Distributed Link Fragmentation and Interleaving over Leased Lines

Table 1 Feature History

Cisco IOS Release	Modification
12.2(4)T3	The Distributed Link Fragmentation and Interleaving feature was introduced. This version of the feature supported Distributed Link Fragmentation and Interleaving for ATM and Frame Relay.
12.2(8)T	The Distributed Link Fragmentation and Interleaving over Leased Lines feature was introduced. This feature introduced Distributed Link Fragmentation and Interleaving on leased lines.
12.0(24)S	The Distributed Link Fragmentation and Interleaving over Leased Lines feature was introduced on Cisco IOS Release 12.0 S. This feature introduced Distributed Link Fragmentation and Interleaving on leased lines. Distributed Link Fragmentation and Interleaving is not available for ATM and Frame Relay using Cisco IOS Release 12.0 S.
12.2(14)SX	The Distributed Link Fragmentation and Interleaving feature was introduced on the Catalyst 6500 series switch and Cisco 7600 series router FlexWAN module.

This document describes the Distributed Link Fragmentation and Interleaving over Leased Lines feature. It includes information such as the benefits of the new feature, related documents, and supported platforms.

This document contains the following sections:

Feature Overview

The Distributed Link Fragmentation and Interleaving over Leased Lines feature extends distributed link fragmentation and interleaving functionality to leased lines. Previously, Distributed Link Fragmentation and Interleaving was only available for Frame Relay and ATM on Cisco IOS Release 12.2 T. Distributed Link Fragmentation and Interleaving is not available for Frame Relay and ATM using Cisco IOS Release 12.0 S.

Note Distributed Link Fragmentation and Interleaving for Frame Relay, ATM, and Leased Lines is often referred to as dLFI in this document. This document covers the procedures for configuring dLFI on Frame Relay and ATM in addition to covering the procedure for configuring dLFI on a leased line.

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) and on leased lines without causing excessive delay to the real-time traffic.

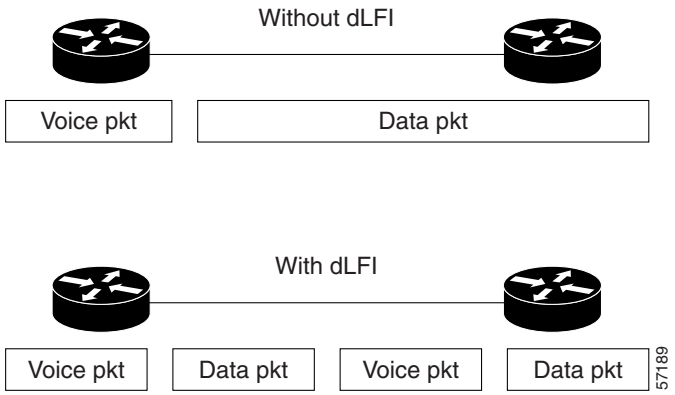
This feature is implemented using multilink PPP (MLP) over Frame Relay, ATM, and leased lines on. The feature enables delay-sensitive real-time packets and non-real-time packets to share the same link by fragmenting the large data packets into a sequence of smaller data packets (fragments). The fragments 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, but have bandwidth problems that delay this real-time traffic due to the transport of large, less time-sensitive data packets. 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 ability to configure Quality of Service (QoS) using the Modular Quality of Service (QoS) Command-Line Interface (CLI) (MQC) while also using distributed MLP (dMLP) is also introduced as part of the dLFI feature. The ability to configure QoS using the MQC while using dMLP was not supported prior to the introduction of the dLFI feature.

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) 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 and on leased lines.

Interoperability with Other QoS Features

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

The dLFI feature works in conjunction with most of 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 VIP-Enabled Cisco 7500 Series Routers

The dLFI feature enhances the scalability on interfaces and Permanent Virtual Circuits (PVCs), thereby allowing support for additional calls on VIP-enabled Cisco 7500 series routers.

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

CRTP Configurations

- Compressed Real-Time Transport Protocol (CRTP) configurations should not be configured on the multilink interface when the following feature combination is configured:
  - Link Fragmentation and Interleaving (LFI) is enabled on the multilink interface
  - The multilink bundle has more than one member link
  - QoS policy with a priority feature is enabled on the multilink interface

Note In a dMLP/dLFI configuration, priority packets do not carry MLP header and sequence number. Thus, MLP distributes the priority packets across all member links. As a result, packets that are compressed by CRTP may arrive out-of-order at the receiving router. This prohibits CRTP from decompressing the packet header and forces CRTP to drop the packets.

MLP Bundle Support

- Only one link per MLP bundle is supported when using dLFI over Frame Relay or dLFI over ATM. If more than one link is used in an MLP bundle when using dLFI over Frame Relay or dLFI over ATM, dLFI is automatically disabled. When using dLFI over leased lines, more than one link can be configured with dLFI in the MLP bundle.

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

VoIP Support

- Only Voice over IP (VoIP) 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)
- Distributed Multilink Point-to-Point Protocol (dMLP)
- 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, the **ip** [**rtp** | **tcp**] **connections** and other compression commands)
  - 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 7600 series routers and Catalyst 6500 series switches with a FlexWAN
- 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 on the Cisco 7500 series routers. The VIP4-80 will provide better performance in most networking environments.

The "Prerequisites" section of this document details other requirements, including port adapter requirements.

Determining Platform Support Through Cisco Feature Navigator

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

Cisco Feature Navigator is a web-based tool that enables you to determine which Cisco IOS software images support a specific set of features and which features are supported in a specific Cisco IOS image. You can search by feature or release. Under the release section, you can compare releases side by side to display both the features unique to each software release and the features in common.

To access Cisco 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 found at this URL:

http://www.cisco.com/register

Cisco Feature Navigator is updated regularly when major Cisco IOS software releases and technology releases occur. For the most current information, go to the Cisco Feature Navigator home page at the following URL:

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

Supported Standards, MIBs, and RFCs

Standards

No new or modified standards are supported by this feature.

MIBs

No new or modified MIBs are supported by this feature.

To locate and download MIBs for selected platforms, Cisco IOS releases, and feature sets, 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 following prerequisites apply for support of dLFI on the Cisco 7500 series routers and on the Cisco 7600 series routers and Catalyst 6500 series switches with a FlexWAN:

- Cisco 7500 Series Routers Prerequisites, page 7
- Cisco 7600 Series Router Prerequisites, page 9

Cisco 7500 Series Routers 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 and VIP CPU is available. If VIP CPU has reached its maximum threshold, additional DRAM and SRAM will not be useful for adding additional PVCs or interfaces.
- 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.

  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 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 must 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-MC-T3
  - 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
- All currently available channelized serial port adapters for Cisco 7500 series routers support dLFI over leased lines. These port adapters include:
  - 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
  - PA-MC-STM1
- 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 supported by dLFI.

Cisco 7600 Series Router Prerequisites

The following prerequisites apply to dLFI support on the Cisco 7600 series routers and Catalyst 6500 series switches with a FlexWAN:

- 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.

  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 to an interface or a PVC.
- A virtual template or a multilink interface must 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 FlexWAN support LFI using MLP over Frame Relay:
  - PA-4T+
  - PA-8T
  - PA-MC-T3
  - PA-MC-2T3+
  - PA-MC-4T1
  - PA-MC-8E1/120
  - PA-MC-8T1
  - PA-MC-E3
- All currently available channelized serial port adapters for the FlexWAN support LFI over leased lines:
  - PA-MC-2T3+
  - PA-MC-4T1
  - PA-MC-8E1/120
  - PA-MC-8T1
  - PA-MC-E3
  - PA-MC-STM1
- MLP over ATM must use a PA-A3 ATM port adapter. The following PA-A3 ATM port adapters support LFI using MLP over ATM:
  - PA-A3-E3
  - PA-A3-OC3
  - PA-A3-T3

Note The PA-A3 IMA port adapter is not supported by dLFI.

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. Not available on Cisco IOS Release 12.0 S)
- Configuring LFI Using MLP over ATM (required for configuring dLFI on ATM. Not available on Cisco IOS Release 12.0 S)
- Configuring LFI Using MLP over a Leased Line (required for configuring dLFI on a leased lines)
- Verifying LFI for Frame Relay, ATM, or Leased Lines (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 on 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] class-map-name`	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 match-criterion`	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.
Step 4	`Router(config)# policy-map policy-name`	Specifies the name of the QoS traffic policy to configure and enters policy map configuration mode.

	Command	Purpose
Step 5	`Router(config-pmap)# **class** *class-map-name*`	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 *class-map-name* option should match the *class-map-name* entered in Step 1 of this procedure.
Step 6	`Router(config-pmap-c)# **priority** [**percent**] [*kpbs* | *percent*] [*bytes*]`	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 on a Virtual Template Interface

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

	Command	Purpose
Step 1	`Router(config)# **interface virtual-template** *number*`	Creates a virtual template and enters interface configuration mode.
Step 2	`Router(config-if)# **bandwidth** *kilobits*`	Sets the bandwidth value for an interface. 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 *kilobits* option in the **bandwidth** command should be entered as 128. Similarly, if the PVC is being shaped to 64 kpbs, the *kilobits* option should be entered as 64.
Step 3	`Router(config-if)# **ip address** *ip-address mask*`	Sets a primary IP address for an interface.

	Command	Purpose
Step 4	`Router(config-if)# service-policy output policy-name`	(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 *leaving* the interface with the virtual template. The **priority** command must be configured in this traffic policy for dLFI to operate properly. In this example, the *policy-name* option should match the *policy-name* option given in Step 4 of the Configuring Distributed Low Latency Queueing and Other QoS Features in a Traffic Policy procedure. **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.
Step 5	`Router(config-if)# service-policy input policy-name`	(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. The **priority** command must be configured in this traffic policy for dLFI to operate properly. In this example, the *policy-name* option should match the *policy-name* option given in Step 4 of the Configuring Distributed Low Latency Queueing and Other QoS Features in a Traffic Policy procedure. **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.
Step 6	`Router(config-if)# ppp multilink`	Enables MLP on the interface.

	Command	Purpose
Step 7	`Router(config-if)# ppp multilink fragment-delay` *milliseconds*	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:

fragment size = bandwidth x fragment-delay / 8

Associating the Virtual Template Interface with a Frame Relay PVC

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

	Command	Purpose
Step 1	`Router(config)# interface` *type number*	Configures an interface type and enters interface configuration mode.
Step 2	`Router(config-if)# frame-relay interface-dlci` *dlci* [`ppp` *virtual-template-name*]	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 in 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 1	`Router(config)# class-map [match-any | match-all] class-map-name`	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 match-criterion`	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.
Step 4	`Router(config)# policy-map policy-name`	Specifies the name of the QoS traffic policy to configure and enters policy map configuration mode.
Step 5	`Router(config-pmap)# class class-map-name`	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 *class-map-name* option should match the *class-map-name* entered in Step 1 of this procedure.
Step 6	`Router(config-pmap-c)# priority [percent] [kpbs | percent] [bytes]`	Reserves a priority queue with a specified amount or percentage 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 dLFI using MLP on a virtual template interface, use the following interface configuration commands:

	Command	Purpose
Step 1	`Router(config)# interface virtual-template number`	Creates a virtual template and enters interface configuration mode.
Step 2	`Router(config-if)# bandwidth kilobits`	Sets the bandwidth value for an interface.
Step 3	`Router(config-if)# ip address ip-address mask`	Sets a primary IP address for an interface.

	Command	Purpose
Step 4	`Router(config-if)# service-policy output policy-name`	(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. The **priority** command must be configured in this traffic policy for dLFI to operate properly. In this example, the *policy-name* option should match the *policy-name* option given in Step 4 of the Configuring Distributed Low Latency Queueing and Other QoS Features in a Traffic Policy procedure. **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.
Step 5	`Router(config-if)# service-policy input policy-name`	(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. The **priority** command must be configured in this traffic policy for dLFI to operate properly. In this example, the *policy-name* option should match the *policy-name* option given in Step 4 of the Configuring Distributed Low Latency Queueing and Other QoS Features in a Traffic Policy procedure. **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.
Step 6	`Router(config-if)# ppp multilink`	Enables MLP on the interface.

	Command	Purpose
Step 7	`Router(config-if)# ppp multilink fragment-delay` *milliseconds*	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:

fragment size = bandwidth x 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:

fragment size = 48 x 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` *slot*`/0` or `Router(config)# interface atm` *slot*`/`*port*	Specifies the ATM interface type and enters interface configuration mode.[1]
Step 2	`Router(config-if)# pvc` [*name*] *vpi*`/`*vci*	Creates an ATM PVC.
Step 3	`Router(config-if-atm-vc)# abr` *output-pcr output-mcr*	Selects ABR[2] QoS and configures the output peak cell rate and output minimum guaranteed cell rate for an ATM PVC.
Step 4	`Router(config-if-atm-vc)# protocol ppp virtual-template` *number*	Specifies that PPP is established over the ATM PVC using the configuration from the specified virtual template.

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

Configuring LFI Using MLP over a Leased Line

LFI over a leased line can be configured using MLP. To configure LFI over a leased line, perform the tasks in the following sections:

- Configuring Distributed Low Latency Queueing and Other QoS Features in a Traffic Policy
- Assigning an Interface to a Multilink Group
- Configuring the Channel Group
- Creating a Multilink Group
- Assigning an Interface to a Multilink Group

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] class-map-name`	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 match-criterion`	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.
Step 4	`Router(config)# policy-map policy-name`	Specifies the name of the QoS traffic policy to configure and enters policy map configuration mode.
Step 5	`Router(config-pmap)# class class-map-name`	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 *class-map-name* option should match the *class-map-name* entered in Step 1 of this procedure.
Step 6	`Router(config-pmap-c)# priority [percent] [kpbs | percent] [bytes]`	Reserves a priority queue with a specified amount or percentage 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.

Note The **bandwidth** command can be used in a QoS traffic policy to specify an amount of bandwidth to be reserved for the traffic policy. If the **bandwidth** command is used in a traffic policy that will be attached to a multilink interface, the following guidelines should be followed:
1. Use **bandwidth percent** in favor of **bandwidth** *kpbs* if possible. If the **bandwidth** *kpbs* option is specified as member links join and leave the bundle, the bandwidth setting will not adjust to the new aggregate bandwidth and the QoS traffic policy will either consume more bandwidth than desired or not have enough available bandwidth. Because the **bandwidth percent** option adjusts accordingly when new members links are added or removed, the amount of available bandwidth is properly adjusted when new member links are added or removed.

2. If **bandwidth** *kpbs* must be used, specify a bandwidth statement for the multilink group to reflect the expected available bandwidth for the multilink group. This bandwidth should be identical to the amount of bandwidth specified in the channel configuration when the **channel-group** command is entered (See Step 2 in the "Configuring the Channel Group" section of this document). For instance, if two channels are defined using the DS0 rate (64 kpbs), the *kilobits* variable should be entered as 128.

Configuring the Channel Group

A channel group is used to configure the controllers. To configure the controller, enter the following commands:

	Command	Purpose
Step 1	`Router(config)# controller [t1 | e1] slot/port`	Configures a T1 or E1 controller.
Step 2	`Router(config-controller)# channel-group channel-number timeslots range [speed {48 | 56 | 64}]`	Defines the time slots that belong to each T1 or E1 circuit.

Creating a Multilink Group

To create a multilink group, use the following commands beginning in interface configuration mode:

	Command	Purpose
Step 1	`Router(config)# interface multilink group-number`	Creates and names a multilink bundle. The name of the multilink bundle is the *group-number*.
Step 2	`Router(config-if)# ip address ip-address mask`	Assigns an IP address for the multilink group.
Step 3	`Router(config-if)# bandwidth kilobits`	(Optional, unless a QoS traffic policy using the **bandwidth** *kpbs* command will be attached to the multilink group) Sets the bandwidth value for an interface. The bandwidth should match the parameters defined in channel configuration. For instance, if two channels are defined using the DS0 rate (64 kpbs), the *kilobits* variable should be entered as 128.
Step 4	`Router(config-if)# ppp multilink`	Enables MLP for the multilink group.
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 packets among the fragments of larger packets on an MLP bundle.

	Command	Purpose
Step 7	`Router(config-if)# service-policy output policy-name`	(Required for traffic leaving the multilink group) 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 bundle. The **priority** command must be configured in this traffic policy for dLFI to operate properly. In this example, the *policy-name* option should match the *policy-name* option given in Step 4 of the Configuring Distributed Low Latency Queueing and Other QoS Features in a Traffic Policy procedure. **Note** For dLFI, the QoS traffic policy that is attached using the **service-policy** command is entered in the multilink group. The QoS traffic policy does not have to be attached to the serial interface that is part of the group.
Step 8	`Router(config-if)# service-policy input policy-name`	(Required for traffic entering the multilink group) 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 bundle. The **priority** command must be configured in this traffic policy for dLFI to operate properly. In this example, the *policy-name* option should match the *policy-name* option given in Step 4 of the Configuring Distributed Low Latency Queueing and Other QoS Features in a Traffic Policy procedure. **Note** For dLFI, the QoS traffic policy that is attached using the **service-policy** command is entered in the multilink group. The QoS traffic policy does not have to be attached to the serial interface that is part of the group.

Assigning an Interface to a Multilink Group

To configure an interface and attach the interface to a multilink group, use the following commands beginning in interface configuration mode:

	Command	Purpose
Step 1	`Router(config)# interface serial interface-number`	Specifies the serial interface to configure. Only serial interfaces can be bundled using multilink groups.
Step 2	`Router(config-if)# no ip address`	Removes any specified IP address.
Step 3	`Router(config-if)# keepalive [seconds]`	Sets the keepalive interval for the interface. The keepalive interval, which is the frequency at which the Cisco IOS software sends messages to itself or to the other end, is used to ensure a network interface is up. The *seconds* variable determines how often these messages are sent; for instance, if **keepalive 5** is entered, a keepalive message is sent every 5 seconds.
Step 4	`Router(config-if)# ppp chap hostname hostname`	Specifies the hostname for the interface when Challenge Handshake Authentication Protocol (CHAP) is used for authentication. The CHAP hostname must be configured in order to avoid potential errors when more than one multilink group exists between the same two routers. A different hostname should be specified for each multilink group on a router.
Step 5	`Router(config-if)# ppp multilink`	Enables multilink PPP for the interface.
Step 6	`Router(config-if)# multilink-group group-number`	Assigns the interface to a multilink group. To assign the interface to a previously configured multilink group, the *group-number* variable in this step must match the *group-number* variable specified in the multilink group (in the "Creating a Multilink Group" section of this document, the *group-number* for the multilink group is specified in Step 1).

Verifying LFI for Frame Relay, ATM, or Leased Lines

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

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

To monitor LFI for Frame Relay, ATM, or leased lines using MLP, use the following privileged EXEC commands:

Command	Purpose
`Router# show ppp multilink`	Displays bundle information for the MLP bundles and their PPP links in the router. Displays dLFI statistics, including the number of fragmented, unfragmented, and reassembled packets, reassembly and fragmentation drops, and fragments that arrived out of sequence.
`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 Leased Line Configuration Example
- Monitoring LFI 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
```

LFI over Leased Line Configuration Example

The following example shows the configuration of LFI over a leased line. LFI must use an MLP bundle to be used over a leased line.

```
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

controller T1 5/1/0
 channel group 0 timeslots 1-2

interface multilink 2
 ip address 172.16.0.0 255.0.0.0
 keepalive 5
 bandwidth 128
 ppp multilink
 ppp multilink fragment-delay 8
 ppp multilink interleave
 service-policy output llq-policy
 service-policy input input-policy
 multilink-group 2


interface serial5/0/0:0
 no ip address
 encapsulation ppp
 keepalive 5
 ppp chap hostname G2
 ppp multilink
 multilink-group 2
```

Monitoring LFI Example

In the following example, the **show ppp multilink** command is used to monitor dLFI traffic. Note that this command output provides the numbers of fragmented, unfragmented, and reassembled packets entering and leaving the bundle.

```
Router# show ppp multilink
Multilink11, bundle name is G11
  Bundle is Distributed
  0 lost fragments, 0 reordered, 0 unassigned
  0 discarded, 0 lost received, 1/255 load
  0x0 received sequence, 0x14 sent sequence
  Member links:2 active, 0 inactive (max not set, min not set)
    Serial4/1/1:2, no frags rcvd  64 weight, 2 max fragments
    Serial4/1/1:3, no frags rcvd  64 weight, 2 max fragments

dLFI statistics:
            DLFI Packets    Pkts In   Chars In   Pkts Out  Chars Out
              Fragmented      20        1372        20       1372
            UnFragmented       0          0          0          0
             Reassembled       2         1228        2        1228

       Reassembly Drops        0
    Fragmentation Drops        0
       Out of Seq Frags        0
```

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.

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 feature. The Distributed Link Fragmentation and Interleaving feature introduced this command for ATM and Frame Relay only.
12.2(8)T	This command was introduced for leased lines for VIP-enabled Cisco 7500 series routers.
12.0(24)S	This command was introduced for leased lines for VIP-enabled Cisco 7500 series routers running Cisco IOS Release 12.0 S. This command cannot be used for ATM and Frame Relay using Cisco IOS Release 12.0 S.
12.2(14)SX	This command was introduced for leased lines for Cisco 7600 series routers and Catalyst 6500 series switches with a FlexWAN.

Usage Guidelines

On the VIP-enabled Cisco 7500 series routers, distributed Cisco Express Forwarding (dCEF) must be enabled in order to use the **ppp multilink interleave** command. On the Cisco 7600series routers and Catalyst 6500 series routers, dCEF is enabled by default.

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

This command can only be configured using virtual template interfaces on VIP-enabled Cisco 7500 series routers for ATM and Frame Relay.

Examples

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
```

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
```

The following example shows the configuration of LFI over a leased line. LFI must use an MLP bundle to be used over a leased line.

```
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

controller T1 5/1/0
 channel group 0 timeslots 1-2

interface multilink 2
 ip address 172.16.0.0 255.0.0.0
 keepalive 5
 bandwidth 128
 ppp multilink
 ppp multilink fragment-delay 8
 ppp multilink interleave
 service-policy output llq-policy
 service-policy input input-policy
 multilink-group 2


interface serial5/0/0:0
 no ip address
 encapsulation ppp
 keepalive 5
 ppp chap hostname G2
 ppp multilink
 multilink-group 2
```

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.