

# Understanding Compression (Including cRTP) and Quality of Service

Document ID: 22308

## Contents

### Introduction

#### Prerequisites

- Requirements
- Components Used
- Conventions

#### Data Compression Overview

#### Cisco Compression Hardware

#### Fancy Queuing and Hardware Compression

#### Fancy Queuing and Software Compression

#### RTP Header Compression and QoS

#### Known Issues

#### Related Information

## Introduction

This document reviews known issues with enabling the Cisco IOS<sup>®</sup> software features of compression and Quality of Service (QoS) on the same router.

Cisco IOS software offers many features that optimize wide-area network (WAN) links to ease the WAN bandwidth bottleneck. Compression is an effective optimization method and includes two types:

- **Data Compression** – Provides each end with a coding scheme that allows characters to be removed from the frames at the sending side of the link, and then correctly replaces them at the receiving side. Since the condensed frames occupy less bandwidth, greater numbers can be transmitted per unit of time. Examples of data compression schemes include STAC, Microsoft Point-to-Point Compression (MPPC), and Frame Relay Forum 9 (FRF.9).
- **Header Compression** – Compresses a header at various layers of the Open System Interconnection (OSI) reference model. Examples include Transmission Control Protocol (TCP) header compression, compressed RTP (cRTP), and compressed Internet Protocol/User Datagram Protocol (IP/UDP).

## Prerequisites

### Requirements

There are no specific requirements for this document.

### Components Used

This document is not restricted to specific software and hardware versions.

The information presented in this document was created from devices in a specific lab environment. All of the devices used in this document started with a cleared (default) configuration. If you are working in a live network, ensure that you understand the potential impact of any command before using it.

## Conventions

For more information on document conventions, refer to Cisco Technical Tips Conventions.

## Data Compression Overview

The basic function of data compression is to reduce the size of a data frame transmitted over a network link. Reducing the size of the frame reduces the time required to transmit the frame across the network.

The two most commonly used data compression algorithms on internetworking devices are Stacker and Predictor.

The following sample configurations show two ways of enabling payload compression on a Frame Relay interface or subinterface.

```
interface Serial0/5
 ip address 10.0.0.1 255.255.255.0
 no ip directed-broadcast
 encapsulation frame-relay IETF
 clockrate 1300000
 frame-relay map ip 10.0.0.2 16 broadcast IETF payload-compression FRF9 stac

interface Serial0/0.105 point-to-point
 ip address 192.168.162.1 255.255.255.0
 no ip directed-broadcast
 frame-relay interface-dlci 105 IETF
 class 128k
 frame-relay payload-compression FRF9 stac
```

## Cisco Compression Hardware

Hardware-assisted data compression achieves the same overall functionality as software-based data compression, but accelerates compression rates by offloading this computationally from the main CPU. In other words:

- Software Compression – Compression is implemented in the Cisco IOS software installed in the router's main processor.
- Hardware Compression – Compression is implemented in the compression hardware installed in a system slot. Hardware compression removes compression and decompression responsibilities from the main processor installed in your router.

The following table lists Cisco compression hardware and supported platforms:

Compression Hardware	Supported Platforms	Notes
SA-Comp/1 and SA-Comp/4 service adapters (CSA)	Cisco 7200 Series Routers and the second-generation Versatile Interface Processor (VIP2) in the Cisco 7000 and 7500 Series Routers	Supports Stacker algorithm over serial interfaces configured with Point-to-Point Protocol (PPP) or Frame Relay encapsulation.
NM-COMPR		

	Cisco 3600 Series Routers	Supports Stacker algorithm over PPP links and Frame Relay links with the FRF.9 compression algorithm.
AIM-COMPR4	Cisco 3660 Series Routers only	Supports Lempel-Ziv Standard (LZS) and MPPC algorithms.

Configuring compression on a serial interface with a command such as **compress stac** automatically enables hardware compression if it is available. Otherwise, software compression is enabled. You can use the **compress stac software** command to force the use of software compression.

## Fancy Queueing and Hardware Compression

This section discusses a resolved issue with the Cisco legacy priority queueing (PQ) feature and compression hardware. Compression hardware originally dequeued packets aggressively from the PQs, effectively removing the benefits of PQ. In other words, PQ worked properly, but queueing functionally moved to the compression hardware's own queues (holdq, hw ring and compQ), which are strictly first-in, first-out (FIFO). The symptoms of this problem are documented in Cisco bug ID CSCdp33759 (marked as a duplicate of CSCdm91180).

The resolution modifies the compression hardware's driver. Specifically, it throttles the rate at which the compression hardware dequeues packets by reducing the size of the hardware queues based on the interface's bandwidth. This back pressure mechanism ensures that packets stay in the fancy queues instead of being held in the compression hardware's queues. Refer to the following bug IDs for more information:

**Note:** More information on these bug IDs can be found by using the Bug Toolkit (registered customers only).

- CSCdm91180 – Applies to Frame Relay encapsulation and the Compression Service Adapter (CSA).
- CSCdp33759 (and CSCdr18251) – Applies to PPP encapsulation and the CSA.
- CSCdr18251 – Applies to PPP encapsulation and the asynchronous interface module-compression (AIM-COMPR).

The hardware-level queues of the Cisco 3660 compression can be seen in the following sample output of the **show pas caim stats** command. If the hardware compression queues are storing many packets, a packet dequeued from the PQ waits at the tail end of this queue, and thus experiences delay.

```
Router> show pas caim stats 0

CompressionAim0
ds:0x80F56A44 idb:0x80F50DB8
  422074 uncomp paks in -->      422076 comp paks out
  422071 comp paks in -->      422075 uncomp paks out
633912308 uncomp bytes in -->  22791798 comp bytes out
 27433911 comp bytes in -->    633911762 uncomp bytes out
   974 uncomp paks/sec in -->    974 comp paks/sec out
   974 comp paks/sec in -->    974 uncomp paks/sec out
 11739116 uncomp bits/sec in --> 422070 comp bits/sec out
  508035 comp bits/sec in --> 11739106 uncomp bits/sec out
433 seconds since last clear
holdq: 0 hw_enable: 1 src_limited: 0 num_cnxts: 4
```

```
no data: 0 drops: 0 nobuffers: 0 enc adj errs: 0 fallbacks: 0
no Replace: 0 num seq errs: 0 num desc errs: 0 cmds complete: 844151
Bad reqs: 0 Dead cnxts: 0 No Paks: 0 enq errs: 0
rx pkt drops: 0 tx pkt drops: 0 >dequeues: 0 requeues: 0
drops disabled: 0 clears: 0 ints: 844314 purges: 0
no cnxts: 0 bad algos: 0 no crams: 0 bad paks: 0
# opens: 0 # closes: 0 # hangs: 0
```

**Note:** CSCdr86700 removes the changes implemented in CSCdm91180 from platforms not supporting a CSA.

In addition, while troubleshooting this problem, packet-expansion issues with small packets (around 4 bytes) and particular repetitive patterns, such as Cisco pings with a pattern of 0xABCDABCD, were resolved with bug ID CSCdm11401. Small packets are less likely to be related to other packets in the stream, and attempting to compress them may result in expanded packets, or cause dictionary resets. The root cause is a problem with the chip used on the CSA. Cisco bug ID CSCdp64837 resolves this problem by changing the FRF.9 compression code to avoid compressing packets having less than 60 bytes of payload.

## Fancy Queueing and Software Compression

In contrast to hardware compression, software compression and fancy queueing, including custom, priority, and weighted fair queueing, are not supported on interfaces configured with PPP encapsulation. This limitation is documented in bug IDs CSCdj45401 and CSCdk86833.

The reason for the limitation is that PPP compression is not stateless and maintains a compression history over the data stream to optimize the compression ratios. The compressed packets must be kept in order to maintain compression history. If packets are compressed before queueing, they must be put in a single queue. Putting them in different queues, as custom and priority queueing do, may lead to the packets arriving out of sequence, which breaks compression. Alternative solutions are sub-optimal and have not been implemented. Such alternatives include compressing packets as they are dequeued (unacceptable for performance reasons), maintaining a separate compression history for each queue (unsupported and involving significant overhead), and resetting the compression history for every packet (substantially impacts compression ratios). As a workaround, you can configure high-level data link control (HDLC) encapsulation, but this configuration may affect system performance and is not recommended. Instead, use hardware compression.

## RTP Header Compression and QoS

RFC 1889 [↗](#) specifies the RTP, which manages the audio path transport for Voice over IP (VoIP). RTP provides such services as sequencing to identify lost packets and 32-bit values to identify and distinguish between multiple senders in a multicast stream. Importantly, it does not provide or ensure QoS.

VoIP packets are composed of one or more speech codec samples or frames encapsulated in 40 bytes of IP/UDP/RTP headers. 40 bytes is a relatively large amount of overhead for the typical 20-byte VoIP payloads, particularly over low-speed links. RFC 2508 [↗](#) specifies compressed RTP (cRTP), which is designed to reduce the IP/UDP/RTP headers to two bytes for most packets in the case where no UDP checksums are being sent, or four bytes with checksums. The compression algorithm defined in this document draws heavily upon the design of TCP/IP header compression as described in RFC 1144 [↗](#).

RFC 2508 actually specifies two formats of cRTP:

- **Compressed RTP (CR)** – Used when the IP, UDP, and RTP headers remain consistent. All three headers are compressed.
- **Compressed UDP (CU)** – Used when there is a large change in the RTP timestamp or when the RTP payload type changes. The IP and UDP headers are compressed, but the RTP header is not.

Cisco IOS software release 12.1(5)T introduced several improvements for compression over Frame Relay permanent virtual circuits (PVCs) on the Cisco 2600, 3600, and 7200 Series Routers. These improvements include the following:

Before Cisco IOS Release 12.1(5)T	Cisco IOS Releases 12.1(5)T and 12.2
Slow-speed WAN edge fragmentation methods needed to ensure voice quality did not work on interfaces with hardware compression. These fragmentation methods, which include MLPPP/LFI, FRF.11 Annex C, and FRF.12, do work with software-based compression.	Fragmentation (FRF.12 or Link Fragmentation and Interleaving (LFI)) are supported together with hardware compression. In addition, FRF.12 and FRF.11 Annex-C Fragmentation are supported with
FRF.9 compressions is supported only on IETF-encap PVCs	FRF.9 hardware compression on the cRTP and FRF.9 compression are supported on the same PVC. Voice packets from the priority queue with low latency compression is supported on PVCs queuing (LLQ) bypass the FRF.9 configured with Cisco and Internet compressor engine. Data packets Engineering Task Force (IETF) encapsulation.
cRTP is supported on Frame Relay PVCs configured with Cisco encapsulation only.	cRTP continues to be supported only on Cisco-encapsulated PVCs.

## Known Issues

The following table lists known issues with cRTP and Cisco IOS QoS features. This list is accurate at the time of publishing. Also refer to the the Release Notes for your version of Cisco IOS software for more information.

Bug ID	Description
CSCdv73543	<p>When a hierarchical QoS policy, using the commands of the modular QoS CLI, is applied to an outbound interface and specifies a two-level policer, the conformed traffic rate may be less than expected. The problem occurs when the action taken on the packet in one level is different from that in the second level. For example, conform at the first level and exceed at the second level. An example policy is illustrated below:</p> <pre> policy-map test-policer   class class-default     police 10000 1500 1500 conform-action transmit exceed-action transmit   service-policy inner-police !</pre>

	<pre> policy-map inner-police   class prec5     police 20000 1500 1500 conform-action     transmit exceed-action transmit </pre>
CSCdt52094	<p>Unexpected packet drops may be seen when using low latency queueing (LLQ) over Frame Relay. The problem was caused by the queueing system not taking the bandwidth gains of cRTP into account.</p>
CSCds43465	<p>Originally, cRTP happened after queueing. The result was that queueing (potentially) saw a much larger packet than what actually was transmitted on the wire. This behavior is changed with this bug. Queueing now considers compressed packets. With this change, you can configure <b>bandwidth</b> statements with CBWFQ based on compressed data rates.</p>

## Related Information

- [QoS Support Page](#)
- [Technical Support – Cisco Systems](#)

---

[Contacts & Feedback](#) | [Help](#) | [Site Map](#)

© 2014 – 2015 Cisco Systems, Inc. All rights reserved. [Terms & Conditions](#) | [Privacy Statement](#) | [Cookie Policy](#) | [Trademarks of Cisco Systems, Inc.](#)

---

Updated: Feb 15, 2008

Document ID: 22308

---