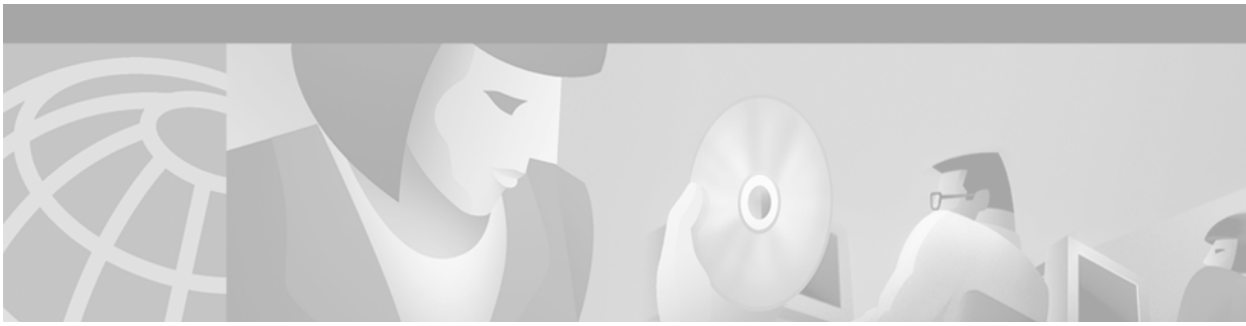


Link Efficiency Mechanisms Overview

Cisco IOS software offers five link-layer efficiency mechanisms or features—Link Fragmentation and Interleaving (LFI) for Multilink PPP (MLP), Link Fragmentation and Interleaving for Frame Relay and ATM VCs, Frame Relay Fragmentation, Compressed Real-Time Protocol (CRTP), and distributed Compressed Real-Time Protocol (dCRTP)—that work with queueing and traffic shaping to improve the efficiency and predictability of the application service levels.

This chapter gives a brief introduction to these link-layer efficiency mechanisms described in the following sections:

Link Fragmentation and Interleaving for MLP

Interactive traffic such as Telnet and Voice over IP (VoIP) is susceptible to increased latency when the network processes large packets such as LAN-to-LAN FTP transfers traversing a WAN. Packet delay is especially significant when the FTP packets are queued on slower links within the WAN. To solve delay problems on slow bandwidth links, a method for fragmenting larger packets and then queueing the smaller packets between fragments of the large packets is required.

The Cisco IOS LFI feature reduces delay on slower-speed links by breaking up large datagrams and interleaving low-delay traffic packets with the smaller packets resulting from the fragmented datagram. The Cisco IOS LFI feature uses the Cisco implementation of MLP, which supports the fragmentation and packet sequencing specifications in RFC 1717.

LFI allows reserve queues to be set up so that Real-Time Protocol (RTP) streams can be mapped into a higher priority queue in the configured weighted fair queue set.

Note A related IETF Draft called "Multiclass Extensions to Multilink PPP (MCML)" describes the MCML feature, which implements nearly the same function as LFI.

For information on how to configure LFI, see the chapter "Configuring Link Fragmentation and Interleaving for Multilink PPP" or "Configuring Link Fragmentation and Interleaving for Frame Relay and ATM Virtual Circuits" in this book.

How It Works

To understand how LFI using MLP works, it helps to understand the problem it addresses. The complete end-to-end delay target for real-time packets, especially voice packets, is 150 to 200 milliseconds (ms). The IP-based datagram transmission techniques for audio transmission do not adequately address the problems posed by limited bandwidth and the very stringent telephony delay bound of 150 ms.

Unacceptable queueing delays for small real-time packets exist regardless of use of QoS features such as Resource Reservation Protocol (RSVP) and weighted fair queueing (WFQ), and use of voice compression algorithms such as code excited linear prediction (CELP) compression, which reduces the inherent bit rate from 64 kbps to as low as 8 kbps. Despite these measures, real-time delay continues to exist because per-packet header overhead is too large and large maximum transmission units (MTUs) are needed to produce acceptable bulk transmission efficiency.

A large MTU of 1500 bytes takes 215 ms to traverse a 56-kbps line, which exceeds the delay target. Therefore, to limit the delay of real-time packets on relatively slow bandwidth links—links such as 56-kbps Frame Relay or 64-kbps ISDN B channels—a method for fragmenting larger packets and queueing smaller packets between fragments of the large packet is needed. MLP helps to solve this problem through LFI.

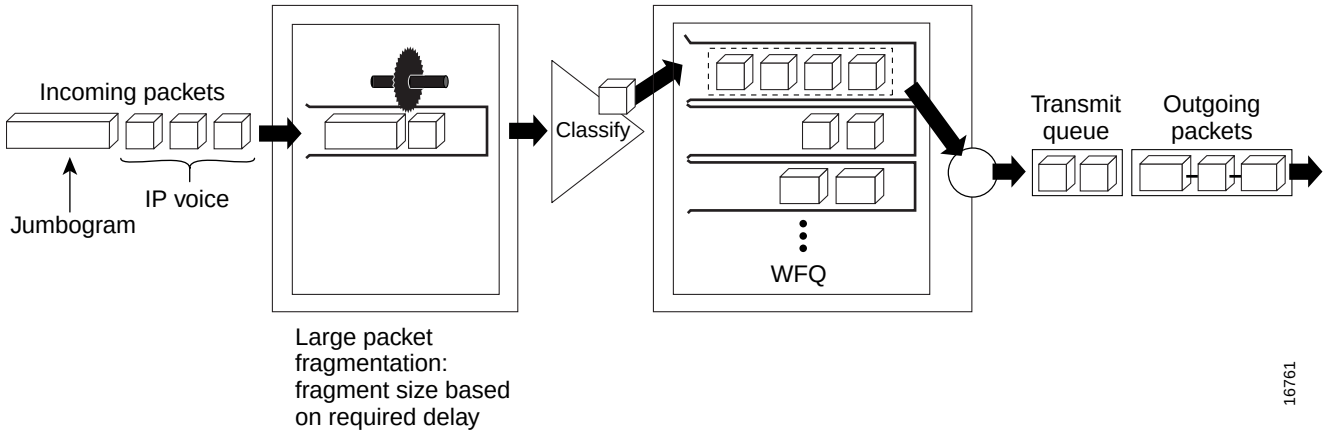
MLP provides a method of splitting, recombining, and sequencing datagrams across multiple logical data links. The LFI scheme is relatively simple: Large datagrams are multilink encapsulated and fragmented to packets of a size small enough to satisfy the delay requirements of the delay-sensitive traffic; small delay-sensitive packets are not multilink encapsulated, but are interleaved between fragments of the large datagram.

MLP allows the fragmented packets to be sent at the same time over multiple point-to-point links to the same remote address. The multiple links come up in response to a dialer load threshold that you define. The load can be calculated on inbound traffic, outbound traffic, or on either, as needed for the traffic between the specific sites. MLP provides bandwidth on demand and reduces transmission latency across WAN links.

Figure 25 shows the mix of traffic destined for an interface as including both jumbograms and smaller, time-sensitive IP voice packets. Based on their classifications, these arriving packets are sorted into queues. After the packets are queued, the jumbogram is fragmented into smaller packets in preparation for interleaving with the time-sensitive IP voice packets. Because WFQ is configured for the interface, packets from each queue—that is, the jumbogram packet fragments and the IP voice packets—are interleaved and scheduled (fairly and based on their weight) for transmission in the output interface queue.

To ensure correct order of transmission and reassembly, LFI adds multilink headers to the datagram fragments after the packets are dequeued and ready to be sent.

Figure 25 Link Fragmentation and Interleaving



Interleaving can occur at process-fast paths. However, because it relies on MLP, its performance is closely tied with multilink behavior.

Note LFI on PPP over Frame Relay is not supported on Cisco IOS Release 12.1E.

Link Fragmentation and Interleaving for Frame Relay and ATM VCs

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

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

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

This new feature enhances 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.

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

- Frame Relay Traffic Shaping (FRTS)
- Low latency queueing (LLQ)
- Class-based weighted fair queueing (CBWFQ)

The LFI for Frame Relay and ATM VCs feature supports RFC 1990, *The PPP Multilink Protocol (MP)*.

Restrictions

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

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

Note LFI on PPP over Frame Relay is not supported on Cisco IOS Release 12.1E.

Prerequisites

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

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

For information on how to configure the LFI for Frame Relay and ATM VCs feature, see the chapter "Configuring Link Fragmentation and Interleaving for Frame Relay and ATM Virtual Circuits" in this book.

Frame Relay Fragmentation

Cisco has developed the following three methods of performing Frame Relay fragmentation:

- End-to-end FRF.12 fragmentation
- Frame Relay fragmentation using FRF.11 Annex C
- Cisco proprietary voice encapsulation

For more information about Frame Relay fragmentation methods, refer to the *Cisco IOS Wide-Area Networking Configuration Guide* and the *Cisco IOS Voice, Video, and Fax Configuration Guide*.

Compressed Real-Time Protocol

Real-Time Protocol (RTP) is the Internet Standard (RFC 1889) protocol for the transport of real-time data. It is intended to provide end-to-end network transport functions for applications that support audio, video, or simulation data over multicast or unicast network services.

RTP provides support for real-time conferencing of groups of any size within the Internet. This support includes source identification and support for gateways such as audio and video bridges, and for multicast-to-unicast translators. RTP offers QoS feedback from receivers to the multicast group and support for the synchronization of different media streams.

RTP includes a data portion and a header portion. The data portion of RTP is a thin protocol that provides support for the real-time properties of applications, such as continuous media, including timing reconstruction, loss detection, and content identification.

The header portion of RTP is considerably large. As shown in Figure 26, the minimal 12 bytes of the RTP header, combined with 20 bytes of IP header (IPH) and 8 bytes of User Datagram Protocol (UDP) header, create a 40-byte IP/UDP/RTP header. For compressed-payload audio applications, the RTP packet typically has a 20-byte to 160-byte payload. Given the size of the IP/UDP/RTP header combinations, it is inefficient to send the IP/UDP/RTP header without compressing it.

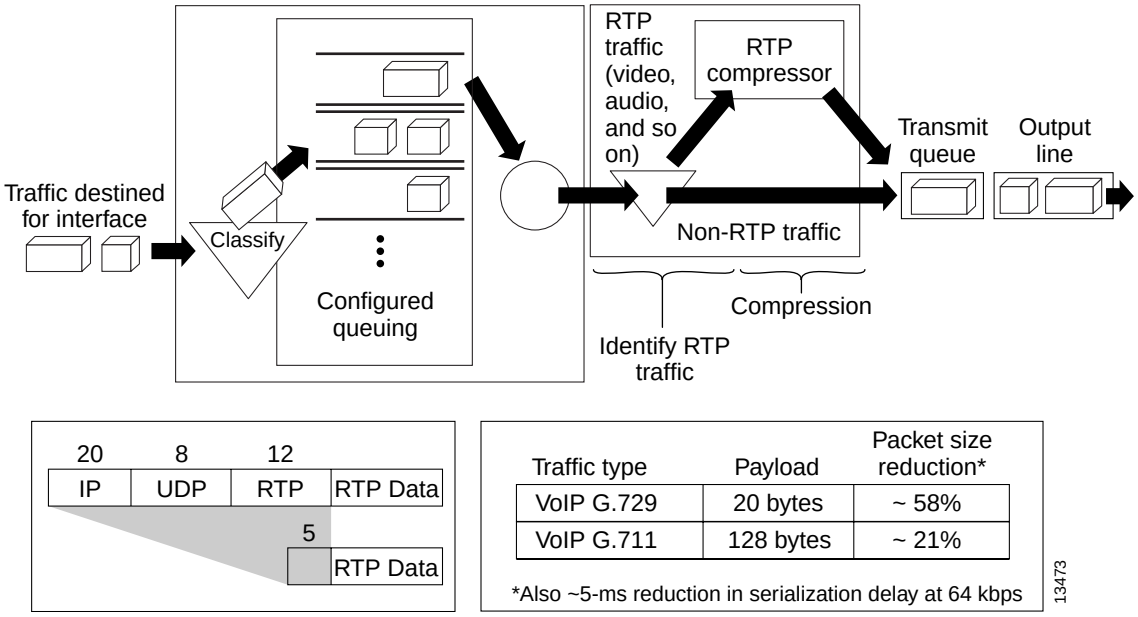
To avoid the unnecessary consumption of available bandwidth, the RTP header compression feature—referred to as CRTP—is used on a link-by-link basis.

For information on how to configure CRTP, see the chapter "Configuring Compressed Real-Time Protocol" in this book.

How It Works

CRTP compresses the IP/UDP/RTP header in an RTP data packet from 40 bytes to approximately 2 to 5 bytes. Figure 26 illustrates this process.

Figure 26 RTP Header Compression



CRTP accrues major gain in terms of packet compression because although several fields in the header change in every packet, the difference from packet to packet is often constant, and therefore the second-order difference is zero. The decompressor can reconstruct the original header without any loss of information.

CRTP is a hop-by-hop compression scheme similar to RFC 1144 for TCP header compression.

Why Use CRTP Header?

The CRTP reduction in line overhead for multimedia RTP traffic results in a corresponding reduction in delay; CRTP is especially beneficial when the RTP payload size is small, for example, for compressed audio payloads of 20 to 50 bytes.

You should use CRTP on any WAN interface where bandwidth is a concern and there is a high portion of RTP traffic. CRTP can be used for media-on-demand and interactive services such as Internet telephony. As with RTP, CRTP provides support for real-time conferencing of groups of any size within the Internet. This support includes source identification and support for gateways such as audio and video bridges and for multicast-to-unicast translators. CRTP can benefit both telephony voice and multicast backbone (MBONE) applications running over slow links.

You should not use CRTP on any high-speed interfaces—that is, anything over T1 speed—because the trade-offs are not desirable.

CRTP is supported on serial lines using Frame Relay, High-Level Data Link Control (HDLC), or PPP encapsulation. It is also supported over ISDN interfaces.

CRTP for Frame Relay is supported using Cisco-format encapsulation only.

Express RTP Header Compression

Before Cisco IOS Release 12.0(7)T, if compression of TCP or RTP headers was enabled, compression was performed in the process switching path, which meant that packets traversing interfaces that had TCP or RTP header compression enabled were queued and passed up to the process to be switched. This procedure slowed down transmission of the packet, and therefore some users preferred to fast-switch uncompressed TCP and RTP packets.

With Release 12.1, if TCP or RTP header compression is enabled, it occurs by default in the fast-switched path or the Cisco Express Forwarding-switched (CEF-switched) path, depending on which switching method is enabled on the interface.

If neither fast-switching nor CEF switching is enabled, if RTP header compression is enabled, it will occur in the process-switched path as before.

The Express RTP Header Compression feature is not available for Async and Dialer interfaces.

For more information on the Express RTP Header Compression feature, refer to the *Cisco IOS IP Configuration Guide*.

Distributed Compressed Real-Time Protocol

Before Cisco IOS Release 12.1(5)T, if compression of TCP or RTP headers was enabled on a Cisco 7500 series router with a Versatile Interface Processor (VIP), the compression was performed in the process-switching path, which meant that packets traversing interfaces that had TCP or RTP header compression enabled were queued and passed up to the Route Switch Processor (RSP) to be switched. This procedure slowed down transmission of the packet, and therefore some users preferred to fast-switch uncompressed TCP and RTP packets rather than enable TCP and RTP compression.

If the dCRTP feature is enabled, the header compression of the combined IP/UDP/RTP header occurs by default in the distributed fast-switched path or the distributed CEF-switched (dCEF-switched) path, depending on which switching method is enabled on the interface.

If distributed fast-switching or dCEF switching are disabled, TCP or RTP header compression will occur in the process-switched path as before.

This feature is supported on Cisco 7500 series routers with a VIP.

The dCRTP feature supports the following RFCs:

- RFC 1144, *Compressing TCP/IP Headers for Low-Speed Serial Links*
- RFC 2507, *IP Header Compression*
- RFC 2508, *Compressing IP/UDP/RTP Headers for Low-Speed Serial Links*

For information on how to configure the dCRTP feature, see the chapter "Configuring Distributed Compressed Real-Time Protocol" in this book.

Benefits

Additional Functionality Capabilities for the RSP

The dCRTP feature offloads the IP/UDP/RTP header compression from the RSP, scaling it for other functionality.

Enhanced 7500 Series Router Scalability for Enterprise and Service Provider Networks

The dCRTP feature helps support Compressed Real-Time Protocol (CRTP) for large enterprise and service provider networks on a single Cisco 7500 series router acting as an aggregation point.

Additional Support for VoIP Streams

The dCRTP feature allows for more VoIP streams to be supported without any major performance degradation on the RSP.

Accelerates Speed of Packet Transmission

The dCRTP feature reduces the size of the packet, which allows for a higher packet transmission speed.

Improved Latency

The dCRTP feature reduces the size of the packet. The smaller packet leaves less latency on a transmission ring, allowing for higher data quality.

Restrictions

The following restrictions apply to the dCRTP feature:

- Because statistical updates are sent to the RSP by the VIP once every 10 seconds, a 10-second delay may be experienced when displaying traffic statistics using the **show ip rtp header-compression** or **show ip tcp header-compression** command.
- The detail option is not available with the **show ip rtp header-compression** and **show ip tcp header-compression** commands when distributed fast-switching is enabled. Users who need the detailed information for either of these commands can retrieve this information by disabling distributed fast-switching and then entering the **show ip rtp header-compression detail** or **show ip tcp header-compression detail** command.
- This restriction affects MLP interfaces that use LFI. In this case, if RTP header compression is configured, RTP packets originating on or destined to the router will be fast-switched if the link is limited to one channel. If the link has more than one channel, the packets will be process-switched.
- This feature is not available for Async and Dialer interfaces.

Prerequisites

In order for this feature to work, the following prerequisites must be met:

- Distributed CEF switching or distributed fast-switching must be enabled on the interface.
- HDLC, PPP, or Frame Relay encapsulation must be configured.
- TCP or RTP header compression, or both, must be enabled:
  - For information on configuring RTP header compression, see the chapter "Configuring Compressed Real-Time Protocol" in this book.

For information on configuring TCP header compression, refer to the *Cisco IOS IP Configuration Guide*.