



QoS: Header Compression Configuration Guide Cisco IOS Release 12.2SR

Americas Headquarters

Cisco Systems, Inc.
170 West Tasman Drive
San Jose, CA 95134-1706
USA
<http://www.cisco.com>
Tel: 408 526-4000
800 553-NETS (6387)
Fax: 408 527-0883

THE SPECIFICATIONS AND INFORMATION REGARDING THE PRODUCTS IN THIS MANUAL ARE SUBJECT TO CHANGE WITHOUT NOTICE. ALL STATEMENTS, INFORMATION, AND RECOMMENDATIONS IN THIS MANUAL ARE BELIEVED TO BE ACCURATE BUT ARE PRESENTED WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED. USERS MUST TAKE FULL RESPONSIBILITY FOR THEIR APPLICATION OF ANY PRODUCTS.

THE SOFTWARE LICENSE AND LIMITED WARRANTY FOR THE ACCOMPANYING PRODUCT ARE SET FORTH IN THE INFORMATION PACKET THAT SHIPPED WITH THE PRODUCT AND ARE INCORPORATED HEREIN BY THIS REFERENCE. IF YOU ARE UNABLE TO LOCATE THE SOFTWARE LICENSE OR LIMITED WARRANTY, CONTACT YOUR CISCO REPRESENTATIVE FOR A COPY.

The Cisco implementation of TCP header compression is an adaptation of a program developed by the University of California, Berkeley (UCB) as part of UCB's public domain version of the UNIX operating system. All rights reserved. Copyright © 1981, Regents of the University of California.

NOTWITHSTANDING ANY OTHER WARRANTY HEREIN, ALL DOCUMENT FILES AND SOFTWARE OF THESE SUPPLIERS ARE PROVIDED "AS IS" WITH ALL FAULTS. CISCO AND THE ABOVE-NAMED SUPPLIERS DISCLAIM ALL WARRANTIES, EXPRESSED OR IMPLIED, INCLUDING, WITHOUT LIMITATION, THOSE OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT OR ARISING FROM A COURSE OF DEALING, USAGE, OR TRADE PRACTICE.

IN NO EVENT SHALL CISCO OR ITS SUPPLIERS BE LIABLE FOR ANY INDIRECT, SPECIAL, CONSEQUENTIAL, OR INCIDENTAL DAMAGES, INCLUDING, WITHOUT LIMITATION, LOST PROFITS OR LOSS OR DAMAGE TO DATA ARISING OUT OF THE USE OR INABILITY TO USE THIS MANUAL, EVEN IF CISCO OR ITS SUPPLIERS HAVE BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

Cisco and the Cisco logo are trademarks or registered trademarks of Cisco and/or its affiliates in the U.S. and other countries. To view a list of Cisco trademarks, go to this URL: www.cisco.com/go/trademarks. Third-party trademarks mentioned 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. (1110R)

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

© 2011 Cisco Systems, Inc. All rights reserved.



CONTENTS

Header Compression 1

Finding Feature Information 1

Information About Header Compression 1

Header Compression Defined 1

RTP Functionality and Header Compression 1

How RTP Header Compression Works 2

Why Use RTP Header Compression 3

Additional References 3

Glossary 4

Configuring RTP Header Compression 7

Finding Feature Information 7

Prerequisites for Configuring RTP Header Compression 7

Information About Configuring RTP Header Compression 8

Configurable RTP Header-Compression Settings 8

RTP Header-Compression Keywords 8

Enhanced RTP Header Compression 9

RTP Header Compression over Satellite Links 10

Periodic Refreshes of a Compressed Packet Stream 10

Optional Disabling of Context-Status Messages 11

How to Configure RTP Header Compression 11

Enabling RTP Header Compression on an Interface 11

Enabling RTP Header Compression on an Interface That Uses Frame Relay Encapsulation 13

Enabling Enhanced RTP Header Compression 15

Enabling RTP Header Compression over a Satellite Link 17

Specifying the Header-Compression Settings 18

Changing the Number of Header-Compression Connections 20

Implications of Changing the Number of Header-Compression Connections 20

Displaying Header-Compression Statistics 22

Configuration Examples for RTP Header Compression 23

Example Enabling RTP Header Compression on an Interface **24**
Example Enabling RTP Header Compression on an Interface That Uses Frame Relay
Encapsulation **24**
Example Enabling Enhanced RTP Header Compression **25**
Example Enabling RTP Header Compression over a Satellite Link **25**
Example Specifying the Header-Compression Settings **26**
Example Changing the Number of Header-Compression Connections **26**
Example Displaying Header-Compression Statistic **26**
Additional References **27**
Glossary **28**
Feature Information for Configuring RTP Header Compression **29**



Header Compression

Header compression is a mechanism that compresses the header in a packet before the packet is transmitted. For Cisco IOS XE Software, Cisco provides RTP header compression (used for RTP packets).

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

- [Finding Feature Information, page 1](#)
- [Information About Header Compression, page 1](#)
- [Additional References, page 3](#)
- [Glossary, page 4](#)

Finding Feature Information

Your software release may not support all the features documented in this module. For the latest feature information and caveats, see the release notes for your platform and software release. To find information about the features documented in this module, and to see a list of the releases in which each feature is supported, see the Feature Information Table at the end of this document.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to www.cisco.com/go/cfn. An account on Cisco.com is not required.

Information About Header Compression

- [Header Compression Defined, page 1](#)
- [RTP Functionality and Header Compression, page 1](#)

Header Compression Defined

Header compression is a mechanism that compresses the header in a data packet before the packet is transmitted. Header compression reduces network overhead and speeds up the transmission of packets. Header compression also reduces the amount of bandwidth consumed when the packets are transmitted.

RTP Functionality and Header Compression

Real-Time Transport Protocol (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.

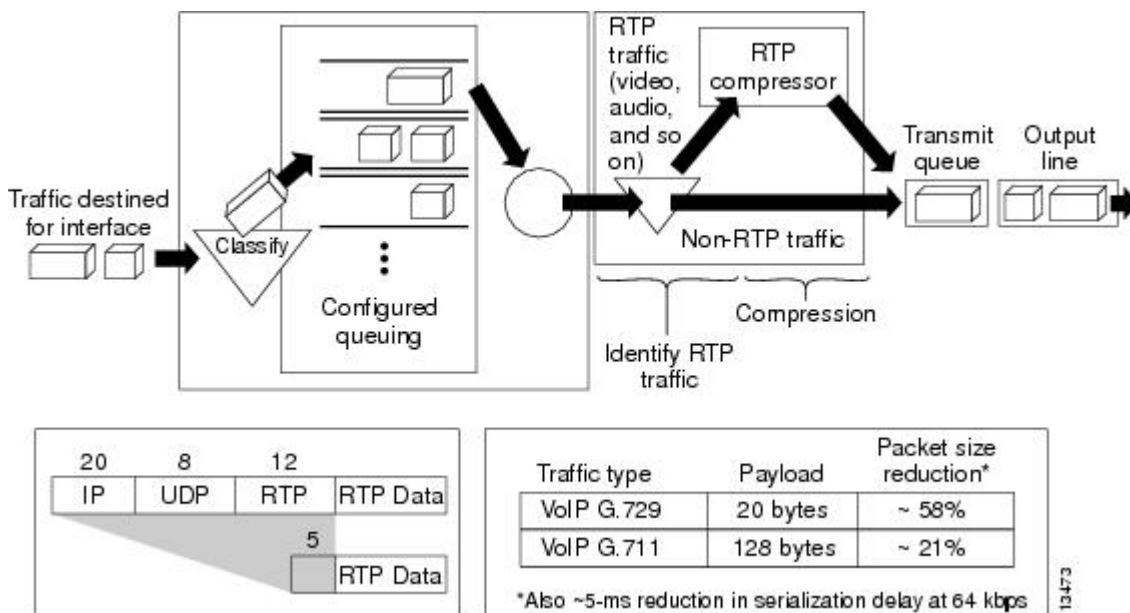
- [How RTP Header Compression Works, page 2](#)
- [Why Use RTP Header Compression, page 3](#)

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. The figure below 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 the figure above, 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 the figure above, 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.

Additional References

The following sections provide references related to header compression.

Related Documents

Related Topic	Document Title
Cisco IOS commands	Cisco IOS Master Commands List, All Releases
QoS commands: complete command syntax, command modes, command history, defaults, usage guidelines, and examples	<i>Cisco IOS Quality of Service Solutions Command Reference</i>
MQC	"Applying QoS Features Using the MQC" module
RTP header compression	"Configuring RTP Header Compression" 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 XE Software 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 3550	<i>A Transport Protocol for Real-Time Applications</i>

Technical Assistance

Description	Link
The Cisco Support and Documentation website provides online resources to download documentation, software, and tools. Use these resources to install and configure the software and to troubleshoot and resolve technical issues with Cisco products and technologies. Access to most tools on the Cisco Support and Documentation website requires a Cisco.com user ID and password.	http://www.cisco.com/cisco/web/support/index.html

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.

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.

Cisco and the Cisco logo are trademarks or registered trademarks of Cisco and/or its affiliates in the U.S. and other countries. To view a list of Cisco trademarks, go to this URL: www.cisco.com/go/trademarks. Third-party trademarks mentioned 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. (1110R)

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



Configuring RTP Header Compression

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

Cisco provides two types of header compression: RTP header compression and TCP header compression. This module describes the concepts and tasks related to configuring RTP header compression.



Note

RTP header compression is configured on a per-interface (or subinterface) basis. If you want to configure RTP header compression on a per-class basis, see the "Configuring Class-Based RTP and TCP Header Compression" module.

- [Finding Feature Information, page 7](#)
- [Prerequisites for Configuring RTP Header Compression, page 7](#)
- [Information About Configuring RTP Header Compression, page 8](#)
- [How to Configure RTP Header Compression, page 11](#)
- [Configuration Examples for RTP Header Compression, page 23](#)
- [Additional References, page 27](#)
- [Glossary, page 28](#)
- [Feature Information for Configuring RTP Header Compression, page 29](#)

Finding Feature Information

Your software release may not support all the features documented in this module. For the latest feature information and caveats, see the release notes for your platform and software release. To find information about the features documented in this module, and to see a list of the releases in which each feature is supported, see the Feature Information Table at the end of this document.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to www.cisco.com/go/cfn. An account on Cisco.com is not required.

Prerequisites for Configuring RTP Header Compression

- Before configuring RTP header compression, read the information in the "Header Compression" module.
- You must configure RTP header compression on both ends of the network.

Information About Configuring RTP Header Compression

- [Configurable RTP Header-Compression Settings](#), page 8
- [RTP Header-Compression Keywords](#), page 8
- [Enhanced RTP Header Compression](#), page 9
- [RTP Header Compression over Satellite Links](#), page 10

Configurable RTP Header-Compression Settings

With RTP header compression, you can configure the maximum size of the compressed IP header, the maximum time between transmitting full-header packets, and the maximum number of compressed packets between full headers. These settings are configured using the following three commands:

- **ip header-compression max-header**
- **ip header-compression max-time**
- **ip header-compression max-period**

The **ip header-compression max-header** command allows you to define the maximum size of the IP header of a packet to be compressed. Any packet with an IP header that exceeds the maximum size is sent uncompressed.

The **ip header-compression max-time** command allows you to specify the maximum time between transmitting full-header packets, and the **ip header-compression max-period** command allows you to specify the maximum number of compressed packets between full headers. With the **ip header-compression max-time** and **ip header-compression max-period** commands, the full-header packet is transmitted at the specified time period or when the maximum number of packets is reached, respectively. The counters for both the time period and the number of packets sent are reset after the full-header packet is sent.

For more information about these commands, see the Cisco IOS Quality of Service Solutions Command Reference.

RTP Header-Compression Keywords

When you configure RTP header compression, you can specify the circumstances under which the RTP packets are compressed and the format that is used when the packets are compressed. These circumstances and formats are defined by the following keywords:

- **passive**
- **iphc-format**
- **ietf-format**

These keywords (described below) are available with many of the quality of service (QoS) commands used to configure RTP header compression, such as the **ip rtp header-compression** command. For more information about the **ip rtp header-compression** command, these keywords, and the other QoS commands, see the Cisco IOS Quality of Service Solutions Command Reference.

The **passive** Keyword

By default, the **ip rtp header-compression** command compresses outgoing RTP traffic. If you specify the **passive** keyword, outgoing RTP traffic is compressed only if *incoming* RTP traffic on the *same* interface is compressed. If you do not specify the **passive** keyword, *all* outgoing RTP traffic is compressed.

The **passive** keyword is ignored on PPP interfaces.

The **iphc-format** Keyword

The **iphc-format** keyword indicates that the IP Header Compression (IPHC) format of header compression will be used. For PPP and HDLC interfaces, when the **iphc-format** keyword is specified, TCP header compression is also enabled. Since both RTP and TCP header compression are enabled, both UDP and TCP packets are compressed.

The **iphc-format** keyword includes checking whether the destination port number is even and is in the ranges of 16,385 to 32,767 (for Cisco audio) or 49,152 to 65,535 (for Cisco video). Valid RTP packets that meet the criteria (that is, the port number is even and is within the specified range) are compressed using the compressed RTP packet format. Otherwise, packets are compressed using the less-efficient compressed non-TCP packet format.

The **iphc-format** keyword is not available for interfaces that use Frame Relay encapsulation.



Note

The header compression format (in this case, IPHC) must be the same at *both* ends of the network. That is, if you specify the **iphc-format** keyword on the local router, you must also specify the **iphc-format** keyword on the remote router.

The **ietf-format** Keyword

The **ietf-format** keyword indicates that the Internet Engineering Task Force (IETF) format of header compression will be used. For HDLC interfaces, the **ietf-format** keyword compresses only UDP packets. For PPP interfaces, when the **ietf-format** keyword is specified, TCP header compression is also enabled. Since both RTP header compression and TCP header compression are enabled, both UDP packets and TCP packets are compressed.

With the **ietf-format** keyword, any even destination port number higher than 1024 can be used. Valid RTP packets that meet the criteria (that is, the port number is even and is higher than 1024) are compressed using the compressed RTP packet format. Otherwise, packets are compressed using the less-efficient compressed non-TCP packet format.

The **ietf-format** keyword is not available for interfaces that use Frame Relay encapsulation.



Note

The header compression format (in this case, IETF) must be the same at *both* ends of the network. That is, if you specify the **ietf-format** keyword on the local router, you must also specify the **ietf-format** keyword on the remote router.

Enhanced RTP Header Compression

The Cisco IOS Release 12.3(11)T introduced a feature that enhances the functionality of RTP header compression. This feature is called Enhanced CRTP for Links with High Delay, Packet Loss, and Reordering (ECRTP).

The EC RTP feature is also known as Enhanced RTP Header Compression. It includes modifications and enhancements to RTP header compression to achieve robust operation over unreliable point-to-point links. This is accomplished by repeating updates and sending absolute (uncompressed) values in addition to delta values for selected context parameters.

During compression of an RTP stream, a session context is defined. For each context, the session state is established and shared between the compressor and the decompressor. The context state consists of the full IP/UDP/RTP headers, a few first-order differential values, a link sequence number, a generation number, and a delta encoding table. Once the context state is established, compressed packets may be sent.

RTP header compression was designed for reliable point-to-point links with short delays. It does not perform well over links with a high rate of packet loss, packet reordering, and long delays. Packet loss results in context corruption, and because of long delay, packets are discarded before the context is repaired. To correct the behavior of RTP header compression over such links, several enhancements have been made to the RTP header compression functionality. The enhancements reduce context corruption by changing the way that the compressor updates the context at the decompressor; updates are repeated and include additions to full and differential context parameters.

With these enhancements, RTP header compression performs well over links with packet loss, packet reordering, and long delays.

RTP Header Compression over Satellite Links

The Cisco IOS Release 12.3(2)T introduced a feature called RTP Header Compression over Satellite Links. The RTP Header Compression over Satellite Links feature allows you to use RTP header compression over an asymmetric link (such as a satellite link), where the uplink and downlink connections are on separate interfaces. This feature provides improved system performance by reducing network overhead and speeding up transmission of RTP packets.

- [Periodic Refreshes of a Compressed Packet Stream, page 10](#)
- [Optional Disabling of Context-Status Messages, page 11](#)

Periodic Refreshes of a Compressed Packet Stream

RTP header compression is a mechanism that compresses the IP header in a packet before the packet is transmitted. RTP header compression requires a context status feedback mechanism to recover when the compressed packet stream experiences packet channel loss. If the round-trip time of the packet between the uplink and the downlink is lengthy or if a feedback path does not exist, the chance of loss propagation is greatly increased when a packet is dropped from the link. For instance, if a feedback path does not exist, a compressed packet stream may never recover. This situation presents a need for a configurable option that allows periodic refreshes of the compressed packet stream using full-header packets.

The periodic-refresh Keyword

When you configure header compression, you can configure periodic refreshes of the compressed packet stream using the **periodic-refresh** keyword. The **periodic-refresh** keyword is available with the following commands:

- **ip rtp header-compression**
- **frame-relay ip rtp header-compression**
- **frame-relay map ip rtp header-compression**

For more information about these commands, see the Cisco IOS Quality of Service Solutions Command Reference.

Optional Disabling of Context-Status Messages

During header compression, a session context is defined. For each context, the session state is established and shared between the compressor and the decompressor. The context state consists of the full IP/UDP/RTP headers, a few first-order differential values, a link sequence number, a generation number, and a delta encoding table. This information is included in the context-status messages.

You can disable the sending of context-status messages in instances either when the time it takes for the packet to traverse the uplink and the downlink portions of the data path is greater than the refresh period (in which case, the sending of the context-status message would not be useful) or when a feedback path does not exist.

Disabling the context-status messages can be accomplished by using the **ip header-compression disable-feedback** command. For more information about this command, see the Cisco IOS Quality of Service Solutions Command Reference.

How to Configure RTP Header Compression

- [Enabling RTP Header Compression on an Interface](#), page 11
- [Enabling RTP Header Compression on an Interface That Uses Frame Relay Encapsulation](#), page 13
- [Enabling Enhanced RTP Header Compression](#), page 15
- [Enabling RTP Header Compression over a Satellite Link](#), page 17
- [Specifying the Header-Compression Settings](#), page 18
- [Changing the Number of Header-Compression Connections](#), page 20
- [Displaying Header-Compression Statistics](#), page 22

Enabling RTP Header Compression on an Interface

To enable RTP header compression on an interface, perform the following steps.

**Note**

To enable RTP header compression on an interface that uses Frame Relay encapsulation, skip these steps and complete the steps in the [Enabling RTP Header Compression on an Interface That Uses Frame Relay Encapsulation](#), page 13 instead.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **interface** *type number* [*name-tag*]
4. **encapsulation** *encapsulation-type*
5. **ip address** *ip-address mask* [**secondary**]
6. **ip rtp header-compression** [**passive** | **iphc-format** | **ietf-format**] [**periodic-refresh**]
7. **end**

DETAILED STEPS

Command or Action	Purpose
<p>Step 1 <code>enable</code></p> <p>Example:</p> <pre>Router> enable</pre>	<p>Enables privileged EXEC mode.</p> <ul style="list-style-type: none"> Enter your password if prompted.
<p>Step 2 <code>configure terminal</code></p> <p>Example:</p> <pre>Router# configure terminal</pre>	<p>Enters global configuration mode.</p>
<p>Step 3 <code>interface type number [name-tag]</code></p> <p>Example:</p> <pre>Router(config)# interface serial0</pre>	<p>Configures an interface type and enters interface configuration mode.</p> <ul style="list-style-type: none"> Enter the interface type and the interface number.
<p>Step 4 <code>encapsulation encapsulation-type</code></p> <p>Example:</p> <pre>Router(config-if)# encapsulation ppp</pre>	<p>Sets the encapsulation method used by the interface.</p> <ul style="list-style-type: none"> Enter the encapsulation method.
<p>Step 5 <code>ip address ip-address mask [secondary]</code></p> <p>Example:</p> <pre>Router(config-if)# ip address 209.165.200.225 255.255.255.224</pre>	<p>Sets a primary or secondary IP address for an interface.</p> <ul style="list-style-type: none"> Enter the IP address and mask for the associated IP subnet.
<p>Step 6 <code>ip rtp header-compression [passive iphc-format ietf-format] [periodic-refresh]</code></p> <p>Example:</p> <pre>Router(config-if)# ip rtp header-compression</pre>	<p>Enables RTP header compression.</p>
<p>Step 7 <code>end</code></p> <p>Example:</p> <pre>Router(config-if)# end</pre>	<p>(Optional) Exits interface configuration mode.</p>

Enabling RTP Header Compression on an Interface That Uses Frame Relay Encapsulation

To enable RTP header compression on an interface that uses Frame Relay encapsulation, perform the following steps.



Note

The encapsulation type is specified by using either the **cisco** or **ietf** keyword of the **frame-relay interface-dlci** command. The **cisco** keyword specifies Cisco proprietary encapsulations, and the **ietf** keyword specifies IETF encapsulations. However, note the following points about these keywords:

- Frame Relay interfaces do not support IETF encapsulations when RTP header compression is enabled. Therefore, the **ietf** keyword is not available for Frame Relay interfaces and is not listed in the command syntax shown below.
- The **cisco** keyword is available for use on point-to-point subinterfaces *only*.

>

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **interface** *type number* [*name-tag*]
4. **encapsulation frame-relay**
5. **ip address** *ip-address mask* [**secondary**]
6. **frame-relay interface-dlci** *dlci* [**cisco**]
7. **frame-relay ip rtp header-compression** [**active** | **passive**][**periodic-refresh**]
- 8.
9. **frame-relay map ip** *ip-address dlci* [**broadcast**] **rtpheader-compression** [**active** | **passive**] [**periodic-refresh**] [*connectionsnumber*]
10. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	enable	Enables privileged EXEC mode. <ul style="list-style-type: none"> • Enter your password if prompted.
	Example: Router> enable	

Command or Action	Purpose
<p>Step 2 <code>configure terminal</code></p> <p>Example:</p> <pre>Router# configure terminal</pre>	<p>Enters global configuration mode.</p>
<p>Step 3 <code>interface type number [name-tag]</code></p> <p>Example:</p> <pre>Router(config)# interface serial0</pre>	<p>Configures an interface type and enters interface configuration mode.</p> <ul style="list-style-type: none"> Enter the interface type and the interface number.
<p>Step 4 <code>encapsulation frame-relay</code></p> <p>Example:</p> <pre>Router(config-if)# encapsulation frame-relay</pre>	<p>Enables Frame Relay encapsulation.</p>
<p>Step 5 <code>ip address ip-address mask [secondary]</code></p> <p>Example:</p> <pre>Router(config-if)# ip address 209.165.200.225 255.255.255.224</pre>	<p>Sets a primary or secondary IP address for an interface.</p> <ul style="list-style-type: none"> Enter the IP address and mask for the associated IP subnet.
<p>Step 6 <code>frame-relay interface-dlci dlci [cisco]</code></p> <p>Example:</p> <pre>Router(config-if)# frame-relay interface-dlci 20</pre>	<p>Assigns a data-link connection identifier (DLCI) to a specified Frame Relay interface on the router.</p>
<p>Step 7 <code>frame-relay ip rtp header-compression [active passive] [periodic-refresh]</code></p> <p>Example:</p> <pre>Router(config-if)# frame-relay ip rtp header-compression</pre>	<p>Enables RTP header compression for all Frame Relay maps on a physical interface.</p>
<p>Step 8</p>	

Command or Action	Purpose
<p>Step 9 <code>frame-relay map ip ip-address dlci [broadcast] rtpheader-compression [active passive] [periodic-refresh] [connectionsnumber]</code></p> <p>Example:</p> <pre>Router(config-if)# frame-relay map ip 10.108.175.220 180 rtp header-compression periodic- refresh</pre>	<p>Assigns to an IP map header-compression characteristics that differ from the compression characteristics of the interface with which the IP map is associated.</p> <ul style="list-style-type: none"> Enter the IP address, DLCI number, and any optional keywords and arguments.
<p>Step 10 <code>end</code></p> <p>Example:</p> <pre>Router(config-if)# end</pre>	<p>(Optional) Exits interface configuration mode.</p>

Enabling Enhanced RTP Header Compression

The Enhanced RTP Header Compression feature (also known as ECRTP) includes modifications and enhancements to RTP header compression to achieve robust operation over unreliable point-to-point links. Enhanced RTP header compression is intended for use on networks subject to high rates of packet loss, packet reordering, and long delays. For more information about Enhanced RTP header compression, see the [Enhanced RTP Header Compression, page 9](#).

To enable enhanced RTP header compression, perform the following steps.

- Configure a serial link using HDLC encapsulation or configure an interface using PPP encapsulation.
- Ensure that RTP header compression is enabled on the interface. See the [Enabling RTP Header Compression on an Interface, page 11](#).



Note

Enhanced RTP header compression is not supported on interfaces that use Frame Relay encapsulation.

>

SUMMARY STEPS

- `enable`
- `configure terminal`
- `interface type number [name-tag]`
- `encapsulation encapsulation-type`
- `ip address ip-address mask [secondary]`
- `ip rtp header-compression [passive | iphc-format | ietf-format] [periodic-refresh]`
- `ip header-compression recoverable-loss {dynamic | packet-drops}`
- `end`

DETAILED STEPS

Command or Action	Purpose
<p>Step 1 <code>enable</code></p> <p>Example:</p> <pre>Router> enable</pre>	<p>Enables privileged EXEC mode.</p> <ul style="list-style-type: none"> Enter your password if prompted.
<p>Step 2 <code>configure terminal</code></p> <p>Example:</p> <pre>Router# configure terminal</pre>	<p>Enters global configuration mode.</p>
<p>Step 3 <code>interface type number [name-tag]</code></p> <p>Example:</p> <pre>Router(config)# interface serial0</pre>	<p>Configures an interface type and enters interface configuration mode.</p> <ul style="list-style-type: none"> Enter the interface type and the interface number.
<p>Step 4 <code>encapsulation encapsulation-type</code></p> <p>Example:</p> <pre>Router(config-if)# encapsulation ppp</pre>	<p>Sets the encapsulation method used on the interface.</p> <ul style="list-style-type: none"> Enter the encapsulation method.
<p>Step 5 <code>ip address ip-address mask [secondary]</code></p> <p>Example:</p> <pre>Router(config-if)# ip address 209.165.200.225 255.255.255.224</pre>	<p>Sets a primary or secondary IP address for an interface.</p> <ul style="list-style-type: none"> Enter the IP address and mask for the associated IP subnet.
<p>Step 6 <code>ip rtp header-compression [passive iphc-format ietf-format] [periodic-refresh]</code></p> <p>Example:</p> <pre>Router(config-if)# ip rtp header-compression ietf-format</pre>	<p>Enables RTP header compression.</p>

Command or Action	Purpose
<p>Step 7 <code>ip header-compression recoverable-loss {dynamic packet-drops}</code></p> <p>Example:</p> <pre>Router(config-if)# ip header-compression recoverable-loss dynamic</pre>	<p>Enables ECRTP on an interface.</p> <p>Note Enter the dynamic keyword to enable dynamic packet loss recovery, or enter the <i>packet-drops</i> argument to specify the maximum number of consecutive packet drops that are acceptable.</p>
<p>Step 8 <code>end</code></p> <p>Example:</p> <pre>Router(config-if)# end</pre>	<p>(Optional) Exits interface configuration mode.</p>

Enabling RTP Header Compression over a Satellite Link

To enable RTP header compression over a satellite link, perform the following steps.

SUMMARY STEPS

1. `enable`
2. `configure terminal`
3. `interface type number [name-tag]`
4. `ip address ip-address mask [secondary]`
5. `ip rtp header-compression [passive | iphc-format | ietf-format] [periodic-refresh]`
6. `ip header-compression disable-feedback`
7. `end`

DETAILED STEPS

Command or Action	Purpose
<p>Step 1 <code>enable</code></p> <p>Example:</p> <pre>Router> enable</pre>	<p>Enables privileged EXEC mode.</p> <ul style="list-style-type: none"> • Enter your password if prompted.
<p>Step 2 <code>configure terminal</code></p> <p>Example:</p> <pre>Router# configure terminal</pre>	<p>Enters global configuration mode.</p>

Command or Action	Purpose
<p>Step 3 <code>interface</code> <i>type number</i> [<i>name-tag</i>]</p> <p>Example:</p> <pre>Router(config)# interface serial0</pre>	<p>Configures an interface type and enters interface configuration mode.</p> <ul style="list-style-type: none"> Enter the interface type and the interface number.
<p>Step 4 <code>ip address</code> <i>ip-address mask</i> [secondary]</p> <p>Example:</p> <pre>Router(config-if)# ip address 209.165.200.225 255.255.255.224</pre>	<p>Sets a primary or secondary IP address for an interface.</p> <ul style="list-style-type: none"> Enter the IP address and mask for the associated IP subnet.
<p>Step 5 <code>ip rtp header-compression</code> [passive iphc-format ietf-format] [periodic-refresh]</p> <p>Example:</p> <pre>Router(config-if)# ip rtp header-compression ietf- format periodic-refresh</pre>	<p>Enables RTP header compression.</p> <p>Note For RTP header compression over a satellite link, use the periodic-refresh keyword.</p>
<p>Step 6 <code>ip header-compression disable-feedback</code></p> <p>Example:</p> <pre>Router(config-if)# ip header-compression disable- feedback</pre>	<p>(Optional) Disables the context status feedback messages from the interface or link.</p>
<p>Step 7 <code>end</code></p> <p>Example:</p> <pre>Router(config-if)# end</pre>	<p>(Optional) Exits interface configuration mode.</p>

Specifying the Header-Compression Settings

With RTP header compression, you can configure the maximum size of the compressed IP header, the time period for an automatic resend of full-header packets, and the number of packets transmitted before a new full-header packet is sent.

To specify these header-compression settings, perform the following steps.

SUMMARY STEPS

1. enable
2. configure terminal
3. interface *type number [name-tag]*
4. ip header-compression max-header *max-header-size*
- 5.
6. ip header-compression max-time *length-of-time*
- 7.
8. ip header-compression max-period *number-of-packets*
9. end

DETAILED STEPS

Command or Action	Purpose
<p>Step 1 enable</p> <p>Example:</p> <pre>Router> enable</pre>	<p>Enables privileged EXEC mode.</p> <ul style="list-style-type: none"> • Enter your password if prompted.
<p>Step 2 configure terminal</p> <p>Example:</p> <pre>Router# configure terminal</pre>	<p>Enters global configuration mode.</p>
<p>Step 3 interface <i>type number [name-tag]</i></p> <p>Example:</p> <pre>Router(config)# interface serial0</pre>	<p>Configures an interface type and enters interface configuration mode.</p> <ul style="list-style-type: none"> • Enter the interface type and the interface number.
<p>Step 4 ip header-compression max-header <i>max-header-size</i></p> <p>Example:</p> <pre>Router(config-if)# ip header-compression max-header 100</pre>	<p>Specifies the maximum size of the compressed IP header.</p> <ul style="list-style-type: none"> • Enter the maximum size of the compressed IP header, in bytes.
<p>Step 5</p>	

Command or Action	Purpose
Step 6 <code>ip header-compression max-time</code> <i>length-of-time</i> Example: <pre>Router(config-if)# ip header-compression max-time 30</pre>	Specifies the maximum amount of time to wait before the compressed IP header is refreshed. <ul style="list-style-type: none"> Enter the amount of time, in seconds.
Step 7	
Step 8 <code>ip header-compression max-period</code> <i>number-of-packets</i> Example: <pre>Router(config-if)# ip header-compression max-period 160</pre>	Specifies the maximum number of compressed packets between full headers. <ul style="list-style-type: none"> Enter the maximum number of compressed packets between full headers.
Step 9 <code>end</code> Example: <pre>Router(config-if)# end</pre>	(Optional) Exits interface configuration mode.

Changing the Number of Header-Compression Connections

For PPP and HDLC interfaces, the default is 16 compression connections. For interfaces that use Frame Relay encapsulation, the default is 256 compression connections.

To change the default number of header-compression connections, perform the following steps.

- [Implications of Changing the Number of Header-Compression Connections, page 20](#)

Implications of Changing the Number of Header-Compression Connections

Each header-compression connection sets up a compression cache entry, so you are in effect specifying the maximum number of cache entries and the size of the cache. Too few cache entries for the specified interface can lead to degraded performance, and too many cache entries can lead to wasted memory. Choose the number of header-compression connections according to the network requirements.



Note

Header-Compression Connections on HDLC and Frame Relay Interfaces

For HDLC interfaces and Frame Relay interfaces (that is, interfaces that use Frame Relay encapsulation), the number of header-compression connections on *both sides* of the network must match. That is, the number configured for use on the local router must match the number configured for use on the remote router.

Header-Compression Connections on PPP Interfaces

For PPP interfaces, if the header-compression connection numbers on both sides of the network do not match, the number used is "autonegotiated." That is, any mismatch in the number of header-compression connections between the local router and the remote router will be automatically negotiated to the lower of the two numbers. For example, if the local router is configured to use 128 header-compression connections, and the remote router is configured to use 64 header-compression connections, the negotiated number will be 64.



Note

This autonegotiation function applies to PPP interfaces *only*. For HDLC interfaces and inter Frame Relay encapsulation, no autonegotiation occurs.

>

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **interface** *type number [name-tag]*
4. **ip rtp compression-connections** *number*
- 5.
6. **frame-relay ip rtp compression-connections** *number*
7. **end**

DETAILED STEPS

Command or Action	Purpose
<p>Step 1 enable</p> <p>Example:</p> <pre>Router> enable</pre>	<p>Enables privileged EXEC mode.</p> <ul style="list-style-type: none"> • Enter your password if prompted.
<p>Step 2 configure terminal</p> <p>Example:</p> <pre>Router# configure terminal</pre>	<p>Enters global configuration mode.</p>

Command or Action	Purpose
Step 3 <code>interface type number [name-tag]</code> Example: <pre>Router(config)# interface serial0</pre>	Configures an interface type and enters interface configuration mode. <ul style="list-style-type: none"> Enter the interface type and the interface number.
Step 4 <code>ip rtp compression-connections number</code> Example: <pre>Router(config-if)# ip rtp compression-connections 150</pre>	Specifies the total number of RTP header-compression connections that can exist on an interface. <ul style="list-style-type: none"> Enter the number of compression connections. Note This command can be used for PPP interfaces, HDLC interfaces, or interfaces that use Frame Relay encapsulation.
Step 5	
Step 6 <code>frame-relay ip rtp compression-connections number</code> Example: <pre>Router(config-if)# frame-relay ip rtp compression-connections 150</pre>	Specifies the maximum number of RTP header-compression connections that can exist on a Frame Relay interface (that is, an interface using Frame Relay encapsulation). <ul style="list-style-type: none"> Enter the number of compression connections. Note This command can be used for interfaces that use Frame Relay encapsulation only.
Step 7 <code>end</code> Example: <pre>Router(config-if)# end</pre>	(Optional) Exits interface configuration mode.

Displaying Header-Compression Statistics

You can display header-compression statistics, such as the number of packets sent, received, and compressed, by using either the **show ip rtp header-compression** command or the **show frame-relay ip rtp header-compression** command.

To display header-compression statistics, perform the following steps.

SUMMARY STEPS

1. **enable**
2. **show ip rtp header-compression** [*interface-type interface-number*] [**detail**]
- 3.
4. **show frame-relay ip rtp header-compression** [**interface type number**]
5. **end**

DETAILED STEPS

Command or Action	Purpose
<p>Step 1 <code>enable</code></p> <p>Example:</p> <pre>Router> enable</pre>	<p>Enables privileged EXEC mode.</p> <ul style="list-style-type: none"> • Enter your password if prompted.
<p>Step 2 <code>show ip rtp header-compression [interface-type interface-number] [detail]</code></p> <p>Example:</p> <pre>Router# show ip rtp header-compression</pre> <p>Example:</p>	<p>Displays RTP header-compression statistics for one or all interfaces.</p>
<p>Step 3</p>	
<p>Step 4 <code>show frame-relay ip rtp header-compression [interface type number]</code></p> <p>Example:</p> <pre>Router# show frame-relay ip rtp header-compression</pre>	<p>Displays Frame Relay RTP header-compression statistics for one or all interfaces.</p>
<p>Step 5 <code>end</code></p> <p>Example:</p> <pre>Router# end</pre>	<p>(Optional) Exits privileged EXEC mode.</p>

Configuration Examples for RTP Header Compression

- [Example Enabling RTP Header Compression on an Interface, page 24](#)
- [Example Enabling RTP Header Compression on an Interface That Uses Frame Relay Encapsulation, page 24](#)
- [Example Enabling Enhanced RTP Header Compression, page 25](#)
- [Example Enabling RTP Header Compression over a Satellite Link, page 25](#)
- [Example Specifying the Header-Compression Settings, page 26](#)
- [Example Changing the Number of Header-Compression Connections, page 26](#)
- [Example Displaying Header-Compression Statistic, page 26](#)

Example Enabling RTP Header Compression on an Interface

In the following example, RTP header compression is enabled on serial interface 0.

```
Router> enable

Router# configure terminal

Router(config)# interface serial0

Router(config-if)# encapsulation ppp

Router(config-if)# ip address 209.165.200.225 255.255.255.224

Router(config-if)# ip rtp header-compression

Router(config-if)# end
```

Example Enabling RTP Header Compression on an Interface That Uses Frame Relay Encapsulation

In the following example, RTP header compression is enabled on serial interface 0. Frame Relay encapsulation has been enabled on this interface by using the **encapsulation frame-relay** command.

```
Router> enable

Router# configure terminal

Router(config)# interface serial0

Router(config-if)# encapsulation frame-relay

Router(config-if)# ip address 209.165.200.225 255.255.255.224

Router(config-if)# frame-relay interface-dlci 20

Router(config-if)# frame-relay ip rtp header-compression

Router(config-if)# end
```

Example Enabling Enhanced RTP Header Compression

In the following example, EC RTP is enabled on serial interface 0. PPP encapsulation is enabled on the interface (a prerequisite for configuring EC RTP on a serial interface). Also, dynamic loss recovery has been specified by using the **dynamic** keyword of the **ip header-compression recoverable-loss** command.

```
Router> enable

Router# configure terminal

Router(config)# interface serial0

Router(config-if)# encapsulation ppp

Router(config-if)# ip address 209.165.200.225 255.255.255.224

Router(config-if)# ip rtp header-compression ietf-format

Router(config-if)# ip header-compression recoverable-loss dynamic

Router(config-if)# end
```

Example Enabling RTP Header Compression over a Satellite Link

In the following example, RTP header compression is enabled on the serial interface 0. In this example, serial interface 0 is a satellite link in the network topology. The **periodic-refresh** keyword has been specified, which means that the compressed IP header will be refreshed periodically. Also, the context-status messages have been turned off (disabled).

```
Router> enable

Router# configure terminal

Router(config)# interface serial0

Router(config-if)# ip address 209.165.200.225 255.255.255.224

Router(config-if)# ip rtp header-compression ietf-format periodic-refresh

Router(config-if)# ip header-compression disable-feedback

Router(config-if)# end
```

Example Specifying the Header-Compression Settings

In the following example, the maximum size of the compressed IP header (100 bytes) has been specified by using the **ip header-compression max-header** command.

```
Router> enable

Router# configure terminal

Router(config)# interface serial0

Router(config-if)# ip header-compression max-header 100

Router(config-if)# end
```

Example Changing the Number of Header-Compression Connections

In the following example, the number of header-compression connections has been changed to 150 by using the **ip rtp compression-connections** command.

```
Router> enable

Router# configure terminal

Router(config)# interface serial0

Router(config-if)# ip rtp compression-connections 150

Router(config-if)# end
```

Example Displaying Header-Compression Statistic

You can use the **show ip rtp header-compression** command to display header-compression statistics such as the number of packets received, sent, and compressed. The following is sample output from the **show ip rtp header-compression** command. In this example, EC RTP has been enabled on serial interface 0.

```
Router# show ip rtp header-compression serial0
RTP/UDP/IP header compression statistics:
Interface Serial0 (compression on, IETF, EC RTP)
  Rcvd:   1473 total, 1452 compressed, 0 errors, 0 status msgs
         0 dropped, 0 buffer copies, 0 buffer failures
  Sent:   1234 total, 1216 compressed, 0 status msgs, 379 not predicted
         41995 bytes saved, 24755 bytes sent
         2.69 efficiency improvement factor
  Connect: 16 rx slots, 16 tx slots,
          6 misses, 0 collisions, 0 negative cache hits, 13 free contexts
          99% hit ratio, five minute miss rate 0 misses/sec, 0 max
```

Additional References

The following sections provide references related to configuring RTP header compression.

Related Documents

Related Topic	Document Title
Cisco IOS commands	Cisco IOS Master Commands List, All Releases
QoS commands: complete command syntax, command modes, command history, defaults, usage guidelines, and examples	<i>Cisco IOS Quality of Service Solutions Command Reference</i>
Frame Relay	"Frame Relay Queueing and Fragmentation at the Interface" module
Header compression overview	"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 2507	<i>IP Header Compression</i>

RFC	Title
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>

Technical Assistance

Description	Link
The Cisco Support and Documentation website provides online resources to download documentation, software, and tools. Use these resources to install and configure the software and to troubleshoot and resolve technical issues with Cisco products and technologies. Access to most tools on the Cisco Support and Documentation website requires a Cisco.com user ID and password.	http://www.cisco.com/cisco/web/support/index.html

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.

context --The state that the compressor uses to compress a header and that the decompressor uses to decompress a header. The context is the uncompressed version of the last header sent and includes other information used to compress and decompress the packet.

context-state packet --A special packet sent from the decompressor to the compressor to communicate a list of (TCP or NON_TCP/RTP) context identifiers (CIDs) for which synchronization has been lost. This packet is sent only over a single link, so it requires no IP header.

DLCI --data-link connection identifier. A value that specifies a permanent virtual circuit (PVC) or switched virtual circuit (SVC) in a Frame Relay network. In the basic Frame Relay specification, DLCIs are locally significant (connected devices might use different values to specify the same connection). In the Local Management Interface (LMI) extended specification, DLCIs are globally significant (DLCIs specify individual end devices).

ECRTP --Enhanced Compressed Real-Time Transport Protocol. A compression protocol that is designed for unreliable point-to-point links with long delays.

encapsulation --A method of wrapping data in a particular protocol header. For example, Ethernet data is wrapped in a specific Ethernet header before network transit. Also, when dissimilar networks are bridged, the entire frame from one network is simply placed in the header used by the data link layer protocol of the other network.

full header (header refresh) --An uncompressed header that updates or refreshes the context for a packet stream. It carries a CID that will be used to identify the context. Full headers for non-TCP packet streams also carry the generation of the context that they update or refresh.

HDLC --High-Level Data Link Control. A bit-oriented synchronous data link layer protocol developed by the 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.

IETF --Internet Engineering Task Force. A task force that consists of over 80 working groups responsible for developing Internet standards.

IPHC --IP Header Compression. A protocol capable of compressing both TCP and UDP headers.

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

lossy serial links --Links in a network that are prone to lose packets.

packet stream --The sequence of packets whose headers are similar and share context. For example, headers in an RTP packet stream have the same source and final destination address and the same port numbers in the RTP header.

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.

Feature Information for Configuring RTP Header Compression

The following table provides release information about the feature or features described in this module. This table lists only the software release that introduced support for a given feature in a given software release train. Unless noted otherwise, subsequent releases of that software release train also support that feature.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to www.cisco.com/go/cfn. An account on Cisco.com is not required.

Table 1 **Feature Information for Configuring RTP Header Compression**

Feature Name	Releases	Feature Information
RTP Header Compression over Satellite Links	12.3(2)T	The RTP Header Compression over Satellite Links feature allows customers to use RTP header compression over an asymmetric link (such as a satellite link), where the uplink and downlink connections are on separate interfaces.
Enhanced CRTP for Links with High Delay, Packet Loss and Reordering	12.3(11)T	The Enhanced Compressed Real-Time Transport Protocol (ECRTP) for Links with High Delay, Packet Loss, and Reordering feature includes modifications and enhancements to CRTP to achieve robust operation over unreliable point-to-point links. This is accomplished by repeating updates and sending absolute (uncompressed) values in addition to delta values for selected context parameters.
RTP Header Compression RTP Header Compression over Satellite Links	15.0(1)S	The RTP Header Compression and RTP Header Compression over Satellite Links features were integrated into the Cisco IOS Release 15.0(1)S release.

Cisco and the Cisco logo are trademarks or registered trademarks of Cisco and/or its affiliates in the U.S. and other countries. To view a list of Cisco trademarks, go to this URL: www.cisco.com/go/trademarks. Third-party trademarks mentioned 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. (1110R)

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