

Network Working Group
Request for Comments: 4629
Obsoletes: 2429
Updates: 3555
Category: Standards Track

J. Ott
Helsinki University of Technology
C. Bormann
Universitaet Bremen TZI
G. Sullivan
Microsoft
S. Wenger
Nokia
R. Even, Ed.
Polycom
January 2007

RTP Payload Format for ITU-T Rec. H.263 Video

Status of This Memo

This document specifies an Internet standards track protocol for the Internet community, and requests discussion and suggestions for improvements. Please refer to the current edition of the "Internet Official Protocol Standards" (STD 1) for the standardization state and status of this protocol. Distribution of this memo is unlimited.

Copyright Notice

Copyright (C) The IETF Trust (2007).

Abstract

This document describes a scheme to packetize an H.263 video stream for transport using the Real-time Transport Protocol (RTP) with any of the underlying protocols that carry RTP.

The document also describes the syntax and semantics of the Session Description Protocol (SDP) parameters needed to support the H.263 video codec.

The document obsoletes RFC 2429 and updates the H263-1998 and H263-2000 media type in RFC 3555.

Table of Contents

1. Introduction	3
1.1. Terminology	3
2. New H.263 Features	3
3. Usage of RTP	4
3.1. RTP Header Usage	5
3.2. Video Packet Structure	6
4. Design Considerations	7
5. H.263+ Payload Header	9
5.1. General H.263+ Payload Header	9
5.2. Video Redundancy Coding Header Extension	10
6. Packetization Schemes	12
6.1. Picture Segment Packets and Sequence Ending Packets (P=1)	12
6.1.1. Packets that begin with a Picture Start Code	12
6.1.2. Packets that begin with GBSC or SSC	13
6.1.3. Packets that begin with an EOS or EOSBS Code	14
6.2. Encapsulating Follow-on Packet (P=0)	15
7. Use of this Payload Specification	15
8. Media Type Definition	17
8.1. Media Type Registrations	17
8.1.1. Registration of Media Type video/H263-1998	17
8.1.2. Registration of Media Type video/H263-2000	21
8.2. SDP Usage	22
8.2.1. Usage with the SDP Offer Answer Model	23
9. Backward Compatibility to RFC 2429	25
9.1. New Optional Parameters for SDP	25
10. IANA Considerations	25
11. Security Considerations	25
12. Acknowledgments	26
13. Changes from Previous Versions of the Documents	26
13.1. Changes from RFC 2429	26
13.2. Changes from RFC 3555	26
14. References	26
14.1. Normative References	26
14.2. Informative References	27

1. Introduction

This document specifies an RTP payload header format applicable to the transmission of video streams based on the 1998 and 2000 versions of International Telecommunication Union-Telecommunication Standardization Sector (ITU-T) Recommendation H.263 [H263]. Because the 1998 and 2000 versions of H.263 are a superset of the 1996 syntax, this format can also be used with the 1996 version of H.263 and is recommended for this use by new implementations. This format replaces the payload format in RFC 2190 [RFC2190], which continues to be used by some existing implementations, and can be useful for backward compatibility. New implementations supporting H.263 SHALL use the payload format described in this document. RFC 2190 is moved to historic status [RFC4628].

The document updates the media type registration that was previously in RFC 3555 [RFC3555].

This document obsoletes RFC 2429 [RFC2429].

1.1. Terminology

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in RFC 2119 [RFC2119] and indicate requirement levels for compliant RTP implementations.

2. New H.263 Features

The 1998 version of ITU-T Recommendation H.263 added numerous coding options to improve codec performance over the 1996 version. In this document, the 1998 version is referred to as H.263+ and the 2000 version as H.263++.

Among the new options, the ones with the biggest impact on the RTP payload specification and the error resilience of the video content are the slice structured mode, the independent segment decoding mode, the reference picture selection mode, and the scalability mode. This section summarizes the impact of these new coding options on packetization. Refer to [H263] for more information on coding options.

The slice structured mode was added to H.263+ for three purposes: to provide enhanced error resilience capability, to make the bitstream more amenable for use with an underlying packet transport such as RTP, and to minimize video delay. The slice structured mode supports fragmentation at macroblock boundaries.

With the independent segment decoding (ISD) option, a video picture frame is broken into segments and encoded in such a way that each segment is independently decodable. Utilizing ISD in a lossy network environment helps to prevent the propagation of errors from one segment of the picture to others.

The reference picture selection mode allows the use of an older reference picture rather than the one immediately preceding the current picture. Usually, the last transmitted frame is implicitly used as the reference picture for inter-frame prediction. If the reference picture selection mode is used, the data stream carries information on what reference frame should be used, indicated by the temporal reference as an ID for that reference frame. The reference picture selection mode may be used with or without a back channel, which provides information to the encoder about the internal status of the decoder. However, no special provision is made herein for carrying back channel information. The Extended RTP Profile for RTP Control Protocol (RTCP)-based Feedback [RFC4585] MAY be used as a back channel mechanism.

H.263+ also includes bitstream scalability as an optional coding mode. Three kinds of scalability are defined: temporal, signal-to-noise ratio (SNR), and spatial scalability. Temporal scalability is achieved via the disposable nature of bi-directionally predicted frames, or B-frames. (A low-delay form of temporal scalability known as P-picture temporal scalability can also be achieved by using the reference picture selection mode, described in the previous paragraph.) SNR scalability permits refinement of encoded video frames, thereby improving the quality (or SNR). Spatial scalability is similar to SNR scalability except that the refinement layer is twice the size of the base layer in the horizontal dimension, vertical dimension, or both.

H.263++ added some new functionalities. Among the new functionalities are support for interlace mode, specified in H.263, annex W.6.3.11, and the definition of profiles and levels in H.263 annex X.

3. Usage of RTP

When transmitting H.263+ video streams over the Internet, the output of the encoder can be packetized directly. All the bits resulting from the bitstream (including the fixed length codes and variable length codes) will be included in the packet, the only exception being that when the payload of a packet begins with a Picture, GOB, Slice, End of Sequence (EOS), or End of Sub-Bit Stream (EOSBS) start code, the first 2 (all-zero) bytes of the start code shall be removed and replaced by setting an indicator bit in the payload header.

For H.263+ bitstreams coded with temporal, spatial, or SNR scalability, each layer may be transported to a different network address. More specifically, each layer may use a unique IP address and port number combination. The temporal relations between layers shall be expressed using the RTP timestamp so that they can be synchronized at the receiving ends in multicast or unicast applications.

The H.263+ video stream will be carried as payload data within RTP packets. A new H.263+ payload header is defined in Section 5; it updates the one specified in RFC 2190. This section defines the usage of the RTP fixed header and H.263+ video packet structure.

3.1. RTP Header Usage

Each RTP packet starts with a fixed RTP header. The following fields of the RTP fixed header used for H.263+ video streams are further emphasized here.

Marker bit (M bit): The Marker bit of the RTP header is set to 1 when the current packet carries the end of current frame and is 0 otherwise.

Payload Type (PT): The RTP profile for a particular class of applications will assign a payload type for this encoding, or, if that is not done, a payload type in the dynamic range shall be chosen by the sender.

Timestamp: The RTP Timestamp encodes the sampling instance of the first video frame data contained in the RTP data packet. The RTP timestamp shall be the same on successive packets if a video frame occupies more than one packet. In a multilayer scenario, all pictures corresponding to the same temporal reference should use the same timestamp. If temporal scalability is used (if B-frames are present), the timestamp may not be monotonically increasing in the RTP stream. If B-frames are transmitted on a separate layer and address, they must be synchronized properly with the reference frames. Refer to ITU-T Recommendation H.263 [H263] for information on required transmission order to a decoder. For an H.263+ video stream, the RTP timestamp is based on a 90 kHz clock, the same as that of the RTP payload for H.261 stream [RFC2032]. Since both the H.263+ data and the RTP header contain time information, that timing information must run synchronously. That is, both the RTP timestamp and the temporal reference (TR in the picture header of H.263) should carry the same relative timing information. Any H.263+ picture clock frequency can be expressed as $1800000/(cd*cf)$ source pictures per second, in which cd is an integer from 1 to 127 and cf is either 1000 or 1001. Using the 90 kHz clock of the RTP timestamp, the time

increment between each coded H.263+ picture should therefore be an integer multiple of $(cd*cf)/20$. This will always be an integer for any "reasonable" picture clock frequency (for example, it is 3003 for 30/1.001 Hz NTSC; 3600 for 25 Hz PAL; 3750 for 24 Hz film; and 1500, 1250, or 1200 for the computer display update rates of 60, 72, or 75 Hz, respectively). For RTP packetization of hypothetical H.263+ bitstreams using "unreasonable" custom picture clock frequencies, mathematical rounding could become necessary for generating the RTP timestamps.

3.2. Video Packet Structure

A section of an H.263+ compressed bitstream is carried as a payload within each RTP packet. For each RTP packet, the RTP header is followed by an H.263+ payload header, which is followed by a number of bytes of a standard H.263+ compressed bitstream. The size of the H.263+ payload header is variable, depending on the payload involved, as detailed in the Section 4. The layout of the RTP H.263+ video packet is shown as

```

+-----+
:   RTP Header                               :
+-----+
:   H.263+ Payload Header                    :
+-----+
:   H.263+ Compressed Data Stream           :
+-----+

```

Any H.263+ start codes can be byte aligned by an encoder by using the stuffing mechanisms of H.263+. As specified in H.263+, picture, slice, and EOSBS start codes shall always be byte aligned, and GOB and EOS start codes may be byte aligned. For packetization purposes, GOB start codes should be byte aligned; however, since this is not required in H.263+, there may be some cases where GOB start codes are not aligned, such as when transmitting existing content, or when using H.263 encoders that do not support GOB start code alignment. In this case, Follow-on Packets (see Section 5.2) should be used for packetization.

All H.263+ start codes (Picture, GOB, Slice, EOS, and EOSBS) begin with 16 zero-valued bits. If a start code is byte aligned and it occurs at the beginning of a packet, these two bytes shall be removed from the H.263+ compressed data stream in the packetization process and shall instead be represented by setting a bit (the P bit) in the payload header.

4. Design Considerations

The goals of this payload format are to specify an efficient way of encapsulating an H.263+ standard compliant bitstream and to enhance the resiliency towards packet losses. Due to the large number of different possible coding schemes in H.263+, a copy of the picture header with configuration information is inserted into the payload header when appropriate. The use of that copy of the picture header along with the payload data can allow decoding of a received packet even in cases when another packet containing the original picture header becomes lost.

There are a few assumptions and constraints associated with this H.263+ payload header design. The purpose of this section is to point out various design issues and also to discuss several coding options provided by H.263+ that may impact the performance of network-based H.263+ video.

- o The optional slice structured mode described in Annex K of [H263] enables more flexibility for packetization. Similar to a picture segment that begins with a GOB header, the motion vector predictors in a slice are restricted to reside within its boundaries. However, slices provide much greater freedom in the selection of the size and shape of the area that is represented as a distinct decodable region. In particular, slices can have a size that is dynamically selected to allow the data for each slice to fit into a chosen packet size. Slices can also be chosen to have a rectangular shape, which is conducive for minimizing the impact of errors and packet losses on motion-compensated prediction. For these reasons, the use of the slice structured mode is strongly recommended for any applications used in environments where significant packet loss occurs.
- o In non-rectangular slice structured mode, only complete slices SHOULD be included in a packet. In other words, slices should not be fragmented across packet boundaries. The only reasonable need for a slice to be fragmented across packet boundaries is when the encoder that generated the H.263+ data stream could not be influenced by an awareness of the packetization process (such as when sending H.263+ data through a network other than the one to which the encoder is attached, as in network gateway implementations). Optimally, each packet will contain only one slice.

- o The independent segment decoding (ISD) described in Annex R of [H263] prevents any data dependency across slice or GOB boundaries in the reference picture. It can be utilized to improve resiliency further in high loss conditions.
- o If ISD is used in conjunction with the slice structure, the rectangular slice submode shall be enabled, and the dimensions and quantity of the slices present in a frame shall remain the same between each two intra-coded frames (I-frames), as required in H.263+. The individual ISD segments may also be entirely intra coded from time to time to realize quick error recovery without adding the latency time associated with sending complete INTRA-pictures.
- o When the slice structure is not applied, the insertion of a (preferably byte-aligned) GOB header can be used to provide resync boundaries in the bitstream, as the presence of a GOB header eliminates the dependency of motion vector prediction across GOB boundaries. These resync boundaries provide natural locations for packet payload boundaries.
- o H.263+ allows picture headers to be sent in an abbreviated form in order to prevent repetition of overhead information that does not change from picture to picture. For resiliency, sending a complete picture header for every frame is often advisable. This means (especially in cases with high packet loss probability in which picture header contents are not expected to be highly predictable) that the sender may find it advisable always to set the subfield UFEP in PLUSPTYPE to '001' in the H.263+ video bitstream. (See [H263] for the definition of the UFEP and PLUSPTYPE fields).
- o In a multi-layer scenario, each layer may be transmitted to a different network address. The configuration of each layer, such as the enhancement layer number (ELNUM), reference layer number (RLNUM), and scalability type should be determined at the start of the session and should not change during the course of the session.
- o All start codes can be byte aligned, and picture, slice, and EOSBS start codes are always byte aligned. The boundaries of these syntactical elements provide ideal locations for placing packet boundaries.
- o We assume that a maximum Picture Header size of 504 bits is sufficient. The syntax of H.263+ does not explicitly prohibit larger picture header sizes, but the use of such extremely large picture headers is not expected.

5. H.263+ Payload Header

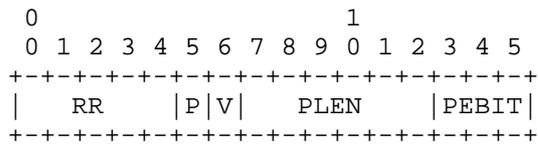
For H.263+ video streams, each RTP packet shall carry only one H.263+ video packet. The H.263+ payload header shall always be present for each H.263+ video packet. The payload header is of variable length. A 16-bit field of the general payload header, defined in 5.1, may be followed by an 8 bit field for Video Redundancy Coding (VRC) information, and/or by a variable-length extra picture header as indicated by PLEN. These optional fields appear in the order given above, when present.

If an extra picture header is included in the payload header, the length of the picture header in number of bytes is specified by PLEN. The minimum length of the payload header is 16 bits, PLEN equal to 0 and no VRC information being present.

The remainder of this section defines the various components of the RTP payload header. Section 6 defines the various packet types that are used to carry different types of H.263+ coded data, and Section 7 summarizes how to distinguish between the various packet types.

5.1. General H.263+ Payload Header

The H.263+ payload header is structured as follows:



RR: 5 bits

Reserved bits. It SHALL be zero and MUST be ignored by receivers.

P: 1 bit

Indicates the picture start or a picture segment (GOB/Slice) start or a video sequence end (EOS or EOSBS). Two bytes of zero bits then have to be prefixed to the payload of such a packet to compose a complete picture/GOB/slice/EOS/EOSBS start code. This bit allows the omission of the two first bytes of the start codes, thus improving the compression ratio.

V: 1 bit

Indicates the presence of an 8-bit field containing information for Video Redundancy Coding (VRC), which follows immediately after the initial 16 bits of the payload header, if present. For syntax and semantics of that 8-bit VRC field, see Section 5.2.

PLEN: 6 bits

Length, in bytes, of the extra picture header. If no extra picture header is attached, PLEN is 0. If PLEN>0, the extra picture header is attached immediately following the rest of the payload header. Note that the length reflects the omission of the first two bytes of the picture start code (PSC). See Section 6.1.

PEBIT: 3 bits

Indicates the number of bits that shall be ignored in the last byte of the picture header. If PLEN is not zero, the ignored bits shall be the least significant bits of the byte. If PLEN is zero, then PEBIT shall also be zero.

5.2. Video Redundancy Coding Header Extension

Video Redundancy Coding (VRC) is an optional mechanism intended to improve error resilience over packet networks. Implementing VRC in H.263+ will require the Reference Picture Selection option described in Annex N of [H263]. By having multiple "threads" of independently inter-frame predicted pictures, damage to an individual frame will cause distortions only within its own thread, leaving the other threads unaffected. From time to time, all threads converge to a so-called sync frame (an INTRA picture or a non-INTRA picture that is redundantly represented within multiple threads); from this sync frame, the independent threads are started again. For more information on codec support for VRC, see [Vredun].

P-picture temporal scalability is another use of the reference picture selection mode and can be considered a special case of VRC in which only one copy of each sync frame may be sent. It offers a thread-based method of temporal scalability without the increased delay caused by the use of B pictures. In this use, sync frames sent in the first thread of pictures are also used for the prediction of a second thread of pictures that fall temporally between the sync frames to increase the resulting frame rate. In this use, the pictures in the second thread can be discarded in order to obtain a reduction of bit rate or decoding complexity without harming the ability to decode later pictures. A third or more threads, can also be added, but each thread is predicted only from the sync frames

(which are sent at least in thread 0) or from frames within the same thread.

While a VRC data stream is (like all H.263+ data) totally self-contained, it may be useful for the transport hierarchy implementation to have knowledge about the current damage status of each thread. On the Internet, this status can easily be determined by observing the marker bit, the sequence number of the RTP header, the thread-id, and a circling "packet per thread" number. The latter two numbers are coded in the VRC header extension.

The format of the VRC header extension is as follows:

```

  0 1 2 3 4 5 6 7
  +---+---+---+---+
  | TID | Trun |S|
  +---+---+---+---+

```

TID: 3 bits

Thread ID. Up to 7 threads are allowed. Each frame of H.263+ VRC data will use as reference information only sync frames or frames within the same thread. By convention, thread 0 is expected to be the "canonical" thread, which is the thread from which the sync frame should ideally be used. In the case of corruption or loss of the thread 0 representation, a representation of the sync frame with a higher thread number can be used by the decoder. Lower thread numbers are expected to contain representations of the sync frames equal to or better than higher thread numbers in the absence of data corruption or loss. See [Vredun] for a detailed discussion of VRC.

Trun: 4 bits

Monotonically increasing (modulo 16) 4-bit number counting the packet number within each thread.

S: 1 bit

A bit that indicates that the packet content is for a sync frame. An encoder using VRC may send several representations of the same "sync" picture, in order to ensure that, regardless of which thread of pictures is corrupted by errors or packet losses, the reception of at least one representation of a particular picture is ensured (within at least one thread). The sync picture can then be used for the prediction of any thread. If packet losses have not occurred, then the sync frame contents of thread 0 can be used, and those of other threads can be discarded (and similarly for other threads). Thread 0 is considered the "canonical" thread, the use of which is preferable

to all others. The contents of packets having lower thread numbers shall be considered as having a higher processing and delivery priority than those with higher thread numbers. Thus, packets having lower thread numbers for a given sync frame shall be delivered first to the decoder under loss-free and low-time-jitter conditions, which will result in the discarding of the sync contents of the higher-numbered threads as specified in Annex N of [H263].

6. Packetization Schemes

6.1. Picture Segment Packets and Sequence Ending Packets (P=1)

A picture segment packet is defined as a packet that starts at the location of a Picture, GOB, or slice start code in the H.263+ data stream. This corresponds to the definition of the start of a video picture segment as defined in H.263+. For such packets, P=1 always.

An extra picture header can sometimes be attached in the payload header of such packets. Whenever an extra picture header is attached as signified by PLEN>0, only the last six bits of its picture start code, '100000', are included in the payload header. A complete H.263+ picture header with byte-aligned picture start code can be conveniently assembled on the receiving end by prepending the sixteen leading '0' bits.

When PLEN>0, the end bit position corresponding to the last byte of the picture header data is indicated by PEBIT. The actual bitstream data shall begin on an 8-bit byte boundary following the payload header.

A sequence ending packet is defined as a packet that starts at the location of an EOS or EOSBS code in the H.263+ data stream. This delineates the end of a sequence of H.263+ video data (more H.263+ video data may still follow later, however, as specified in ITU-T Recommendation H.263). For such packets, P=1 and PLEN=0 always.

The optional header extension for VRC may or may not be present as indicated by the V bit flag.

6.1.1. Packets that begin with a Picture Start Code

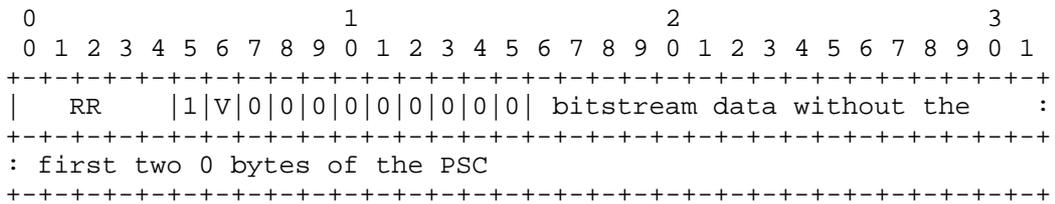
Any packet that contains the whole or the start of a coded picture shall start at the location of the picture start code (PSC) and should normally be encapsulated with no extra copy of the picture header. In other words, normally PLEN=0 in such a case. However, if the coded picture contains an incomplete picture header (UFEP = "000"), then a representation of the complete (UFEP = "001") picture header may be attached during packetization in order to provide

greater error resilience. Thus, for packets that start at the location of a picture start code, PLEN shall be zero unless both of the following conditions apply:

- 1) The picture header in the H.263+ bitstream payload is incomplete (PLUSPTYPE present and UFEP="000").
- 2) The additional picture header that is attached is not incomplete (UFEP="001").

A packet that begins at the location of a Picture, GOB, slice, EOS, or EOSBS start code shall omit the first two (all zero) bytes from the H.263+ bitstream and signify their presence by setting P=1 in the payload header.

Here is an example of encapsulating the first packet in a frame (without an attached redundant complete picture header):



6.1.2. Packets that begin with GBSC or SSC

For a packet that begins at the location of a GOB or slice start code (GBSC), PLEN may be zero or nonzero, depending on whether a redundant picture header is attached to the packet. In environments with very low packet loss rates, or when picture header contents are very seldom likely to change (except as can be detected from the GOB Frame ID (GFID) syntax of H.263+), a redundant copy of the picture header is not required. However, in less ideal circumstances a redundant picture header should be attached for enhanced error resilience, and its presence is indicated by PLEN>0.

Assuming a PLEN of 9 and P=1, below is an example of a packet that begins with a byte-aligned GBSC or a Slice Start Code (SSC):

```

0          1          2          3
0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-----+-----+-----+-----+-----+-----+-----+-----+
|  RR      |1|V|0 0 1 0 0 1|PEBIT|1 0 0 0 0 0| picture header  :
+-----+-----+-----+-----+-----+-----+-----+-----+
: starting with TR, PTYPE ...                                     |
+-----+-----+-----+-----+-----+-----+-----+-----+
| ...                                                         | bitstream  :
+-----+-----+-----+-----+-----+-----+-----+-----+
: data starting with GBSC/SSC without its first two 0 bytes
+-----+-----+-----+-----+-----+-----+-----+-----+
    
```

Notice that only the last six bits of the picture start code, '100000', are included in the payload header. A complete H.263+ picture header with byte aligned picture start code can be conveniently assembled, if needed, on the receiving end by prepending the sixteen leading '0' bits.

6.1.1.3. Packets that begin with an EOS or EOSBS Code

For a packet that begins with an EOS or EOSBS code, PLEN shall be zero, and no Picture, GOB, or Slice start codes shall be included within the same packet. As with other packets beginning with start codes, the two all-zero bytes that begin the EOS or EOSBS code at the beginning of the packet shall be omitted, and their presence shall be indicated by setting the P bit to 1 in the payload header.

System designers should be aware that some decoders may interpret the loss of a packet containing only EOS or EOSBS information as the loss of essential video data and may thus respond by not displaying some subsequent video information. Since EOS and EOSBS codes do not actually affect the decoding of video pictures, they are somewhat unnecessary to send at all. Because of the danger of misinterpretation of the loss of such a packet (which can be detected by the sequence number), encoders are generally to be discouraged from sending EOS and EOSBS.

Below is an example of a packet containing an EOS code:

```

0          1          2
0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3
+-----+-----+-----+-----+-----+-----+-----+-----+
|  RR      |1|V|0|0|0|0|0|0|0|0|0|0|0|0|1|1|1|1|1|1|0|0|
+-----+-----+-----+-----+-----+-----+-----+-----+
    
```

6.2. Encapsulating Follow-on Packet (P=0)

A Follow-on Packet contains a number of bytes of coded H.263+ data that do not start at a synchronization point. That is, a Follow-on Packet does not start with a Picture, GOB, Slice, EOS, or EOSBS header, and it may or may not start at a macroblock boundary. Since Follow-on Packets do not start at synchronization points, the data at the beginning of a Follow-on Packet is not independently decodable. For such packets, P=0 always. If the preceding packet of a Follow-on Packet got lost, the receiver may discard that Follow-on Packet, as well as all other following Follow-on Packets. Better behavior, of course, would be for the receiver to scan the interior of the packet payload content to determine whether any start codes are found in the interior of the packet that can be used as resync points. The use of an attached copy of a picture header for a Follow-on Packet is useful only if the interior of the packet or some subsequent Follow-on Packet contains a resync code, such as a GOB or slice start code. PLEN>0 is allowed, since it may allow resync in the interior of the packet. The decoder may also be resynchronized at the next segment or picture packet.

Here is an example of a Follow-on Packet (with PLEN=0):

```

      0                1                2                3
      0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-----+-----+-----+-----+-----+-----+-----+-----+
|  RR      |0|V|0|0|0|0|0|0|0|0|0|0|0|0|0|0| bitstream data
+-----+-----+-----+-----+-----+-----+-----+-----+

```

7. Use of this Payload Specification

There is no syntactical difference between a picture segment packet and a Follow-on Packet, other than the indication P=1 for picture segment or sequence ending packets and P=0 for Follow-on Packets. See the following for a summary of the entire packet types and ways to distinguish between them.

It is possible to distinguish between the different packet types by checking the P bit and the first 6 bits of the payload along with the header information. The following table shows the packet type for permutations of this information (see also the picture/GOB/Slice header descriptions in H.263+ for details):

First 6 bits of Payload	P-Bit (payload hdr.)	PLEN	Packet	Remarks
100000	1	0	Picture	Typical Picture
100000	1	> 0	Picture	Note UFEP
1xxxxx	1	0	GOB/Slice/EOS/EOSBS	See possible GNs
1xxxxx	1	> 0	GOB/Slice	See possible GNs
Xxxxxx	0	0	Follow-on	
Xxxxxx	0	> 0	Follow-on	Interior Resync

The details regarding the possible values of the five bit Group Number (GN) field that follows the initial "1" bit when the P-bit is "1" for a GOB, Slice, EOS, or EOSBS packet are found in Section 5.2.3 of H.263 [H263].

As defined in this specification, every start of a coded frame (as indicated by the presence of a PSC) has to be encapsulated as a picture segment packet. If the whole coded picture fits into one packet of reasonable size (which is dependent on the connection characteristics), this is the only type of packet that may need to be used. Due to the high compression ratio achieved by H.263+, it is often possible to use this mechanism, especially for small spatial picture formats such as Quarter Common Intermediate Format (QCIF) and typical Internet packet sizes around 1500 bytes.

If the complete coded frame does not fit into a single packet, two different ways for the packetization may be chosen. In case of very low or zero packet loss probability, one or more Follow-on Packets may be used for coding the rest of the picture. Doing so leads to minimal coding and packetization overhead, as well as to an optimal use of the maximal packet size, but does not provide any added error resilience.

The alternative is to break the picture into reasonably small partitions, called Segments (by using the Slice or GOB mechanism), that do offer synchronization points. By doing so and using the Picture Segment payload with PLEN>0, decoding of the transmitted packets is possible even in cases in which the Picture packet containing the picture header was lost (provided any necessary reference picture is available). Picture Segment packets can also be used in conjunction with Follow-on Packets for large segment sizes.

8. Media Type Definition

This section specifies optional parameters that MAY be used to select optional features of the H.263 codec. The parameters are specified here as part of the Media Type registration for the ITU-T H.263 codec. A mapping of the parameters into the Session Description Protocol (SDP) [RFC4566] is also provided for applications that use SDP. Multiple parameters SHOULD be expressed as a media type string, in the form of a semicolon-separated list of parameter=value pairs.

8.1. Media Type Registrations

This section describes the media types and names associated with this payload format. The section updates the previous registered version in RFC 3555 [RFC3555].

8.1.1. Registration of Media Type video/H263-1998

Type name: video

Subtype name: H263-1998

Required parameters: None

Optional parameters:

SQCIF: Specifies the MPI (Minimum Picture Interval) for SQCIF resolution. Permissible values are integer values from 1 to 32, which correspond to a maximum frame rate of $30/(1.001 * \text{the specified value})$ frames per second.

QCIF: Specifies the MPI (Minimum Picture Interval) for QCIF resolution. Permissible values are integer values from 1 to 32, which correspond to a maximum frame rate of $30/(1.001 * \text{the specified value})$ frames per second.

CIF: Specifies the MPI (Minimum Picture Interval) for CIF resolution. Permissible values are integer values from 1 to 32, which correspond to a maximum frame rate of $30/(1.001 * \text{the specified value})$ frames per second.

CIF4: Specifies the MPI (Minimum Picture Interval) for 4CIF resolution. Permissible values are integer values from 1 to 32, which correspond to a maximum frame rate of $30/(1.001 * \text{the specified value})$ frames per second.

CIF16: Specifies the MPI (Minimum Picture Interval) for 16CIF resolution. Permissible values are integer values from 1 to 32, which correspond to a maximum frame rate of $30/(1.001 * \text{the specified value})$ frames per second.

CUSTOM: Specifies the MPI (Minimum Picture Interval) for a custom-defined resolution. The custom parameter receives three comma-separated values, Xmax, Ymax, and MPI. The Xmax and Ymax parameters describe the number of pixels in the X and Y axis and must be evenly divisible by 4. The permissible values for MPI are integer values from 1 to 32, which correspond to a maximum frame rate of $30/(1.001 * \text{the specified value})$.

A system that declares support of a specific MPI for one of the resolutions SHALL also implicitly support a lower resolution with the same MPI.

A list of optional annexes specifies which annexes of H.263 are supported. The optional annexes are defined as part of H263-1998, H263-2000. H.263 annex X [H263] defines profiles that group annexes for specific applications. A system that supports a specific annex SHALL specify its support using the optional parameters. If no annex is specified, then the stream is Baseline H.263.

The allowed optional parameters for the annexes are "F", "I", "J", "T", "K", "N", and "P".

"F", "I", "J", and "T" if supported, SHALL have the value "1". If not supported, they should not be listed or SHALL have the value "0".

"K" can receive one of four values 1 - 4:

1: Slices In Order, Non-Rectangular

2: Slices In Order, Rectangular

3: Slices Not Ordered, Non-Rectangular

4: Slices Not Ordered, Rectangular

"N": Reference Picture Selection mode - Four numeric choices (1 - 4) are available, representing the following modes:

1: NEITHER: No back-channel data is returned from the decoder to the encoder.

- 2: ACK: The decoder returns only acknowledgment messages.
- 3: NACK: The decoder returns only non-acknowledgment messages.
- 4: ACK+NACK: The decoder returns both acknowledgment and non-acknowledgment messages.

No special provision is made herein for carrying back channel information. The Extended RTP Profile for RTCP-based Feedback [RFC4585] MAY be used as a back channel mechanism.

"P": Reference Picture Resampling, in which the following submodes are represented as a number from 1 to 4:

- 1: dynamicPictureResizingByFour
- 2: dynamicPictureResizingBySixteenthPel
- 3: dynamicWarpingHalfPel
- 4: dynamicWarpingSixteenthPel

Example: P=1,3

PAR: Arbitrary Pixel Aspect Ratio. Defines the width:height ratio by two colon-separated integers between 0 and 255. Default ratio is 12:11, if not otherwise specified.

CPCF: Arbitrary (Custom) Picture Clock Frequency: CPCF is a comma-separated list of eight parameters specifying a custom picture clock frequency and the MPI (minimum picture interval) for the supported picture sizes when using that picture clock frequency. The first two parameters are *cd*, which is an integer from 1 to 127, and *cf*, which is either 1000 or 1001. The custom picture clock frequency is given by the formula $1800000/(cd*cf)$ provided in the RTP Timestamp semantics in Section 3.1 above (as specified in H.263 section 5.1.7). Following the values of *cd* and *cf*, the remaining six parameters are SQCIFMPI, QCIFMPI, CIFMPI, CIF4MPI, CIF16MPI, and CUSTOMMPI, which each specify an integer MPI (minimum picture interval) for the standard picture sizes SQCIF, QCIF, CIF, 4CIF, 16CIF, and CUSTOM, respectively, as described above. The MPI value indicates a maximum frame rate of $1800000/(cd*cf*MPI)$ frames per second for MPI parameters having a value in the range from 1 to 2048, inclusive. An MPI value of 0 specifies that the associated picture size is not supported for the custom picture clock frequency. If the CUSTOMMPI parameter is not equal to 0, the CUSTOM parameter SHALL also be present (so

that the Xmax and Ymax dimensions of the custom picture size are defined).

BPP: BitsPerPictureMaxKb. Maximum number of bits in units of 1024 bits allowed to represent a single picture. If this parameter is not present, then the default value, based on the maximum supported resolution, is used. BPP is integer value between 0 and 65536.

HRD: Hypothetical Reference Decoder. See annex B of H.263 specification [H263]. This parameter, if supported, SHALL have the value "1". If not supported, it should not be listed or SHALL have the value "0".

Encoding considerations:

This media type is framed and binary; see Section 4.8 in [RFC4288]

Security considerations: See Section 11 of RFC 4629

Interoperability considerations:

These are receiver options; current implementations will not send any optional parameters in their SDP. They will ignore the optional parameters and will encode the H.263 stream without any of the annexes. Most decoders support at least QCIF and CIF fixed resolutions, and they are expected to be available almost in every H.263-based video application.

Published specification: RFC 4629

Applications that use this media type:

Audio and video streaming and conferencing tools.

Additional information: None

Person and email address to contact for further information:

Roni Even: roni.even@polycom.co.il

Intended usage: COMMON

Restrictions on usage:

This media type depends on RTP framing and thus is only defined for transfer via RTP [RFC3550]. Transport within other framing protocols is not defined at this time.

Author: Roni Even

Change controller:

IETF Audio/Video Transport working group, delegated from the IESG.

8.1.1.2. Registration of Media Type video/H263-2000

Type name: video

Subtype name: H263-2000

Required parameters: None

Optional parameters:

The optional parameters of the H263-1998 type MAY be used with this media subtype. Specific optional parameters that may be used with the H263-2000 type are as follows:

PROFILE: H.263 profile number, in the range 0 through 10, specifying the supported H.263 annexes/subparts based on H.263 annex X [H263]. The annexes supported in each profile are listed in table X.1 of H.263 annex X. If no profile or H.263 annex is specified, then the stream is Baseline H.263 (profile 0 of H.263 annex X).

LEVEL: Level of bitstream operation, in the range 0 through 100, specifying the level of computational complexity of the decoding process. The level are described in table X.2 of H.263 annex X.

According to H.263 annex X, support of any level other than level 45 implies support of all lower levels. Support of level 45 implies support of level 10.

A system that specifies support of a PROFILE MUST specify the supported LEVEL.

INTERLACE: Interlaced or 60 fields indicates the support for interlace display mode, as specified in H.263 annex W.6.3.11. This parameter, if supported SHALL have the value "1". If not supported, it should not be listed or SHALL have the value "0".

Encoding considerations:

This media type is framed and binary; see Section 4.8 in [RFC4288]

Security considerations: See Section 11 of RFC 4629

Interoperability considerations:

The optional parameters PROFILE and LEVEL SHALL NOT be used with any of the other optional parameters.

Published specification: RFC 4629

Applications that use this media type:

Audio and video streaming and conferencing tools.

Additional information: None

Person and email address to contact for further information :

Roni Even: roni.even@polycom.co.il

Intended usage: COMMON

Restrictions on usage:

This media type depends on RTP framing and thus is only defined for transfer via RTP [RFC3550]. Transport within other framing protocols is not defined at this time.

Author: Roni Even

Change controller:

IETF Audio/Video Transport working group delegated from the IESG.

8.2. SDP Usage

The media types video/H263-1998 and video/H263-2000 are mapped to fields in the Session Description Protocol (SDP) as follows:

- o The media name in the "m=" line of SDP MUST be video.
- o The encoding name in the "a=rtpmap" line of SDP MUST be H263-1998 or H263-2000 (the media subtype).
- o The clock rate in the "a=rtpmap" line MUST be 90000.
- o The optional parameters, if any, MUST be included in the "a=fmtp" line of SDP. These parameters are expressed as a media type string, in the form of a semicolon-separated list of parameter=value pairs. The optional parameters PROFILE and LEVEL SHALL NOT be used with any of the other optional parameters.

8.2.1. Usage with the SDP Offer Answer Model

For offering H.263 over RTP using SDP in an Offer/Answer model [RFC3264], the following considerations are necessary.

Codec options (F,I,J,K,N,P,T): These options MUST NOT appear unless the sender of these SDP parameters is able to decode those options. These options designate receiver capabilities even when sent in a "sendonly" offer.

Profile: The offer of a SDP profile parameter signals that the offerer can decode a stream that uses the specified profile. Each profile uses different H.263 annexes, so there is no implied relationship between them. An answerer SHALL NOT change the profile parameter and MUST reject the payload type containing an unsupported profile. A decoder that supports a profile SHALL also support H.263 baseline profile (profile 0). An offerer is RECOMMENDED to offer all the different profiles it is interested to use as individual payload types. In addition an offerer, sending an offer using the PROFILE optional parameter, is RECOMMENDED to offer profile 0, as this will enable communication, and in addition allows an answerer to add those profiles it does support in an answer.

LEVEL: The LEVEL parameter in an offer indicates the maximum computational complexity supported by the offerer in performing decoding for the given PROFILE. An answerer MAY change the value (both up and down) of the LEVEL parameter in its answer to indicate the highest value it supports.

INTERLACE: The parameter MAY be included in either offer or answer to indicate that the offerer or answerer respectively supports reception of interlaced content. The inclusion in either offer or answer is independent of each other.

Picture sizes and MPI: Supported picture sizes and their corresponding minimum picture interval (MPI) information for H.263 can be combined. All picture sizes can be advertised to the other party, or only a subset. The terminal announces only those picture sizes (with their MPIs) which it is willing to receive. For example, MPI=2 means that the maximum (decodable) picture rate per second is 15/1.001 (approximately 14.985).

If the receiver does not specify the picture size/MPI optional parameter, then it SHOULD be ready to receive QCIF resolution with MPI=1.

Parameters offered first are the most preferred picture mode to be received.

Here is an example of the usage of these parameters:

```
CIF=4;QCIF=3;SQCIF=2;CUSTOM=360,240,2
```

This means that the encoder SHOULD send CIF picture size, which it can decode at MPI=4. If that is not possible, then QCIF with MPI value 3 should be sent; if neither are possible, then SQCIF with MPI value=2. The receiver is capable of (but least preferred) decoding custom picture sizes (max 360x240) with MPI=2. Note that most decoders support at least QCIF and CIF fixed resolutions, and that they are expected to be available almost in every H.263-based video application.

Below is an example of H.263 SDP in an offer:

```
a=fmtp:xx CIF=4;QCIF=2;F=1;K=1
```

This means that the sender of this message can decode an H.263 bit stream with the following options and parameters: preferred resolution is CIF (at up to 30/4.004 frames per second), but if that is not possible then QCIF size is also supported (at up to 30/2.002 frames per second). Advanced Prediction mode (AP) and slicesInOrder-NonRect options MAY be used.

Below is an example of H.263 SDP in an offer that includes the CPCF parameter.

```
a=fmtp:xx CPCF=36,1000,0,1,1,0,0,2;CUSTOM=640,480,2;CIF=1;QCIF=1
```

This means that the sender of this message can decode an H.263 bit stream with a preferred custom picture size of 640x480 at a maximum frame rate of 25 frames per second using a custom picture clock frequency of 50 Hz. If that is not possible, then the 640x480 picture size is also supported at up to 30/2.002 frames per second using the ordinary picture clock frequency of 30/1.001 Hz. If neither of those is possible, then the CIF and QCIF picture sizes are also supported at up to 50 frames per second using the custom picture clock frequency of 50 Hz or up to 30/1.001 frames per second using the ordinary picture clock frequency of 30/1.001 Hz, and CIF is preferred over QCIF.

The following limitation applies for usage of these media types when performing offer/answer for sessions using multicast transport. An answerer SHALL NOT change any of the parameters in an answer, instead if the indicated values are not supported the payload type MUST be rejected.

9. Backward Compatibility to RFC 2429

The current document is a revision of RFC 2429 and obsoletes it. This section will address the backward compatibility issues.

9.1. New Optional Parameters for SDP

The document adds new optional parameters to the H263-1998 and H263-2000 payload type, defined in RFC 3555 [RFC3555]. Since these are optional parameters we expect that old implementations will ignore these parameters, and that new implementations that will receive the H263-1998 and H263-2000 payload types with no parameters will behave as if the other side can accept H.263 at QCIF resolution at a frame rate not exceeding 15/1.001 (approximately 14.985) frames per second.

10. IANA Considerations

This document updates the H.263 (1998) and H.263 (2000) media types, described in RFC 3555 [RFC3555]. The updated media type registrations are in Section 8.1.

11. Security Considerations

RTP packets using the payload format defined in this specification are subject to the security considerations discussed in the RTP specification [RFC3550] and any appropriate RTP profile (for example, [RFC3551]). This implies that confidentiality of the media streams is achieved by encryption. Because the data compression used with this payload format is applied end-to-end, encryption may be performed after compression, so there is no conflict between the two operations.

A potential denial-of-service threat exists for data encoding using compression techniques that have non-uniform receiver-end computational load. The attacker can inject pathological datagrams into the stream that are complex to decode and cause the receiver to be overloaded. The usage of authentication of at least the RTP packet is RECOMMENDED.

As with any IP-based protocol, in some circumstances a receiver may be overloaded simply by the receipt of too many packets, either desired or undesired. Network-layer authentication may be used to discard packets from undesired sources, but the processing cost of the authentication itself may be too high. In a multicast environment, pruning of specific sources may be implemented in future versions of IGMP [RFC2032] and in multicast routing protocols to allow a receiver to select which sources are allowed to reach it.

A security review of this payload format found no additional considerations beyond those in the RTP specification.

12. Acknowledgements

This is to acknowledge the work done by Chad Zhu, Linda Cline, Gim Deisher, Tom Gardos, Christian Maciocco, and Donald Newell from Intel Corp., who co-authored RFC 2429.

We would also like to acknowledge the work of Petri Koskelainen from Nokia and Nermeen Ismail from Cisco, who helped with composing the text for the new media types.

13. Changes from Previous Versions of the Documents

13.1. Changes from RFC 2429

The changes from the RFC 2429 are:

1. The H.263 1998 and 2000 media type are now in the payload specification.
2. Added optional parameters to the H.263 1998 and 2000 media types.
3. Mandate the usage of RFC 2429 for all H.263. RFC 2190 payload format should be used only to interact with legacy systems.

13.2. Changes from RFC 3555

This document adds new optional parameters to the H263-1998 and H263-2000 payload types.

14. References

14.1. Normative References

- [H263] International Telecommunications Union - Telecommunication Standardization Sector, "Video coding for low bit rate communication", ITU-T Recommendation H.263, January 2005.
- [RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, March 1997.
- [RFC3550] Schulzrinne, H., Casner, S., Frederick, R., and V. Jacobson, "RTP: A Transport Protocol for Real-Time Applications", STD 64, RFC 3550, July 2003.

- [RFC3551] Schulzrinne, H. and S. Casner, "RTP Profile for Audio and Video Conferences with Minimal Control", STD 65, RFC 3551, July 2003.
- [RFC3555] Casner, S. and P. Hoschka, "MIME Type Registration of RTP Payload Formats", RFC 3555, July 2003.
- [RFC4566] Handley, M., Jacobson, V., and C. Perkins, "SDP: Session Description Protocol", RFC 4566, July 2006.

14.2. Informative References

- [RFC2032] Turletti, T., "RTP Payload Format for H.261 Video Streams", RFC 2032, October 1996.
- [RFC2190] Zhu, C., "RTP Payload Format for H.263 Video Streams", RFC 2190, September 1997.
- [RFC2429] Bormann, C., Cline, L., Deisher, G., Gardos, T., Maciocco, C., Newell, D., Ott, J., Sullivan, G., Wenger, S., and C. Zhu, "RTP Payload Format for the 1998 Version of ITU-T Rec. H.263 Video (H.263+)", RFC 2429, October 1998.
- [RFC3264] Rosenberg, J. and H. Schulzrinne, "An Offer/Answer Model with Session Description Protocol (SDP)", RFC 3264, June 2002.
- [RFC4288] Freed, N. and J. Klensin, "Media Type Specifications and Registration Procedures", BCP 13, RFC 4288, December 2005.
- [RFC4585] Ott, J., Wenger, S., Sato, N., Burmeister, C., and J. Rey, "Extended RTP Profile for Real-time Transport Control Protocol (RTCP)-Based Feedback (RTP/AVPF)", RFC 4585, July 2006.
- [RFC4628] Even, R., "RTP Payload Format for H.263 Moving RFC 2190 to Historic Status", RFC 4628, January 2007.
- [Vredun] Wenger, S., "Video Redundancy Coding in H.263+", Proc. Audio-Visual Services over Packet Networks, Aberdeen, U.K. 9/1997, September 1997.

Authors' Addresses

Joerg Ott
Helsinki University of Technology
Networking Laboratory
PO Box 3000
02015 TKK, Finland

EMail: jo@netlab.tkk.fi

Carsten Bormann
Universitaet Bremen TZI
Postfach 330440
D-28334 Bremen, GERMANY

Phone: +49.421.218-7024
Fax: +49.421.218-7000
EMail: cabo@tzi.org

Gary Sullivan
Microsoft Corp.
One Microsoft Way
Redmond, WA 98052
USA

EMail: garysull@microsoft.com

Stephan Wenger
Nokia Research Center
P.O. Box 100
33721 Tampere
Finland

EMail: stewe@stewe.org

Roni Even (editor)
Polycom
94 Derech Em Hamoshavot
Petach Tikva 49130
Israel

EMail: roni.even@polycom.co.il

Full Copyright Statement

Copyright (C) The IETF Trust (2007).

This document is subject to the rights, licenses and restrictions contained in BCP 78, and except as set forth therein, the authors retain all their rights.

This document and the information contained herein are provided on an "AS IS" basis and THE CONTRIBUTOR, THE ORGANIZATION HE/SHE REPRESENTS OR IS SPONSORED BY (IF ANY), THE INTERNET SOCIETY, THE IETF TRUST AND THE INTERNET ENGINEERING TASK FORCE DISCLAIM ALL WARRANTIES, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO ANY WARRANTY THAT THE USE OF THE INFORMATION HEREIN WILL NOT INFRINGE ANY RIGHTS OR ANY IMPLIED WARRANTIES OF MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE.

Intellectual Property

The IETF takes no position regarding the validity or scope of any Intellectual Property Rights or other rights that might be claimed to pertain to the implementation or use of the technology described in this document or the extent to which any license under such rights might or might not be available; nor does it represent that it has made any independent effort to identify any such rights. Information on the procedures with respect to rights in RFC documents can be found in BCP 78 and BCP 79.

Copies of IPR disclosures made to the IETF Secretariat and any assurances of licenses to be made available, or the result of an attempt made to obtain a general license or permission for the use of such proprietary rights by implementers or users of this specification can be obtained from the IETF on-line IPR repository at <http://www.ietf.org/ipr>.

The IETF invites any interested party to bring to its attention any copyrights, patents or patent applications, or other proprietary rights that may cover technology that may be required to implement this standard. Please address the information to the IETF at ietf-ipr@ietf.org.

Acknowledgement

Funding for the RFC Editor function is currently provided by the Internet Society.

