



Link Efficiency Mechanisms Overview

Cisco IOS software offers two link-layer efficiency mechanisms—Link Fragmentation and Interleaving (LFI) for Multilink Point-to-Point Protocol (MLP), and Compressed Real-Time Protocol (CRTP) header—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
- Frame Relay Fragmentation
- Compressed Real-Time Protocol Header

Link Fragmentation and Interleaving

Interactive traffic such as Telnet and Voice over IP is susceptible to increased latency when the network processes large packets such as LAN-to-LAN File Transfer Protocol (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 Cisco's 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.



Note

For information on how to configure LFI, see the chapter “Configuring Link Fragmentation and Interleaving for Multilink PPP” 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 Compressed Encoding for Linear Prediction (CELP), 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.

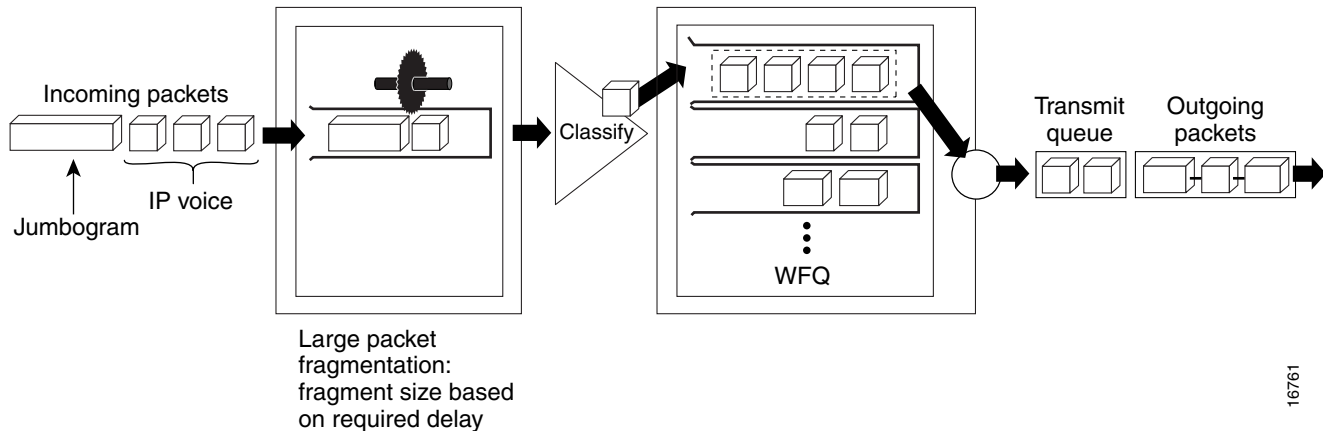
MLP provides a method of splitting, recombining, and sequencing datagrams across multiple logical data links. The LFI scheme is relatively simple: Large datagrams are multilink encapsulated and fragmented to packets of a size small enough to satisfy the delay requirements of the delay-sensitive traffic; 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 15 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 15 Link Fragmentation and Interleaving



16761

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

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 on these Frame Relay fragmentation methods, refer to the “Configuring Voice over Frame Relay” chapter in the *Cisco IOS Multiservice Applications Configuration Guide*.

Compressed Real-Time Protocol Header

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 as well as 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 16, the minimal 12 bytes of the RTP header, combined with 20 bytes of IP header (IPH) and 8 bytes of 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.

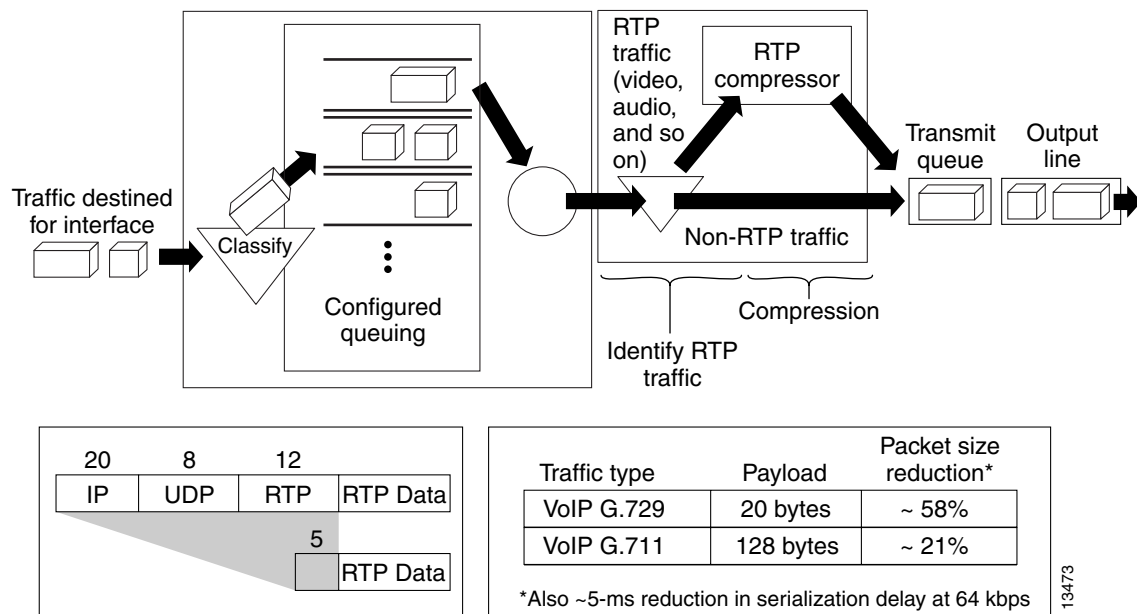
**Note**

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 16 illustrates this process.

Figure 16 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?

CRTP's 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 as well as 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 Real-Time Transport Protocol (RTP) headers was enabled, compression was performed in the process switching path. That 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.

**Note**

For more information on the Express RTP Header Compression feature, refer to the “Configuring IP Multicast Routing” chapter in the *Cisco IOS IP and IP Routing Configuration Guide*.
