



Header Compression

First Published: January 30, 2006
Last Updated: June 19, 2006

Header compression is a mechanism that compresses the IP header in a packet before the packet is transmitted. Cisco provides two types of header compression: RTP header compression (used for RTP packets) and TCP header compression (used for TCP packets).

This module contains a high-level overview of header compression. Before configuring header compression, you need to understand the information contained in this module.

Contents

- [Information About Header Compression, page 1](#)
- [Where to Go Next, page 6](#)
- [Additional References, page 6](#)
- [Glossary, page 9](#)

Information About Header Compression

Before configuring header compression, you should understand the following concepts:

- [Header Compression Defined, page 2](#)
- [Types of Header Compression, page 2](#)
- [RTP Functionality and Header Compression, page 2](#)
- [TCP Functionality and Header Compression, page 4](#)
- [Class-Based Header Compression Functionality, page 5](#)
- [IPHC Profiles and Header Compression, page 6](#)



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

© 2007 Cisco Systems, Inc. All rights reserved.

Header Compression Defined

Header compression is a mechanism that compresses the IP header in a data packet before the packet is transmitted. Header compression reduces network overhead and speeds up the transmission of Real-Time Transport Protocol (RTP) and Transmission Control Protocol (TCP) packets. Header compression also reduces the amount of bandwidth consumed when the RTP or TCP packets are transmitted.

Types of Header Compression

Cisco provides the following two types of header compression:

- RTP header compression (used for RTP packets)
- TCP header compression (used for TCP packets)

Both RTP header compression and TCP header compression treat packets in a similar fashion, as described in the sections that follow.

**Note**

RTP and TCP header compression are typically configured on a *per-interface* (or *subinterface*) basis. However, you can choose to configure either RTP header compression or TCP header compression on a *per-class* basis using the Modular Quality of Service (QoS) Command-Line Interface (CLI) (MQC). More information about class-based RTP and TCP header compression is provided later in this module.

RTP Functionality and Header Compression

RTP provides end-to-end network transport functions for applications that support audio, video, or simulation data over unicast or multicast services.

RTP provides support for real-time conferencing of groups of any size within the Internet. This support includes source identification support for gateways such as audio and video bridges, and support for multicast-to-unicast translators. RTP provides 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 larger than the data portion. The header portion consists of the IP segment, the User Datagram Protocol (UDP) segment, and the RTP segment. Given the size of the IP/UDP/RTP segment combinations, it is inefficient to send the IP/UDP/RTP header without compressing it.

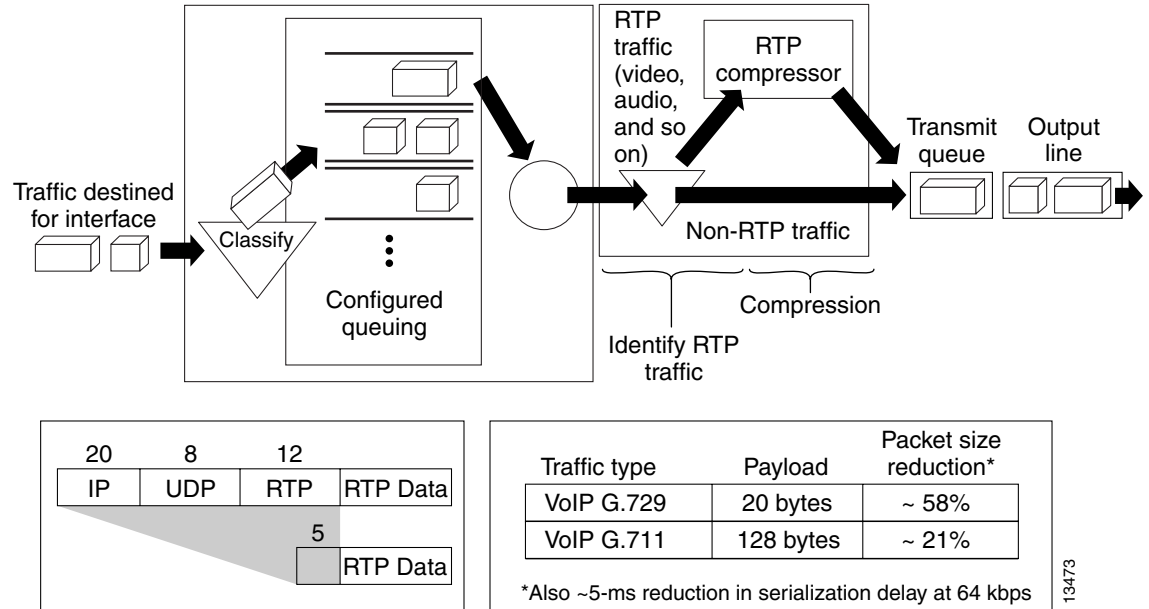
To avoid the unnecessary consumption of available bandwidth, RTP header compression is used on a link-by-link basis.

How RTP Header Compression Works

RTP header compression compresses the RTP header (that is, the combined IP, UDP, and RTP segments) in an RTP packet. [Figure 1](#) illustrates this process and shows how RTP header compression treats incoming packets.

In this example, packets arrive at an interface and the packets are classified. After the packets are classified, they are queued for transmission according to the configured queuing mechanism.

Figure 1 RTP Header Compression



For most audio applications, the RTP packet typically has a 20- to 128-byte payload.

RTP header compression identifies the RTP traffic and then compresses the IP header portion of the RTP packet. The IP header portion consists of an IP segment, a UDP segment, and an RTP segment. In [Figure 1](#), the minimal 20 bytes of the IP segment, combined with the 8 bytes of the UDP segment, and the 12 bytes of the RTP segment, create a 40-byte IP/UDP/RTP header. In [Figure 1](#), the RTP header portion is compressed from 40 bytes to approximately 5 bytes.



Note

RTP header compression is supported on serial interfaces using Frame Relay, HDLC, or PPP encapsulation. It is also supported over ISDN interfaces.

Why Use RTP Header Compression

RTP header compression accrues major gains 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.

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

Use RTP header compression on any WAN interface where you are concerned about bandwidth and where there is a high portion of RTP traffic. RTP header compression can be used for media-on-demand and interactive services such as Internet telephony. RTP header compression provides support for real-time conferencing of groups of any size within the Internet. This support includes source

identification support for gateways such as audio and video bridges, and support for multicast-to-unicast translators. RTP header compression can benefit both telephony voice and multicast backbone (MBONE) applications running over slow links.

**Note**

Using RTP header compression on any high-speed interfaces—that is, anything over T1 speed—is not recommended. Any bandwidth savings achieved with RTP header compression may be offset by an increase in CPU utilization on the router.

TCP Functionality and Header Compression

TCP provides a reliable end-to-end network transport for applications such as FTP, Telnet, and HTTP. TCP uses the connectionless service provided by IP.

TCP includes a data portion and a header portion. The header portion of TCP is considerably larger than the data portion. The header portion consists of the IP segment and the TCP segment. Given the size of the TCP/IP segment combinations, it is inefficient to send the TCP/IP header without compressing it.

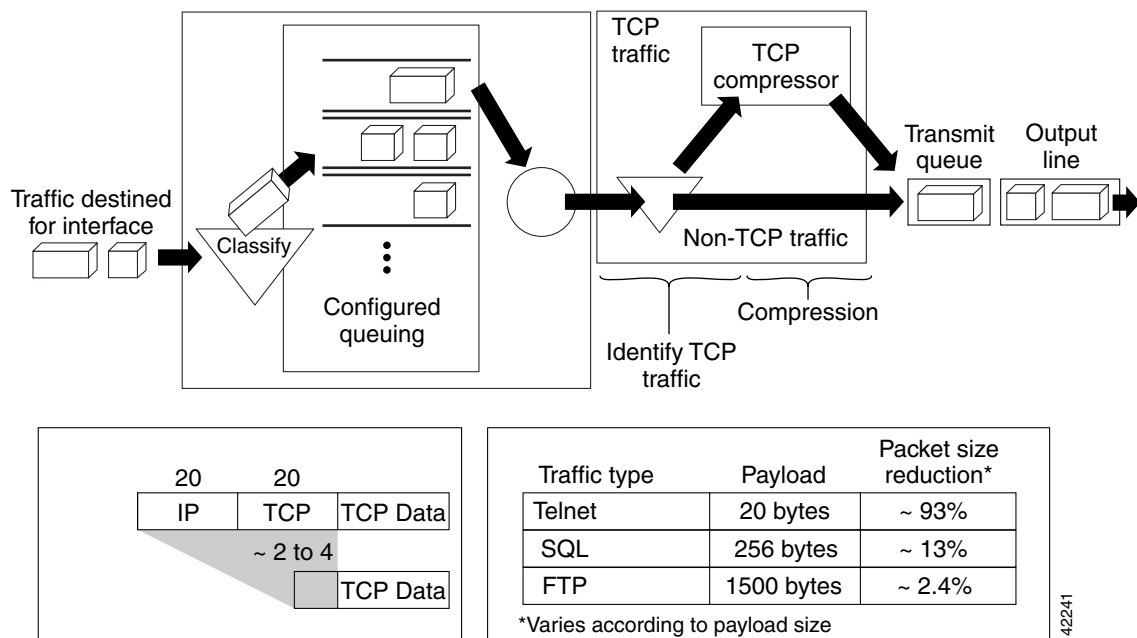
To avoid the unnecessary consumption of available bandwidth, TCP header compression is used on a link-by-link basis.

How TCP Header Compression Works

TCP header compression compresses the TCP header (that is, the combined IP and TCP segments) in a TCP packet. [Figure 2](#) illustrates this process and shows how TCP header compression treats incoming packets.

In this example, packets arrive at an interface and the packets are classified. After the packets are classified, they are queued for transmission according to the configured queuing mechanism.

Figure 2 TCP Header Compression



142241

For TCP applications, the TCP packet typically has a 1- to 1500-byte payload.

TCP header compression identifies the TCP traffic and then compresses the IP header portion of the TCP packet. The IP header portion consists of an IP segment and a TCP segment. In [Figure 2](#), the 20 bytes of the IP segment, combined with the 20 bytes of the TCP segment, creates a 40-byte TCP/IP header. In [Figure 2](#), the TCP/IP header portion is compressed from 40 bytes to approximately 2 to 4 bytes.

**Note**

TCP header compression is supported on serial interfaces using Frame Relay, HDLC, or PPP encapsulation. It is also supported over ISDN interfaces.

Why Use TCP Header Compression

TCP header compression accrues major gains 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.

TCP header compression also reduces overhead. The reduction in overhead for TCP traffic results in a corresponding reduction in delay; TCP header compression is especially beneficial when the TCP payload size is small, for example, for interactive traffic such as Telnet.

Use TCP header compression on any WAN interface where you are concerned about bandwidth and where there is a high portion of TCP traffic.

**Note**

Using TCP header compression on any high-speed interfaces—that is, anything over T1 speed—is not recommended. Any bandwidth savings achieved with TCP header compression may be offset by an increase in CPU utilization on the router.

Class-Based Header Compression Functionality

Class-based header compression uses the same functionality as RTP and TCP header compression described earlier in this module. That is, class-based header compression treats packets the same way as described in the [“RTP Functionality and Header Compression”](#) and the [“TCP Functionality and Header Compression”](#) sections of this module, respectively.

Class-based header compression is simply another method you can choose when you configure either RTP header compression or TCP header compression on your network.

RTP and TCP header compression are typically configured on a *per-interface* (or subinterface) basis. Class-based header compression (RTP or TCP) is configured on a *per-class* basis using the MQC.

The MQC is a CLI that allows you to create classes within policy maps (traffic policies) and then attach the policy maps to interfaces. The policy maps are used to configure specific QoS features (such as RTP or TCP header compression) on your network.

For more information about the MQC, see the [“Applying QoS Features Using the MQC”](#) module.

Why Use Class-Based Header Compression

Class-based header compression allows you to compress (and then decompress) a subset of the packets on your network. Class-based header compression acts as a filter; it allows you to specify at a much finer level the packets that you want to compress using either RTP header compression or TCP header compression.

For example, instead of compressing all RTP (or all TCP) packets that are traversing your network, you can configure RTP header compression to compress only those packets that meet certain criteria (for example, protocol type “ip” in a class called “voice”).

IPHC Profiles and Header Compression

One method of configuring header compression on your network is to use an IP header compression (IPHC) profile. An IPHC profile is a kind of template within which you can configure the type of header compression you want to use, set all of the optional features and settings for header compression, and then apply the profile to an interface, subinterface, or Frame Relay permanent virtual circuit (PVC).

Why Use IPHC Profiles

Provides More Flexibility

You can enable header compression (along with any optional settings you want to use) *once* in an IPHC profile, and then apply that IPHC profile to as many interfaces, subinterfaces, or Frame Relay PVCs as needed.

Where to Go Next

Where you go next depends on the type of header compression that you want to configure, and whether you want to use IPHC profiles to configure header compression.

- To configure RTP header compression, see the [“Configuring RTP Header Compression”](#) module.
- To configure TCP header compression, see the [“Configuring TCP Header Compression”](#) module.
- To configure class-based RTP or TCP header compression, see the [“Configuring Class-Based RTP and TCP Header Compression”](#) module.
- To configure header compression using IPHC profiles, see the [“Configuring Header Compression Using IPHC Profiles”](#) module.

Additional References

The following sections provide references related to header compression.

Related Documents

Related Topic	Document Title
QoS commands: complete command syntax, command modes, command history, defaults, usage guidelines, and examples	Cisco IOS Quality of Service Solutions Command Reference
MQC	“Applying QoS Features Using the MQC” module
RTP header compression	“Configuring RTP Header Compression” module
TCP header compression	“Configuring TCP Header Compression” module
Class-based RTP and TCP header compression	“Configuring Class-Based RTP and TCP Header Compression” module
IPHC profiles and header compression	“Configuring Header Compression Using IPHC Profiles” module

Standards

Standard	Title
No new or modified standards are supported, and support for existing standards has not been modified.	—

MIBs

MIB	MIBs Link
No new or modified MIBs are supported, and support for existing MIBs has not been modified.	To locate and download MIBs for selected platforms, Cisco IOS releases, and feature sets, use Cisco MIB Locator found at the following URL: http://www.cisco.com/go/mibs

RFCs

RFC	Title
RFC 1144	<i>Compressing TCP/IP Headers for Low-Speed Serial Links</i>
RFC 2507	<i>IP Header Compression</i>
RFC 2508	<i>Compressing IP/UDP/RTP Headers for Low-Speed Serial Links</i>
RFC 3544	<i>IP Header Compression over PPP</i>
RFC 3545	<i>Enhanced Compressed RTP (CRTP) for Links with High Delay, Packet Loss and Reordering</i>
RFC 3550	<i>A Transport Protocol for Real-Time Applications</i>

Technical Assistance

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

Glossary

compression—The running of a data set through an algorithm that reduces the space required to store the data set or the bandwidth required to transmit the data set.

decompression—The act of reconstructing a compressed header.

HDLC—High-Level Data Link Control. A bit-oriented synchronous data link layer protocol developed by International Organization for Standardization (ISO). Derived from Synchronous Data Link Control (SDLC), HDLC specifies a data encapsulation method on synchronous serial links using frame characters and checksums.

header—A chain of subheaders.

incorrect decompression—The circumstance in which a compressed and then decompressed header is different from the uncompressed header. This variance is usually due to a mismatched context between the compressor and decompressor or bit errors during transmission of the compressed header.

ISDN—Integrated Services Digital Network. A communication protocol offered by telephone companies that permits telephone networks to carry data, voice, and other source traffic.

MQC—Modular Quality of Service Command-Line Interface. The MQC allows you to create traffic classes and policy maps and then attach the policy maps to interfaces. The policy maps apply QoS features to your network.

PPP—Point-to-Point Protocol. A protocol that provides router-to-router and host-to-network connections over synchronous and asynchronous circuits.

regular header—A normal, uncompressed header. A regular header does not carry a context identifier (CID) or generation association.

RTP—Real-Time Transport Protocol. A protocol that is designed to provide end-to-end network transport functions for applications that transmit real-time data, such as audio, video, or simulation data, over unicast or multicast network services. RTP provides such services as payload type identification, sequence numbering, timestamping, and delivery monitoring to real-time applications.

subheader—An IPv6 base header, an IPv6 extension header, an IPv4 header, a UDP header, an RTP header, or a TCP header.

TCP—Transmission Control Protocol. A connection-oriented transport layer protocol that provides reliable full-duplex data transmission. TCP is part of the TCP/IP protocol stack.

UDP—User Datagram Protocol. A connectionless transport layer protocol in the TCP/IP protocol stack. UDP is a simple protocol that exchanges datagrams without acknowledgments or guaranteed delivery, requiring that error processing and retransmission be handled by other protocols. UDP is defined in RFC 768.

CCDE, CCENT, CCSI, Cisco Eos, Cisco HealthPresence, Cisco IronPort, the Cisco logo, Cisco Lumin, Cisco Nexus, Cisco Nurse Connect, Cisco Pulse, Cisco StackPower, Cisco StadiumVision, Cisco TelePresence, Cisco Unified Computing System, Cisco WebEx, DCE, Flip Channels, Flip for Good, Flip Mino, Flipshare (Design), Flip Ultra, Flip Video, Flip Video (Design), Instant Broadband, and Welcome to the Human Network are trademarks; Changing the Way We Work, Live, Play, and Learn, Cisco Capital, Cisco Capital (Design), Cisco:Financed (Stylized), Cisco Store, and Flip Gift Card are service marks; and Access Registrar, Aironet, AllTouch, AsyncOS, Bringing the Meeting To You, Catalyst, CCDA, CCDP, CCIE, CCIP, CCNA, CCNP, CCSP, CCVP, Cisco, the Cisco Certified Internetwork Expert logo, Cisco IOS, Cisco Press, Cisco Systems, Cisco Systems Capital, the Cisco Systems logo, Cisco Unity, Collaboration Without Limitation, Continuum, EtherFast, EtherSwitch, Event Center, Explorer, Fast Step, Follow Me Browsing, FormShare, GainMaker, GigaDrive, HomeLink, iLynX, Internet Quotient, IOS, iPhone, iQuick Study, IronPort, the IronPort logo, Laser Link, LightStream, Linksys, MediaTone, MeetingPlace, MeetingPlace Chime Sound, MGX, Networkers, Networking Academy, Network Registrar, PCNow, PIX, PowerKEY, PowerPanels, PowerTV, PowerTV (Design), PowerVu, Prisma, ProConnect, ROSA, ScriptShare, SenderBase, SMARTnet, Spectrum Expert, StackWise, The Fastest Way to Increase Your Internet Quotient, TransPath, WebEx, and the WebEx logo are registered trademarks of Cisco Systems, Inc. and/or its affiliates in the United States and certain other countries.

All other trademarks mentioned in this document or website are the property of their respective owners. The use of the word partner does not imply a partnership relationship between Cisco and any other company. (0908R)

Any Internet Protocol (IP) addresses used in this document are not intended to be actual addresses. Any examples, command display output, and figures included in the document are shown for illustrative purposes only. Any use of actual IP addresses in illustrative content is unintentional and coincidental.

© 2007 Cisco Systems, Inc. All rights reserved.