QoS: Header Compression Configuration Guide, Cisco IOS XE Fuji 16.8.x

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

CONTENTS

CHAPTER 1

Read Me First

Important Information about Cisco IOS XE 16

Effective Cisco IOS XE Release 3.7.0E (for Catalyst Switching) and Cisco IOS XE Release 3.17S (for Access and Edge Routing) the two releases evolve (merge) into a single version of converged release—the Cisco IOS XE 16—providing one release covering the extensive range of access and edge products in the Switching and Routing portfolio.

Feature Information

Use Cisco Feature Navigator to find information about feature support, platform support, and Cisco software image support. An account on Cisco.com is not required.

Related References

- Cisco IOS Command References, All Releases

Obtaining Documentation and Submitting a Service Request

For information on obtaining documentation, using the Cisco Bug Search Tool (BST), submitting a service request, and gathering additional information, see What's New in Cisco Product Documentation.

To receive new and revised Cisco technical content directly to your desktop, you can subscribe to the What's New in Cisco Product Documentation RSS feed. RSS feeds are a free service.

CHAPTER 2

Header Compression

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.

Finding Feature Information

Your software release may not support all the features documented in this module. For the latest caveats and feature information, see Bug Search Tool and 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 module.

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

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.

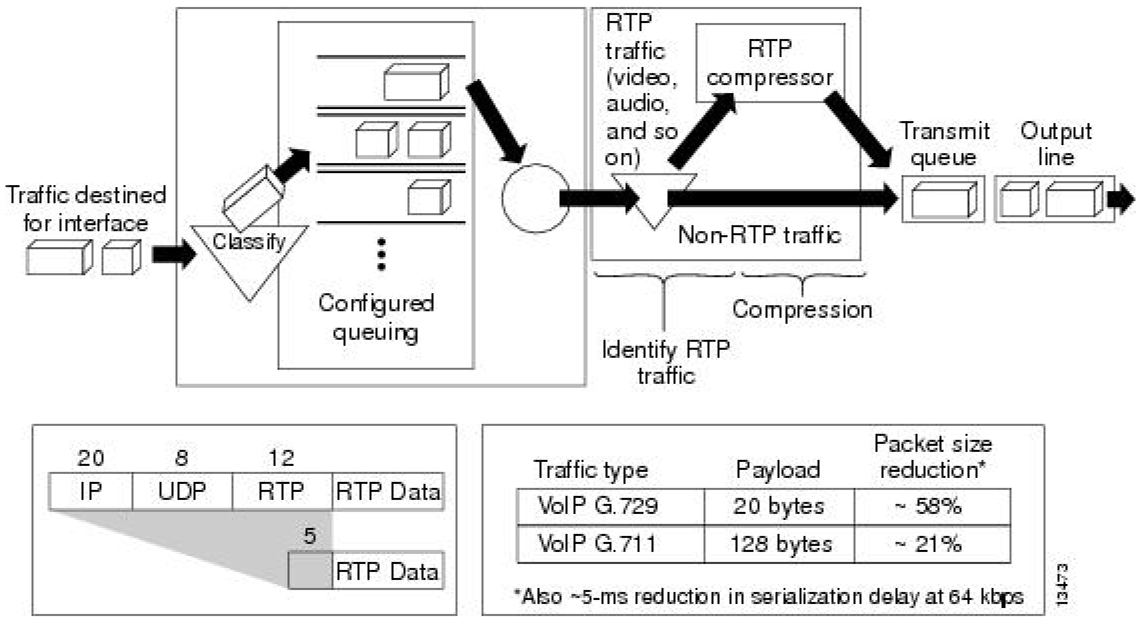
To avoid the unnecessary consumption of available bandwidth, RTP header compression is used on a link-by-link basis.

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. 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

Related Documents

Related Topic	Document Title
Cisco IOS commands	Cisco IOS Master Commands List, All Releases
QoS commands	*Cisco IOS QoS Command Reference*
MQC	"Applying QoS Features Using the MQC"
RTP header compression	"Configuring RTP Header Compression"

Standards and RFCs

Standard/RFC	Title
No new or modified standards are supported, and support for existing standards has not been modified.	--
• RFC 1144 • RFC 2507 • RFC 2508 • RFC 3544 • RFC 3550	• Compressing TCP/IP Headers for Low-Speed Serial Links • IP Header Compression • Compressing IP/UDP/RTP Headers for Low-Speed Serial Links • IP Header Compression over PPP • A Transport Protocol for Real-Time Applications

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 software releases, and feature sets, use Cisco MIB Locator found at the following URL: http://www.cisco.com/go/mibs

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.

CHAPTER 3

Configuring RTP Header Compression

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

Finding Feature Information

Your software release may not support all the features documented in this module. For the latest caveats and feature information, see Bug Search Tool and 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 module.

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

With RTP header compression, you can configure the maximum size of the compressed 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 **ipheader-compressionmax-header** command allows you to define the maximum size of the header of a packet to be compressed. Any packet with an header that exceeds the maximum size is sent uncompressed.

The **ipheader-compressionmax-time**command allows you to specify the maximum time between transmitting full-header packets, and the **ipheader-compressionmax-period** command allows you to specify the maximum number of compressed packets between full headers. With the **ipheader-compressionmax-time** and **ipheader-compressionmax-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**
- cisco

These keywords (described below) are available with many of the quality of service (QoS) commands used to configure RTP header compression, such as the **iprtpheader-compression** command. For more information about the **iprtpheader-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 **iprtpheader-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 both TCP and UDP packets. UPD and TCP packets are compressed separately. 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.

The cisco Keyword

The **cisco** keyword indicates that the Cisco-propriety or "original" format of header compression will be used.

RTP header-compression using the cisco format supports even-numbered UDP destination ports in the Cisco audio range of 16384 to 32767 or in the video range of 49152 to 65535.

The **cisco** keyword is only available on interfaces that use Frame Relay or HDLC encapsulation.

How to Configure RTP Header Compression

Enabling RTP Header Compression on an Interface

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

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**| **cisco**] [**periodic-refresh**]
7. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**interface** *type number* [*name-tag*] **Example:** `Router(config)# interface serial0`	Configures an interface type and enters interface configuration mode. • Enter the interface type and the interface number.
Step 4	**encapsulation** *encapsulation-type* **Example:** `Router(config-if)# encapsulation ppp`	Sets the encapsulation method used by the interface. • Enter the encapsulation method.
Step 5	**ip address** *ip-address mask* [**secondary**]	Sets a primary or secondary IP address for an interface.

	Command or Action	Purpose
	Example: `Router(config-if)# ip address 209.165.200.225 255.255.255.224`	• Enter the IP address and mask for the associated IP subnet.
Step 6	**ip rtp header-compression** [**passive** | **iphc-format** | **ietf-format**| **cisco**] [**periodic-refresh**] **Example:** `Router(config-if)# ip rtp header-compression`	Enables RTP header compression.
Step 7	**end** **Example:** `Router(config-if)# end`	(Optional) Exits interface configuration mode.

Specifying the Header-Compression Settings

With RTP header compression, you can configure the maximum size of the compressed 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
Step 1	**enable**	Enables privileged EXEC mode.

	Command or Action	Purpose
	Example: `Router> enable`	• Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**interface** *type number* [*name-tag*] **Example:** `Router(config)# interface serial0`	Configures an interface type and enters interface configuration mode. • Enter the interface type and the interface number.
Step 4	**ip header-compression max-header** *max-header-size* **Example:** `Router(config-if)# ip header-compression max-header 100`	Specifies the maximum size of the compressed IP header. • Enter the maximum size of the compressed IP header, in bytes.
Step 5		
Step 6	**ip header-compression max-time** *length-of-time* **Example:** `Router(config-if)# ip header-compression max-time 30`	Specifies the maximum amount of time to wait before the compressed IP header is refreshed. • Enter the amount of time, in seconds.
Step 7		
Step 8	**ip header-compression max-period** *number-of-packets* **Example:** `Router(config-if)# ip header-compression max-period 160`	Specifies the maximum number of compressed packets between full headers. • Enter the maximum number of compressed packets between full headers.
Step 9	**end** **Example:** `Router(config-if)# end`	(Optional) Exits interface configuration mode.

Changing the Number of Header-Compression Connections

For PPP and HDLC interfaces, the default is 16 compression connections.

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

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 Interfaces**

For HDLC interfaces, 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, no autonegot occurs.

>

SUMMARY STEPS

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

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable**	Enables privileged EXEC mode.

	Command or Action	Purpose
	Example: `Router> enable`	• Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**interface** *type number* [*name-tag*] **Example:** `Router(config)# interface serial0`	Configures an interface type and enters interface configuration mode. • Enter the interface type and the interface number.
Step 4	**ip rtp compression-connections** *number* **Example:** `Router(config-if)# ip rtp compression-connections 150`	Specifies the total number of RTP header-compression connections that can exist on an interface. • Enter the number of compression connections. **Note** This command can be used for PPP interfaces and HDLC interfaces.
Step 5	**end** **Example:** `Router(config-if)# end`	(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 the **showiprtpheader-compression** command.

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

SUMMARY STEPS

1. **enable**
2. **show ip rtp header-compression** [*interface-typeinterface-number*]
3. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**show ip rtp header-compression** [*interface-typeinterface-number*] **Example:** `Router# show ip rtp header-compression` **Example:**	Displays RTP header-compression statistics for one or all interfaces.
Step 3	**end** **Example:** `Router# end`	(Optional) Exits privileged EXEC mode.

Configuration Examples for RTP Header Compression

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 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 **ipheader-compressionmax-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 Statistics

You can use the **showiprtpheader-compression**command to display header-compression statistics such as the number of packets received, sent, and compressed. The following is sample output from the **showiprtpheader-compression** command.

```
Router# show ip rtp header-compression
 serial0
RTP/UDP/IP header compression statistics:
  Interface Serial0 (compression on, IETF)
    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	*Cisco IOS Quality of Service Solutions Command Reference*
Header compression overview	"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 2507	*IP Header Compression*
RFC 2508	*Compressing IP/UDP/RTP Headers for Low-Speed Serial Links*
RFC 3544	*IP Header Compression over PPP*

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

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
Express RTP and TCP Header Compression	Cisco IOS XE Release 2.1	This feature was introduced on Cisco ASR 1000 Series Routers.
RTP Header Compression	Cisco IOS XE Release 2.1	This feature was introduced on Cisco ASR 1000 Series Routers.

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).

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.