



Distributed Link Fragmentation and Interleaving for Frame Relay and ATM Interfaces on Cisco 7500 Series Routers

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

This document contains the following sections:

- [Feature Overview, page 1](#)
- [Supported Platforms, page 5](#)
- [Supported Standards, MIBs, and RFCs, page 5](#)
- [Prerequisites, page 6](#)
- [Configuration Tasks, page 7](#)
- [Monitoring and Maintaining LFI for Frame Relay and ATM, page 14](#)
- [Configuration Examples, page 15](#)
- [Command Reference, page 17](#)
- [Glossary, page 21](#)

Feature Overview

The Distributed Link Fragmentation and Interleaving (dLFI) for Frame Relay and ATM Interfaces on Cisco 7500 Series Routers feature extends link fragmentation and interleaving functionality to VIP-enabled Cisco 7500 series routers. The distributed packet processing architecture improves performance of the existing Link Fragmentation and Interleaving feature, which was already available on non-VIP-enabled Cisco 7500 series routers.



Note

The Distributed Link Fragmentation and Interleaving for Frame Relay and ATM Interfaces on Cisco 7500 Series Routers feature is often referred to as dLFI in this document.

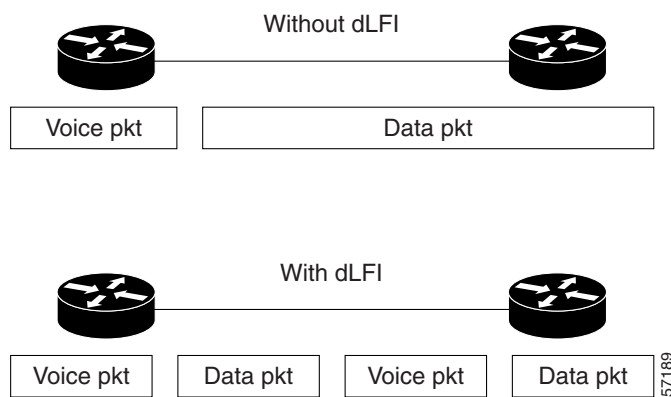
The dLFI feature supports the transport of real-time traffic, such as voice, and non-real-time traffic, such as data, on lower-speed Frame Relay and ATM virtual circuits (VCs) without causing excessive delay to the real-time traffic.

This feature is implemented using multilink PPP (MLP) over Frame Relay and ATM on VIP-enabled Cisco 7500 series routers. The feature enables delay-sensitive real-time packets and non-real-time packets to share the same link by fragmenting the large 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 packet reconstructed.

The dLFI feature is often useful in networks that send real-time traffic using Distributed Low Latency Queueing, such as voice traffic, but have bandwidth problems that delay this real-time traffic due to the transport of large, less time-sensitive data packets. The dLFI feature, which is required to properly use the dLFI feature, can be used in these networks to disassemble the large data packets into multiple segments. The real-time traffic packets then can be sent between these segments of the data packets. In this scenario, the real-time traffic does not experience a lengthy delay waiting for the low-priority data packets to traverse the network. The data packets are reassembled at the receiving side of the link, so the data is delivered intact.

The following figure illustrates how dLFI fragments a larger data packet to allow time-sensitive traffic, in this case voice traffic, to be delivered in a more timely manner.

Figure 1 *Distributed Link Fragmentation and Interleaving Example*



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 Distributed LFI feature works concurrently with and on the same switching path as other QoS features, ensuring high quality and scalable VoIP deployment.

The Distributed LFI feature works in conjunction with most of the Quality of Service (QoS) features available for the VIP-enabled Cisco 7500 series routers, including the following QoS features:

- Distributed Low Latency Queueing (dLLQ, the **priority** command)
- Distributed Traffic Shaping (dTS, the **shape** command)

- Distributed Compressed Real-Time Transport Protocol (dCRTP, the **ip [rtp | tcp] connections** and other compression commands)
- Distributed Class-Based Weighted Fair Queueing (dCBWFQ, the **bandwidth**, **fair-queue**, and **queue-limit** commands)
- Distributed Weighted Random Early Detection (dWRED, the **random-detect** command)
- Class-Based Marking (the **set** command)
- Traffic Policing (the **police** command)

Increased Call Support for Voice on the VIP-Enabled Cisco 7500

The Distributed LFI feature enhances the scalability on interfaces and PVCs, thereby allowing support for additional calls on the VIP-enabled Cisco 7500 series router.

Restrictions

The following restrictions apply to the Distributed Link Fragmentation and Interleaving feature:

Queuing Mechanisms Supported

- Many of the older queueing mechanisms are not supported by dLFI. These mechanisms include:
 - Fair-queueing on a virtual-template interface
 - Random-detect on a virtual-template interface
 - Custom queueing
 - Priority queueing



Note Fair queuing, random detection (dWRED), and priority queueing can be configured in a traffic policy using the Modular QoS CLI.

MLP Bundle Support

- 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.
QoS traffic policies will function properly in MLP bundles with more than one link, however.

VoIP Support

- Only Voice over IP is supported; Voice over Frame Relay and Voice over ATM are not supported by the dLFI feature.

QoS Policies Attached in an ATM Interface

- QoS policies can be attached to *either* subinterfaces *or* permanent virtual circuits (PVCs) with the ATM interface. As a general rule, the QoS policies must be attached the same way across the entire ATM interface. That is, all the QoS policies must be attached to either subinterfaces or PVCs, not a combination of subinterfaces and PVCs.



Note However, dLFI can be enabled on a PVC only. Therefore, if dLFI is enabled, dLFI requires that the QoS policy be attached at the PVC level for all the PVCs within the ATM interface. A QoS policy attached at the subinterface level is rejected.

QoS Policies Attached in a Frame Relay Network

- QoS policies can be attached to *either* a Frame Relay map-class *or* a subinterface within the Frame Relay network. As a general rule, the QoS policies must be attached the same way across the entire Frame Relay network. That is, all the QoS policies must be attached to either the Frame Relay map-class or the subinterface, not a combination of Frame Relay map-classes and subinterfaces.

**Note**

If FRF.12 is enabled, dLFI requires that the QoS policy be attached to the Frame Relay map-class. A QoS policy attached to the subinterface is rejected.

Related Features and Technologies

- Frame Relay/ATM interworking (FRF.8)
- Distributed Frame Relay fragmentation (FRF.12)
- The dLFI feature works in conjunction with most Quality of Service (QoS) features, including the QoS features listed below:
 - Distributed Low Latency Queueing (dLLQ, the **priority** command)
 - Distributed Traffic Shaping (dTS, the **shape** command)
 - Distributed Compressed Real-Time Transport Protocol (dCRTP)
 - Distributed Multilink Point-to-Point Protocol (dMLPPP)
 - Distributed Class-Based Weighted Fair Queueing (dCBWFQ, the **bandwidth**, **fair-queue**, and **queue-limit** commands)
 - Class-Based Marking (the **set** command)
 - Traffic Policing (the **police** command)

Related Documents

- *Distributed Low Latency Queueing* Cisco IOS Release 12.1(5)T feature module
- *Cisco IOS Quality of Service Solutions Configuration Guide*, Release 12.2
 - “Modular Quality of Service Command-Line Interface” section of the *Cisco IOS Quality of Service Solutions Configuration Guide*, Release 12.2
- *Cisco IOS Quality of Service Solutions Command Reference*, Release 12.2
- *Cisco IOS Wide-Area Networking Configuration Guide*, Release 12.2
- *Cisco IOS Wide-Area Networking Command Reference*, Release 12.2
- *Distributed Multilink Point-to-Point Protocol for Cisco 7500 Series Routers* Cisco IOS Release 12.0(3)T feature module
- *Distributed Traffic Shaping* Cisco IOS Release 12.1(5)T feature module
- *Distributed Compressed Real-Time Transport Protocol* Cisco IOS Release 12.1(5)T feature module
- *Distributed Class-Based Weighted Fair Queueing* Cisco IOS Release 12.1(5)T feature module

Supported Platforms

- Cisco 7500 series routers with a VIP2-50 or higher

**Note**

A VIP2-50 with a minimum of 64 MB DRAM and 4 MB SRAM or a VIP4-80 with a minimum of 128 MB SDRAM as program memory and 64 MB SDRAM as packet memory are required to run dLFI. The VIP4-80 will provide better performance in most networking environments.

The “[Prerequisites](#)” section of this document details other requirements.

Platform Support Through Feature Navigator

Cisco IOS software is packaged in feature sets that support specific platforms. To get updated information regarding platform support for this feature, access Feature Navigator. Feature Navigator dynamically updates the list of supported platforms as new platform support is added for the feature.

Feature Navigator is a web-based tool that enables you to quickly determine which Cisco IOS software images support a specific set of features and which features are supported in a specific Cisco IOS image.

To access Feature Navigator, you must have an account on Cisco.com. If you have forgotten or lost your account information, send a blank e-mail to cco-locksmith@cisco.com. An automatic check will verify that your e-mail address is registered with Cisco.com. If the check is successful, account details with a new random password will be e-mailed to you. Qualified users can establish an account on Cisco.com by following the directions at <http://www.cisco.com/register>.

Feature Navigator is updated when major Cisco IOS software releases and technology releases occur. As of May 2001, Feature Navigator supports M, T, E, S, and ST releases. You can access Feature Navigator at the following URL:

<http://www.cisco.com/go/fn>

Supported Standards, MIBs, and RFCs

Standards

No new or modified standards are supported.

MIBs

No new or modified MIBs are supported.

To obtain lists of supported MIBs by platform and Cisco IOS release, and to download MIB modules, go to the Cisco MIB website on Cisco.com at the following URL:

<http://www.cisco.com/public/sw-center/netmgmt/cmtk/mibs.shtml>

RFCs

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

Prerequisites

- The minimum required VIP for dLFI is a VIP2-50. A VIP2-50 with a minimum of 64 MB DRAM and 4 MB SRAM or a VIP4-80 with a minimum of 128 MB SDRAM as program memory and 64 MB SDRAM as packet memory is required to run dLFI. The VIP4-80 will provide better performance in most networking environments.
The amount of required DRAM and SRAM is proportional to the number of PVCs and interfaces; therefore, additional DRAM and SRAM might be necessary if additional PVCs or interfaces are added.

- RSP2 with a minimum of 64 MB of DRAM
- Distributed Low Latency Queueing (dLLQ). The interleaving of packets occurs only when a QoS traffic policy that contains a dLLQ configuration is attached to a PVC or an interface. If dLLQ is not configured on the PVC or interface, packets will be fragmented but not interleaved and the dLFI configuration will have no effect on network performance.

The **priority** policy map class command is used to configure dLLQ in a QoS traffic policy, and the **service-policy** interface command is used to attach the QoS traffic policy containing the **priority** configuration to an interface or a PVC.

- Distributed Cisco Express Forwarding (dCEF) must be globally enabled. You can enable dCEF using the **ip cef distributed** command in global configuration mode.
- A virtual template or a multilink interface has to be shutdown and then re-enabled (using the **shutdown** command followed by the **no shutdown** command) to change any PPP configuration. The exception to this restriction is the QoS traffic policy, which does not require the **shutdown/no shutdown** sequence in order to be enabled.
- All currently available serial port adapters for the Cisco 7500 series routers support dLFI using MLP over Frame Relay. These port adapters include:
 - PA-4T+
 - PA-8T
 - PA-CT3/4T1
 - PA-CE3
 - PA-MC-2E1/120
 - PA-MC-2T1
 - PA-MC-2T3+
 - PA-MC-4T1
 - PA-MC-8DSX1
 - PA-MC-8E1/120
 - PA-MC-8T1
 - PA-MC-E3

- MLP over ATM must use a PA-A3 ATM port adapter. Therefore, only PA-A3 ATM port adapters support dLFI using MLP over ATM.
 - PA-A3-E3
 - PA-A3-OC3
 - PA-A3-T3

**Note**

The PA-A3 IMA port adapter is not currently supported by dLFI in Cisco IOS Release 12.2 T.

Configuration Tasks

See the following sections for configuration tasks for the dLFI feature. Each task in the list is identified as optional or required.

- [Configuring LFI Using MLP over Frame Relay](#) (required for configuring dLFI on Frame Relay)
- [Configuring LFI Using MLP over ATM](#) (required for configuring dLFI on ATM)
- [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 Distributed Low Latency Queueing and Other QoS Features in a Traffic Policy](#)
- [Configuring LFI Using MLP in a Virtual Template Interface](#)
- [Associating the Virtual Template Interface with a Frame Relay PVC](#)

Configuring Distributed Low Latency Queueing and Other QoS Features in a Traffic Policy

The dLLQ feature must be enabled in order for the dLFI feature to interleave packet fragments. The dLLQ feature is configured in a QoS traffic policy which is attached to the multilink group. Other QoS features can also be configured in the traffic policy.

A traffic policy using dLLQ and other QoS features can be configured by entering the following commands:



	Command	Purpose
Step 1	Router(config)# class-map [match-any match-all] <i>class-map-name</i>	Specifies the user-defined name of the traffic class and enters class map configuration mode. A traffic class is used to classify traffic.
Step 2	Router(config-cmap)# match <i>match-criterion</i>	Specifies the criteria to classify traffic against. If traffic matches the specified match-criteria, traffic is said to belong to the traffic class. Multiple match criterion can be specified in a single traffic class.
Step 3	Router(config-cmap)# exit	Exits class map configuration mode.

	Command	Purpose
Step 4	Router(config)# policy-map <i>policy-name</i>	Specifies the name of the QoS traffic policy to configure and enters policy map configuration mode.
Step 5	Router(config-pmap)# class <i>class-map-name</i>	Specifies the name of a predefined class included in the service policy. This traffic class classifies traffic; the QoS features configured in the traffic policy determine how to forward traffic that matches the traffic class configuration. In these instructions, the <i>class-map-name</i> option should match the class-map-name entered in Step 1 of this procedure.
Step 6	Router(config-pmap-c)# priority [<i>percent</i>] [<i>kpbs</i> <i>percent</i>] [<i>bytes</i>]	Reserves a priority queue with a specified amount or percent of available bandwidth for high-priority traffic. The priority command is used to enable dLLQ.
Step 7	Router(config-pmap-c)#	Enables a QoS feature in the traffic policy.

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 <i>number</i>	Creates a virtual template and enters interface configuration mode.
Step 2	Router(config-if)# bandwidth <i>kilobits</i>	Sets the bandwidth value for an interface. The bandwidth value for the interface should match the traffic speed of the PVC; for instance, if the VBR peak cell rate is 128 kpbs, the <i>kilobits</i> option in the bandwidth command should be entered as 128. Similarly, if the PVC is being shaped to 64 kpbs, the <i>kilobits</i> option should be entered as 64.
Step 3	Router(config-if)# ip address <i>ip-address mask</i>	Sets a primary IP address for an interface.

Command	Purpose
Step 4 Router(config-if)# service-policy output <i>policy-name</i>	<p>(Required for traffic leaving the virtual template interface) Attaches a previously configured QoS traffic policy, which contains QoS classification and configuration parameters, that evaluates and applies QoS features for traffic <i>leaving</i> the interface with the virtual template.</p> <p>The priority command must be configured in this traffic policy for dLFI to operate properly. In this example, the <i>policy-name</i> option should match the <i>policy-name</i> option given in Step 4 of the Configuring Distributed Low Latency Queueing and Other QoS Features in a Traffic Policy procedure.</p> <hr/> <p> Note For dLFI, the QoS traffic policy that is attached using the service-policy command is attached to the virtual template. The QoS traffic policy does not have to be attached to the Frame Relay map class.</p>
Step 5 Router(config-if)# service-policy input <i>policy-name</i>	<p>(Required for traffic entering the virtual template interface) Attaches a previously configured QoS traffic policy, which contains QoS classification and configuration parameters, that evaluates and applies QoS features, including dLLQ, for traffic <i>entering</i> the interface with the virtual template.</p> <p>The priority command must be configured in this traffic policy for dLFI to operate properly. In this example, the <i>policy-name</i> option should match the <i>policy-name</i> option given in Step 4 of the Configuring Distributed Low Latency Queueing and Other QoS Features in a Traffic Policy procedure.</p> <hr/> <p> Note For dLFI, the QoS traffic policy that is attached using the service-policy command is attached on the virtual template. The QoS traffic policy does not have to be attached to the Frame Relay map class.</p>
Step 6 Router(config-if)# ppp multilink	Enables MLP on the interface.

	Command	Purpose
Step 7	Router(config-if)# ppp multilink fragment-delay <i>milliseconds</i>	Configures the maximum delay allowed for transmission of a packet fragment on an MLP bundle.
Step 8	Router(config-if)# ppp multilink interleave	Enables interleaving of packets among the fragments of larger packets on an MLP bundle.

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

$$\text{fragment size} = \text{bandwidth} \times \text{fragment-delay} / 8$$

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 <i>type number</i>	Configures an interface type and enters interface configuration mode.
Step 2	Router(config-if)# frame-relay interface-dlci <i>dlci</i> [ppp <i>virtual-template-name</i>]	Associates a virtual template interface with a Frame Relay DLCI. ¹

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

- [Configuring Distributed Low Latency Queueing and Other QoS Features in a Traffic Policy](#)
- [Configuring LFI Using MLP on a Virtual Template Interface](#)
- [Associating the Virtual Template Interface with an ATM PVC](#)

Configuring Distributed Low Latency Queueing and Other QoS Features in a Traffic Policy


The dLLQ feature must be enabled in order for the dLFI feature to interleave packet fragments. The dLLQ feature is configured in a QoS traffic policy which is attached to the multilink group. Other QoS features can also be configured in the traffic policy.


A traffic policy using dLLQ and other QoS features can be configured by entering the following commands:

	Command	Purpose
Step 3	Router(config)# class-map [match-any match-all] <i>class-map-name</i>	Specifies the user-defined name of the traffic class and enters class map configuration mode. A traffic class is used to classify traffic.
Step 4	Router(config-cmap)# match <i>match-criterion</i>	Specifies the criteria to classify traffic against. If traffic matches the specified match-criteria, traffic is said to belong to the traffic class. Multiple match criterion can be specified in a single traffic class.
Step 5	Router(config-cmap)# exit	Exits class map configuration mode.
Step 6	Router(config)# policy-map <i>policy-name</i>	Specifies the name of the QoS traffic policy to configure and enters policy map configuration mode.
Step 7	Router(config-pmap)# class <i>class-map-name</i>	Specifies the name of a predefined class included in the service policy. This traffic class classifies traffic; the QoS features configured in the traffic policy determine how to forward traffic that matches the traffic class configuration. In these instructions, the <i>class-map-name</i> option should match the <i>class-map-name</i> entered in Step 1 of this procedure.
Step 8	Router(config-pmap-c)# priority [percent] [<i>kpbs</i> <i>percent</i>] [<i>bytes</i>]	Reserves a priority queue with a specified amount of available bandwidth for high-priority traffic. The priority command is used to enable dLLQ.
Step 9	Router(config-pmap-c)#	Enables a QoS feature in the traffic policy.

Configuring LFI Using MLP on a Virtual Template Interface

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

	Command	Purpose
Step 1	Router(config)# interface virtual-template <i>number</i>	Creates a virtual template and enters interface configuration mode.
Step 2	Router(config-if)# bandwidth <i>kilobits</i>	Sets the bandwidth value for an interface.
Step 3	Router(config-if)# ip address <i>ip-address mask</i>	Sets a primary IP address for an interface.
Step 4	Router(config-if)# service-policy output <i>policy-name</i>	<p>(Required for traffic leaving the virtual template interface) Attaches a previously configured QoS traffic policy, which contains QoS classification and configuration parameters, that evaluates and applies QoS features, including dLLQ, for traffic leaving the interface with the virtual template.</p> <p>The priority command must be configured in this traffic policy for dLFI to operate properly. In this example, the <i>policy-name</i> option should match the <i>policy-name</i> option given in Step 4 of the Configuring Distributed Low Latency Queueing and Other QoS Features in a Traffic Policy procedure.</p> <p> Note For dLFI, the QoS traffic policy that is attached using the service-policy command is attached to the virtual template. The QoS traffic policy does not have to be attached to the ATM PVC.</p>

	Command	Purpose
Step 5	Router(config-if)# service-policy input <i>policy-name</i>	<p>(Required for traffic entering the virtual template interface) Attaches a previously configured QoS traffic policy, which contains QoS classification and configuration parameters, that evaluates and applies QoS features, including dLLQ, for traffic entering the interface with the virtual template.</p> <p>The priority command must be configured in this traffic policy for dLFI to operate properly. In this example, the <i>policy-name</i> option should match the <i>policy-name</i> option given in Step 4 of the Configuring Distributed Low Latency Queueing and Other QoS Features in a Traffic Policy procedure.</p> <p> Note For dLFI, the QoS traffic policy that is attached using the service-policy command is entered on the virtual template. The QoS traffic policy does not have to be attached to the ATM PVC</p>
Step 6	Router(config-if)# ppp multilink	Enables MLP on the interface.
Step 7	Router(config-if)# ppp multilink fragment-delay <i>milliseconds</i>	Configures the maximum delay allowed for transmission of a packet fragment on an MLP bundle.
Step 8	Router(config-if)# ppp multilink interleave	Enables interleaving of packets among the fragments of larger packets on an MLP bundle.

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

$$\text{fragment size} = \text{bandwidth} \times \text{fragment-delay} / 8$$

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:

$$\text{fragment size} = 48 \times \text{number of cells} - 10$$

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 <i>slot/0</i> OR Router(config)# interface atm <i>slot/port</i>	Specifies the ATM interface type and enters interface configuration mode. ¹
Step 2	Router(config-if)# pvc [<i>name</i>] <i>vpi/vci</i>	Creates an ATM PVC.

	Command	Purpose
Step 3	Router(config-if-atm-vc)# abr <i>output-pcr output-mcr</i>	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 <i>number</i>	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 router documentation.
2. ABR = available bit rate

Verifying LFI for Frame Relay and ATM

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

Command	Purpose
Router# show frame-relay pvc <i>dlci</i>	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.
Router# show policy-map interface	Displays configurations and statistics of all input and output traffic policies attached to an interface.

Monitoring and Maintaining LFI for Frame Relay and ATM

To monitor LFI for Frame Relay and ATM using MLP, 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 Configuration Example](#)
- [LFI over ATM Configuration Example](#)

LFI over Frame Relay Configuration Example

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

```
class-map voip
  match ip precedence 5

class-map business
  match ip precedence 3

policy-map llq-policy
  class voip
    priority 32
  class business
    bandwidth 32

policy-map shape-llq-policy
  class class-default
    shape average 80000 320 320
    service-policy llq-policy

policy-map input-policy
  class voip
    police 32000 1500 1500 conform-action transmit exceed-action drop

controller T1 5/1/0
  framing esf
  linecode b8zs
  channel-group 0 timeslots 1-2

interface Serial5/1/0:0
  no ip address
  encapsulation frame-relay

interface Serial5/1/0:0.1 point-to-point
  frame-relay interface-dlci 20 ppp Virtual-Template2

interface Virtual-Template2
  bandwidth 78
  ip address 98.0.0.2 255.0.0.0
  no keepalive
  service-policy output llq-policy
  service-policy input input-policy
  ppp multilink
  ppp multilink fragment-delay 8
  ppp multilink interleave
```

LFI over ATM Configuration Example

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

```
class-map voip
  match ip precedence 5

class-map business
  match ip precedence 3

policy-map llq-policy
  class voip
    priority 32
  class business
    bandwidth 32

policy-map input-policy
  class voip
    police 32000 1500 1500 conform-action transmit exceed-action drop

interface ATM4/0/0
  no ip address
  no atm ilmi-keepalive

interface ATM4/0/0.1 point-to-point
  pvc 0/34
  abr 100 80
  protocol ppp Virtual-Template4

interface Virtual-Template4
  bandwidth 78
  ip address 88.0.0.2 255.0.0.0
  service-policy output llq-policy
  service-policy input input-policy
  ppp multilink
  ppp multilink fragment-delay 8
  ppp multilink interleave
```

Command Reference

This section documents the modified **ppp multilink interleave** command. All other commands used with this feature are not new or modified and are documented in the Cisco IOS Release 12.2 command reference publications.

**Note**

The command reference for **ppp multilink interleave** is written for all cases of the **ppp multilink interleave** command; the command reference is not specific to the VIP-enabled Cisco 7500 series router but does include information for this platform.

ppp multilink interleave

To enable interleaving of packets among the fragments of larger packets on a Multilink PPP (MLP) bundle, use the **ppp multilink interleave** interface configuration command. To disable interleaving, use the **no** form of this command.

ppp multilink interleave

no ppp multilink interleave

Syntax Description This command has no arguments or keywords.

Defaults Interleaving is disabled by default.

Command Modes Interface configuration

Command History

Release	Modification
11.3	This command was introduced.
12.2(4)T3	This command was introduced on the VIP-enabled Cisco 7500 series routers as part of the Distributed Link Fragmentation and Interleaving for Frame Relay and ATM Interfaces on Cisco 7500 Series Routers feature.

Usage Guidelines

On the VIP-enabled Cisco 7500 series routers, distributed Cisco Express Forwarding (dCEF) and distributed Low Latency Queuing (dLLQ) must be enabled in order to use the **ppp multilink interleave** command.

If interleaving is enabled when fragment-delay is not configured, then the default fragment size is 78 bytes of payload.

On every platform except the VIP-enabled Cisco 7500 series router, interleaving works only when the queueing mode on the bundle has been set to Fair Queueing.

The **ppp multilink interleave** command applies only to interfaces that can configure a bundle interface, such as virtual templates, dialer interfaces, and ISDN BRI, or PRI interfaces. This command can only be configured using virtual template interfaces on VIP-enabled Cisco 7500 series routers for ATM and Frame Relay.

Examples**LFI over Frame Relay Configuration Example**

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

```
class-map voip
  match ip precedence 5

class-map business
  match ip precedence 3

policy-map llq-policy
  class voip
    priority 32
  class business
    bandwidth 32

policy-map shape-llq-policy
  class class-default
    shape average 80000 320 320
    service-policy llq-policy

policy-map input-policy
  class voip
    police 32000 1500 1500 conform-action transmit exceed-action drop

controller T1 5/1/0
  framing esf
  linecode b8zs
  channel-group 0 timeslots 1-2

interface Serial5/1/0:0
  no ip address
  encapsulation frame-relay

interface Serial5/1/0:0.1 point-to-point
  frame-relay interface-dlci 20 ppp Virtual-Template2

interface Virtual-Template2
  bandwidth 78
  ip address 98.0.0.2 255.0.0.0
  no keepalive
  service-policy output llq-policy
  service-policy input input-policy
  ppp multilink
  ppp multilink fragment-delay 8
  ppp multilink interleave
```

LFI over ATM Configuration Example

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

```

class-map voip
  match ip precedence 5

class-map business
  match ip precedence 3

policy-map llq-policy
  class voip
    priority 32
  class business
    bandwidth 32

policy-map input-policy
  class voip
    police 32000 1500 1500 conform-action transmit exceed-action drop

interface ATM4/0/0
  no ip address
  no atm ilmi-keepalive

interface ATM4/0/0.1 point-to-point
  pvc 0/34
  abr 100 80
  protocol ppp Virtual-Template4

interface Virtual-Template4
  bandwidth 78
  ip address 88.0.0.2 255.0.0.0
  service-policy output llq-policy
  service-policy input input-policy
  ppp multilink
  ppp multilink fragment-delay 8
  ppp multilink interleave

```

Related Commands

Command	Description
show ppp multilink	Displays bundle information for the MLP bundles and their PPP links in the router.
ppp multilink fragment delay	Specifies a maximum size, in units of time, for packet fragments on an MLP bundle.

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.

