MLP on a Virtual Template Interface (Single VT), page 7 (required)
- Configuring LFI Using MLP on a Virtual Template Interface (Single VT), page 7 (required)
- Associating the Virtual Template Interface with an ATM PVC (Single VT), page 8 (required)
- Configuring LFI Using MLP on a Virtual Template Interface (One VT per PVC), page 10 (required)
- Associating the Virtual Template Interface with an ATM PVC (One VT per PVC), page 11 (required)

Prerequisites for Configuring LFI Using MLP on a Virtual Template Interface (Single VT)

Enabling PPP Encapsulation

Enabling LFI using MLP over ATM requires that you also configure PPP encapsulation (AAL5 MUX, Cisco proprietary, or LLC/SNAP) for the PVC when it will be part of the MLP bundle.

Ensuring That the Bundle Interface Is Operational

Before attaching a service policy to an MLP bundle configured through a virtual template, make sure that the bundle interface is operational. If the bundle interface is not operational, attaching the service policy fails. If an MLP 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 an MLP 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. Because the service policy is successfully attached to the MLP bundle interface, this is not an error condition. If you want to verify that the service policy is attached correctly, use the **show interfaces** command and review the queuing policy.

Configuring LFI Using MLP on a Virtual Template Interface (Single VT)

With this configuration method, you create a single virtual template that is used for each MLP LFI session. You set the per-LFI configuration parameters directly in ATM permanent virtual circuit configuration mode.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **interface virtual-template** *number*
4. **no ip address**
5. **ppp multilink**
6. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enters privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**interface virtual-template** *number* **Example:** `Router(config)# interface virtual-template 1`	Creates a virtual template and enters interface configuration mode.
Step 4	**no ip address** **Example:** `Router(config-if)# no ip address`	Disables IP processing for the virtual template.
Step 5	**ppp multilink** **Example:** `Router(config-if)# ppp multilink`	Enables MLP on the interface.
Step 6	**end** **Example:** `Router(config-if)# end`	Ends the configuration session and returns to privileged EXEC mode.

Associating the Virtual Template Interface with an ATM PVC (Single VT)

To associate the virtual template interface with an ATM PVC, use the following steps.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **interface atm** *slot/port*
4. **pvc** [*name*] *vpi/vci*
5. **vbr-nrt** *output-pcr output-scr*
6. **tx-ring-limit** *ring-limit*
7. **protocol ppp virtual-template** *number*
8. **ppp multilink group** *number*
9. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enters privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**interface atm** *slot*/*port* **Example:** `Router(config)# interface atm 2/0`	Specifies the ATM interface type and enters interface configuration mode. To determine the correct form of the **interface atm** command, see your ATM network module, port adapter, or router documentation.
Step 4	**pvc** [*name*] *vpi*/*vci* **Example:** `Router(config-if)# pvc 1/33`	Creates an ATM PVC and enters ATM permanent virtual circuit configuration mode.
Step 5	**vbr-nrt** *output-pcr output-scr* **Example:** `Router(config-if-atm-vc)# vbr-nrt 128 128`	Configures the variable bit rate-nonreal time (VBR-NRT) output peak cell rate (PCR) and output sustainable cell rate (SCR).
Step 6	**tx-ring-limit** *ring-limit* **Example:** `Router(config-if-atm-vc)# tx-ring-limit 3`	Limits the number of particles or packets that can be used on a transmission ring on the interface.
Step 7	**protocol ppp virtual-template** *number* **Example:** `Router(config-if-atm-vc)# protocol ppp virtual-template 1`	Specifies that PPP is established over the ATM PVC using the configuration from the specified virtual template.
Step 8	**ppp multilink group** *number* **Example:** `Router(config-if-atm-vc)# ppp multilink group 1`	Restricts a physical link to joining only a designated multilink-group interface.
Step 9	**end** **Example:** `Router(config-if-atm-vc)# end`	Ends the configuration session and returns to privileged EXEC mode.

Configuring LFI Using MLP on a Virtual Template Interface (One VT per PVC)

When you use this method to configure LFI using MLP, you must perform this procedure for each MLP LFI session. You can configure up to 200 virtual templates.

You should specify values for bandwidth and fragment delay to specify a fragment size that allows the fragments to fit into an exact multiple of ATM cells (each cell has 48 bytes of data). You calculate the ideal fragment size for MLP over ATM by using the following formula:

fragment size = (48 x number of cells) – 10

Then, you configure the bandwidth and fragment delay so that:

fragment size = (bandwidth x fragment delay) / 8

To configure LFI using MLP on a virtual template interface, use the following steps.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **interface virtual-template** *number*
4. **bandwidth** *kilobits*
5. **service-policy output** *policy-name*
6. **ppp multilink**
7. **ppp multilink fragment-delay** *milliseconds*
8. **ppp multilink interleave**
9. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enters privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**interface virtual-template** *number* **Example:** `Router(config)# interface virtual-template 1`	Creates a virtual template and enters interface configuration mode.
Step 4	**bandwidth** *kilobits* **Example:** `Router(config-if)# bandwidth 78`	Sets the bandwidth value for an interface.

	Command or Action	Purpose
Step 5	**service-policy output** *policy-name* **Example:** `Router(config-if)# service-policy output xyz`	Attaches the specified policy map to the output interface.
Step 6	**ppp multilink** **Example:** `Router(config-if)# ppp multilink`	Enables MLP on the interface.
Step 7	**ppp multilink fragment-delay** *milliseconds* **Example:** `Router(config-if)# ppp multilink fragment-delay 8`	Configures the maximum delay allowed for transmission of a packet fragment on an MLP bundle.
Step 8	**ppp multilink interleave** **Example:** `Router(config-if)# ppp multilink interleave`	Enables interleaving of RTP packets among the fragments of larger packets on an MLP bundle.
Step 9	**end** **Example:** `Router(config-if)# end`	Ends the configuration session and returns to privileged EXEC mode.

Associating the Virtual Template Interface with an ATM PVC (One VT per PVC)

To associate the virtual template interface with an ATM PVC, use the following steps.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **interface atm** *slot/port*
4. **pvc** [*name*] *vpi/vci*
5. **abr** *output-pcr output-mcr*
6. **protocol ppp virtual-template** *number*
7. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enters privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**interface atm** *slot*/*port* **Example:** `Router(config)# interface atm 2/0`	Specifies the ATM interface type and enters interface configuration mode. To determine the correct form of the **interface atm** command, see your ATM network module, port adapter, or router documentation.
Step 4	**pvc** [*name*] *vpi*/*vci* **Example:** `Router(config-if)# pvc 0/32`	Creates an ATM PVC and enters ATM permanent virtual circuit configuration mode.
Step 5	**abr** *output-pcr output-mcr* **Example:** `Router(config-if-atm-vc)# abr 100 80`	Selects available bit rate (ABR) QoS and configures the output peak cell rate and output minimum guaranteed cell rate for an ATM PVC.
Step 6	**protocol ppp virtual-template** *number* **Example:** `Router(config-if-atm-vc)# protocol ppp virtual-template 1`	Specifies that PPP is established over the ATM PVC using the configuration from the specified virtual template.
Step 7	**end** **Example:** `Router(config-if-atm-vc)# end`	Ends the configuration session and returns to privileged EXEC mode.

Configuring LFI Using MLP over ATM on Dialer Interfaces

This section describes how to configure LFI using MLP on a dialer interface and then associate that dialer interface with an ATM PVC and consists of the following configuration tasks:

- Prerequisites for Configuring LFI Using MLP on a Dialer Interface, page 13 (required)
- Configuring LFI Using MLP on a Dialer Interface, page 13 (required)
- Associating the Dialer Interface with an ATM PVC, page 15 (required)

Prerequisites for Configuring LFI Using MLP on a Dialer Interface

Enabling LFI using MLP over ATM requires that you also configure PPP encapsulation (AAL5 MUX, Cisco proprietary, or LLC/SNAP) for the PVC when it will be part of the MLP bundle.

Configuring LFI Using MLP on a Dialer Interface

You should specify values for bandwidth and fragment delay to specify a fragment size that allows the fragments to fit into an exact multiple of ATM cells (each cell has 48 bytes of data). You calculate the ideal fragment size for MLP over ATM by using the following formula:

fragment size = (48 x number of cells) – 10

Then, you configure the bandwidth and fragment delay so that:

fragment size = (bandwidth x fragment delay) / 8

To configure LFI using MLP on a dialer interface, use the following steps.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **interface dialer** *number*
4. **bandwidth** *kilobits*
5. **ip address** *ip-address mask*
   or
   ip unnumbered *type number*
6. **encapsulation ppp**
7. **dialer pool** *number*
8. **service-policy output** *name*
9. **ppp authentication chap**
10. **ppp chap hostname** *name*
11. **ppp chap password** *secret*
12. **ppp multilink**
13. **ppp multilink fragment-delay** *milliseconds*
14. **ppp multilink interleave**
15. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enters privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**interface dialer** *number* **Example:** `Router(config)# interface dialer 1`	Creates a dialer interface and enters interface configuration mode.
Step 4	**bandwidth** *kilobits* **Example:** `Router(config-if)# bandwidth 86`	Sets the bandwidth value for an interface.
Step 5	**ip address** *ip-address mask* or **ip unnumbered** *type number* **Example:** `Router(config-if)# ip address 192.168.1.18 255.255.255.252`	Configures the IP address for the interface. or Enables IP processing on a serial interface without assigning an explicit IP address to the interface.
Step 6	**encapsulation ppp** **Example:** `Router(config-if)# encapsulation ppp`	Enables PPP encapsulation on the interface.
Step 7	**dialer pool** *number* **Example:** `Router(config-if)# dialer pool 1`	For a dialer interface, specifies which dialing pool to use to connect to a specific destination subnetwork.
Step 8	**service-policy output** *name* **Example:** `Router(config-if)# service-policy output abc`	Attaches a policy map to an output interface or VC to be used as the service policy for that interface or VC.
Step 9	**ppp authentication chap** **Example:** `Router(config-if)# ppp authentication chap`	(Optional) Enables Challenge Handshake Authentication Protocol (CHAP) on the interface.

	Command or Action	Purpose
Step 10	**ppp chap hostname** *name* **Example:** `Router(config-if)# ppp chap hostname router2`	(Optional) Creates a pool of dialup routers that all appear to be the same host when authenticating with CHAP.
Step 11	**ppp chap password** *secret* **Example:** `Router(config-if)# ppp chap password pass1`	(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 12	**ppp multilink** **Example:** `Router(config-if)# ppp multilink`	Enables MLP on the interface.
Step 13	**ppp multilink fragment-delay** *milliseconds* **Example:** `Router(config-if)# ppp multilink fragment-delay 8`	Configures the maximum delay allowed for transmission of a packet fragment on an MLP bundle.
Step 14	**ppp multilink interleave** **Example:** `Router(config-if)# ppp multilink interleave`	Enables interleaving of RTP packets among the fragments of larger packets on an MLP bundle.
Step 15	**end** **Example:** `Router(config-if)# end`	Ends the configuration session and returns to privileged EXEC mode.

Associating the Dialer Interface with an ATM PVC

To associate the dialer interface with an ATM PVC, use the following steps.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **interface atm** *slot*/*port*
4. **pvc** [*name*] *vpi*/*vci*
5. **abr** *output-pcr output-mcr*
6. **encapsulation aal5mux ppp dialer**
7. **dialer pool-member** *number*
8. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enters privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**interface atm** *slot*/*port* **Example:** `Router(config)# interface atm 2/0`	Specifies the ATM interface type and enters interface configuration mode. To determine the correct form of the **interface atm** command, see your ATM network module, port adapter, or router documentation.
Step 4	**pvc** [*name*] *vpi*/*vci* **Example:** `Router(config-if)# pvc 1/32`	Creates an ATM PVC and enters ATM permanent virtual circuit configuration mode.
Step 5	**abr** *output-pcr output-mcr* **Example:** `Router(config-if-atm-vc)# abr 100 80`	Selects available bit rate QoS and configures the output peak cell rate and output minimum guaranteed cell rate for an ATM PVC.
Step 6	**encapsulation aal5mux ppp dialer** **Example:** `Router(config-if-atm-vc)# encapsulation aal5mux ppp dialer`	Specifies that the encapsulation type is PPP and that the PVC is associated with a dialer interface.
Step 7	**dialer pool-member** *number* **Example:** `Router(config-if-atm-vc)# dialer pool-member 1`	Configures the interface to be a member of a dialer profile dialing pool.
Step 8	**end** **Example:** `Router(config-if-atm-vc)# end`	Ends the configuration session and returns to privileged EXEC mode.

Verifying LFI for Frame Relay and ATM

To display information about LFI for Frame Relay and ATM, use the following **show** commands in privileged EXEC mode. You can use these commands in any combination or order.

The **show atm pvc** command displays the ATM PVC information, and the **show ppp multilink** command displays the PPP information. You can use these two commands together to help determine any problems with the association of a PVC to an MLP LFI bundle.

SUMMARY STEPS

1. **enable**
2. **show atm pvc**
3. **show atm vc**
4. **show frame-relay pvc** *dlci*
5. **show interfaces**
6. **show interfaces virtual-access**
7. **show ppp multilink**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enters privileged EXEC mode. • Enter your password if prompted.
Step 2	**show atm pvc** **Example:** `Router# show atm pvc`	Displays LFI-related information such as the virtual access interfaces.
Step 3	**show atm vc** **Example:** `Router# show atm vc`	Displays all ATM PVCs and switched virtual circuits (SVCs) and their traffic information.
Step 4	**show frame-relay pvc** *dlci* **Example:** `Router# show frame-relay pvc 110`	Displays statistics about PVCs for Frame Relay interfaces.
Step 5	**show interfaces** **Example:** `Router# show interfaces`	Displays interleaving statistics. Interleaving data is displayed only if interleaving occurs.

	Command or Action	Purpose
Step 6	**show interfaces virtual-access** **Example:** `Router# show interfaces virtual-access`	Displays multilink bundle information.
Step 7	**show ppp multilink** **Example:** `Router# show ppp multilink`	Displays bundle information for the MLP bundles and their PPP links in the router.

Troubleshooting LFI for Frame Relay and ATM

To troubleshoot LFI for Frame Relay and ATM, use the following **debug** commands in privileged EXEC mode. You can use these commands in any combination or order.

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

SUMMARY STEPS

1. **enable**
2. **debug condition interface** *interface-type interface-number* [**dlci** *dlci*] [**vc** {*vci* | *vpi*/*vci*}]
3. **debug atm events**
4. **debug atm lfi**
5. **debug ppp multilink events**
6. **debug ppp multilink fragments**
7. **debug voice RTP**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enters privileged EXEC mode. • Enter your password if prompted.
Step 2	**debug condition interface** *interface-type interface-number* [**dlci** *dlci*] [**vc** {*vci* | *vpi*/*vci*}] **Example:** `Router# debug condition interface atm1/0 vc 1/100`	Limits output for some debugging commands based on the interface and virtual circuit.

	Command or Action	Purpose
Step 3	**debug atm events** **Example:** `Router# debug atm events`	Displays ATM events that occur on all ATM PVCs.
Step 4	**debug atm lfi** **Example:** `Router# debug atm lfi`	Displays general information about MLP LFI over ATM PVCs.
Step 5	**debug ppp multilink events** **Example:** `Router# debug ppp multilink events`	Displays information about events affecting multilink groups established for bandwidth allocation control protocol (BACP).
Step 6	**debug ppp multilink fragments** **Example:** `Router# debug ppp multilink fragments`	Displays information about individual multilink fragments and important multilink events.
Step 7	**debug voice RTP** **Example:** `Router# debug voice RTP`	Displays information about the interleaving of voice and data packets.

Configuration Examples for LFI for Frame Relay and ATM PVCs

This section provides the following configuration examples:

- Configuring LFI Using MLP over Frame Relay: Example, page 20
- Configuring LFI Using MLP over ATM on Virtual Template Interfaces: Examples, page 21
- Configuring LFI Using MLP over ATM on Dialer Interfaces: Example, page 23

Configuring LFI Using MLP over Frame Relay: Example

The following example shows how to configure 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 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 ipv4:172.16.80.10
 dtmf-relay cisco-rtp
 ip precedence 5
!
dial-peer voice 1 pots
 destination-pattern 1333
 port 2/1/0
```

Configuring LFI Using MLP over ATM on Virtual Template Interfaces: Examples

The following examples show how to configure LFI using MLP on ATM virtual template interfaces:

Configuring LFI Using MLP on Virtual Template Interfaces (Single VT)

The following example shows how to configure LFI using MLP over ATM on a virtual template interface. In this example, you associate the PVC with the MLP bundle directly in ATM permanent virtual circuit configuration mode.

```
! The following commands configure the multilink interface.
interface Multilink1
 ip address 10.6.6.1 255.255.255.0
 service-policy output cisco
 ppp multilink
 ppp multilink fragmentation
 ppp multilink fragment-delay 8
 ppp multilink interleave
 ppp multilink group 1
 exit
!
! The following commands create the virtual template.
interface Virtual-Template1
 no ip address
 ppp multilink
 exit
!
! The following commands associate the PVC with the multilink interface.
interface ATM4/0
 pvc 1/33
  vbr-nrt 128 128
  tx-ring-limit 3
  protocol ppp Virtual-Template1
! The following command associates the PVC with the MLP bundle.
  ppp multilink group 1
  end
```

Configuring LFI Using MLP on Virtual Template Interfaces (One VT per PVC)

The following example shows how to configure LFI using MLP over ATM on 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 and associate Virtual-Template1 with PVC 0/32.
interface 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 ipv4:172.16.80.10
 dtmf-relay cisco-rtp
 ip precedence 5
!
dial-peer voice 1 pots
 destination-pattern 1333
 port 2/1/0
```

Configuring LFI Using MLP over ATM on Dialer Interfaces: Example

The following example shows how to configure LFI using MLP over ATM on 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
 ppp authentication chap
 ppp chap hostname router2
 ppp chap password pass1
 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 ipv4:172.16.80.10
 dtmf-relay cisco-rtp
 ip precedence 5
!
dial-peer voice 1 pots
 destination-pattern 1333
 port 2/1/0
```

Additional References

The following sections provide references related to LFI for Frame Relay and ATM PVCs.

Related Documents

Related Topic	Document Title
Frame Relay traffic shaping	*Cisco IOS Quality of Service Solutions Configuration Guide*, Release 12.4 *Cisco IOS Quality of Service Solutions Command Reference*, Release 12.4
Frame Relay/ATM interworking	*Cisco IOS Wide-Area Networking Configuration Guide*, Release 12.4 *Cisco IOS Wide-Area Networking Command Reference*, Release 12.4
Frame Relay fragmentation	*Cisco IOS Wide-Area Networking Configuration Guide*, Release 12.4 *Cisco IOS Wide-Area Networking Command Reference*, Release 12.4

Standards

Standard	Title
FRF.8	*Frame Relay/ATM PVC Service Interworking Implementation Agreement*
FRF.12	*Frame Relay Fragmentation Implementation Agreement*

MIBs

MIB	MIBs Link
None	To locate and download MIBs for selected platforms, Cisco IOS releases, and feature sets, use Cisco MIB Locator found at the following URL: http://www.cisco.com/go/mibs

RFCs

RFC	Title
RFC 1990	*The PPP Multilink Protocol (MP)*

Technical Assistance

Description	Link
The Cisco Technical Support website contains thousands of pages of searchable technical content, including links to products, technologies, solutions, technical tips, and tools. Registered Cisco.com users can log in from this page to access even more content.	http://www.cisco.com/techsupport

Command Reference

This section documents a modified command only.

- **ppp multilink group**

ppp multilink group

To restrict a physical link to joining only a designated multilink-group interface, use the **ppp multilink group** command in interface configuration mode. To remove the restrictions, use the **no** form of this command.

ppp multilink group *group-number*

no ppp multilink group

Syntax Description

group-number	Multilink-group number (a nonzero number).

Defaults

Command is disabled.

Command Modes

Interface configuration

Command History

Release	Modification
12.0(3)T	This command was introduced as **multilink-group**.
12.2	This command was changed to **ppp multilink group**. The **multilink-group** command was accepted by the command line interpreter through Cisco IOS Release 12.2.
12.2(28)SB	This command was integrated into Cisco IOS Release 12.2(28)SB.

Usage Guidelines

By default this command is disabled, which means the link can negotiate to join any bundle in the system.

When the **ppp multilink group** command is configured, the physical link is restricted from joining any but the designated multilink-group interface. If a peer at the other end of the link tries to join a different bundle, the connection is severed. This restriction applies when Multilink PPP (MLP) is negotiated between the local end and the peer system. The link can still come up as a regular PPP interface.

This command is primarily used with the MLP inverse multiplexer described in the "Configuring Media-Independent PPP and Multilink PPP" chapter in the *Cisco IOS Dial Technologies Configuration Guide*.

Examples

The following example designates serial interface 1 as part of multilink bundle 1:

```
interface serial 1
 encapsulation ppp
 ppp multilink group 1
 ppp multilink
 ppp authentication chap
 pulse-time 3
```

Related Commands	Command	Description
	interface multilink	Creates a multilink bundle or enters multilink interface configuration mode.

Glossary

BACP—bandwidth allocation control protocol. Provides MLP peers with the ability to govern link utilization. After peers successfully negotiate BACP, they can use the bandwidth allocation protocol (BAP), which is a subset of BACP, to negotiate bandwidth allocation.

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

CHAP—Challenge Handshake Authentication Protocol. Security feature supported on lines using PPP encapsulation that prevents unauthorized access. CHAP does not itself prevent unauthorized access, but merely identifies the remote end. The router or access server then determines whether that user is allowed access.

DLCI—data-link connection identifier. Value that specifies a PVC or an SVC in a Frame Relay network. In the basic Frame Relay specification, DLCIs are locally significant (connected devices might use different values to specify the same connection). In the LMI extended specification, DLCIs are globally significant (DLCIs specify individual end devices).

FIFO—first-in, first-out. Refers to a buffering scheme where the first byte of data entering the buffer is the first byte retrieved by the CPU. In telephony, FIFO refers to a queuing scheme where the first calls received are the first calls processed.

FIFO queuing—first-in, first-out queuing. 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 on an interface in the order in which they arrive.

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

FRF.12—The *Frame Relay Fragmentation Implementation Agreement*.

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

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

PVC—permanent virtual circuit (or connection). Virtual circuit that is permanently established. PVCs save bandwidth associated with circuit establishment and teardown in situations where certain virtual circuits must exist all the time. In ATM terminology, this is called a permanent virtual connection.

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

RTP—Real-Time Transport Protocol. Provides end-to-end network transport functions suitable for applications that transmit real-time data (such as audio, video, or simulation data) over multicast or unicast network services.

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 or switched.

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.

WFQ—weighted fair queuing. 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 to stabilize network behavior during periods of congestion and results in increased performance and reduced retransmission.

Note See *Internetworking Terms and Acronyms* for terms not included in this glossary.