Network Working Group
Request for Comments: 2429
Category: Standards Track

C. Bormann
Univ. Bremen
L. Cline
G. Deisher
T. Gardos
C. Maciocco
D. Newell
Intel
J. Ott
Univ. Bremen
G. Sullivan
PictureTel
S. Wenger
TU Berlin
C. Zhu
Intel

October 1998

# RTP Payload Format for the 1998 Version of ITU-T Rec. H.263 Video (H.263+)

## Status of this Memo

## Copyright Notice

## 1. Introduction

This document specifies an RTP payload header format applicable to the transmission of video streams generated based on the 1998 version of ITU-T Recommendation H.263 [4]. Because the 1998 version of H.263 is a superset of the 1996 syntax, this format can also be used with the 1996 version of H.263 [3], and is recommended for this use by new implementations. This format does not replace RFC 2190, which continues to be used by existing implementations, and may be required for backward compatibility in new implementations. Implementations using the new features of the 1998 version of H.263 shall use the format described in this document.

The 1998 version of ITU-T Recommendation H.263 added numerous coding options to improve codec performance over the 1996 version. The 1998 version is referred to as H.263+ in this document. 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 [4] 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 to 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 can 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.

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 the refinement layer is twice the size of the base layer in the horizontal dimension, vertical dimension, or both.

## 2. 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, with the only exception being that when the payload of a packet begins with a Picture, GOB, Slice, EOS, or EOSBS start code, the first two (all-zero) bytes of the start code are 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 4. This section defines the usage of the RTP fixed header and H.263+ video packet structure.

### 2.1 RTP Header Usage

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

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 Payload Type shall specify the H.263+ video payload format.

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 the 1998 ITU-T Recommendation H.263 [4] 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 [5]. Since both the H.263+ data and the RTP header contain time information, it is required that those timing information 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 a integer multiple of (cd*cf)/20. This will always be an integer for any "reasonable" picture clock frequency (for example, it is 3003 for 29.97 Hz NTSC, 3600 for 25 Hz PAL, 3750 for 24 Hz film, and 1500, 1250 and 1200 for the computer display update rates of 60, 72 and 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.

## 2.2 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:

```
  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
 +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
 |    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 starts 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.

## 3. 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 such cases in which 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.

- The optional slice structured mode described in Annex K of H.263+ [4] 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 which is represented as a distinct decodable region. In particular, slices can have a size which 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.

- 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 which 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.

- The independent segment decoding (ISD) described in Annex R of [4] prevents any data dependency across slice or GOB boundaries in the reference picture. It can be utilized to further improve resiliency in high loss conditions.

- 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.

- 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.

- 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 that (especially in cases with high packet loss probability in which picture header contents are not expected to be highly predictable), the sender may find it advisable to always set the subfield UFEP in PLUSPTYPE to ’001’ in the H.263+ video bitstream. (See [4] for the definition of the UFEP and PLUSPTYPE fields).

- 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.

- 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.

- 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.

## 4. H.263+ Payload Header

For H.263+ video streams, each RTP packet carries only one H.263+ video packet. The H.263+ payload header is always present for each H.263+ video packet. The payload header is of variable length. A 16 bit field of the basic payload header 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, corresponding to PLEN equal to 0 and no VRC information present.

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

### 4.1 General H.263+ payload header

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

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

RR: 5 bits
Reserved bits. Shall be zero.

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 4.2.

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 the length reflects the omission of the first two bytes of the picture start code (PSC). See section 5.1.

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.

## 4.2 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 [4]. By having multiple "threads" of independently inter-frame predicted pictures, damage of individual frame will cause distortions only within its own thread but leave 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 which 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 [7].

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 which 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 as well, 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, and 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 equal or better representations of the sync frames than higher thread numbers in the absence of data corruption or loss. See [7] 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 [4].

## 5. Packetization schemes

### 5.1 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.

### 5.1.1 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"), and

2) The additional picture header which is attached is not incomplete (UFEP="001").

A packet which 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):

```
 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|0|0|0|0|0|0|0| bitstream data without the    |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
| first two 0 bytes of the PSC                                ...
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

### 5.1.2 Packets that begin with GBSC or SSC

For a packet that begins at the location of a GOB or slice start code, PLEN may be zero or may be 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 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 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.

### 5.1.3 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|1|1|1|1|1|1|0|0|
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

### 5.2 Encapsulating Follow-On Packet (P=0)

A Follow-on packet contains a number of bytes of coded H.263+ data which does 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 which 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| bitstream data              ...
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

## 6. 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 | PLEN hdr.) | 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 which 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 [4].

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 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 such 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.

## 7. Security Considerations

RTP packets using the payload format defined in this specification are subject to the security considerations discussed in the RTP specification [1], and any appropriate RTP profile (for example [2]). 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 encodings using compression techniques that have non-uniform receiver-end computational load. The attacker can inject pathological datagrams into the stream which are complex to decode and cause the receiver to be overloaded. However, this encoding does not exhibit any significant non-uniformity.

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 [5] 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.

## 8. Addresses of Authors

Carsten Bormann
Universitaet Bremen FB3 TZI      EMail: cabo@tzi.org
Postfach 330440                  Phone: +49.421.218-7024
D-28334 Bremen, GERMANY          Fax:   +49.421.218-7000

Linda Cline
Intel Corp. M/S JF3-206          EMail: lscline@jf.intel.com
2111 NE 25th Avenue              Phone: +1 503 264 3501
Hillsboro, OR 97124, USA         Fax:   +1 503 264 3483

Gim Deisher
Intel Corp. M/S JF2-78           EMail: gim.l.deisher@intel.com
2111 NE 25th Avenue              Phone: +1 503 264 3758
Hillsboro, OR 97124, USA         Fax:   +1 503 264 9372

Tom Gardos
Intel Corp. M/S JF2-78           EMail: thomas.r.gardos@intel.com
2111 NE 25th Avenue              Phone: +1 503 264 6459
Hillsboro, OR 97124, USA         Fax:   +1 503 264 9372

Christian Maciocco
Intel Corp. M/S JF3-206          EMail: christian.maciocco@intel.com
2111 NE 25th Avenue              Phone: +1 503 264 1770
Hillsboro, OR 97124, USA         Fax:   +1 503 264 9428

Donald Newell
Intel Corp. M/S JF3-206          EMail: donald.newell@intel.com
2111 NE 25th Avenue              Phone: +1 503 264 9234
Hillsboro, OR 97124, USA         Fax:   +1 503 264 9428

Joerg Ott
Universitaet Bremen FB3 TZI
Postfach 330440
D-28334 Bremen, GERMANY
EMail: jo@tzi.org
Phone: +49.421.218-7024
Fax: +49.421.218-7000

Gary Sullivan
PictureTel Corp. M/S 635
100 Minuteman Road
Andover, MA 01810, USA
EMail: garys@pictel.com
Phone: +1 978 623 4324
Fax: +1 978 749 2804

Stephan Wenger
Technische Universitaet Berlin FB13
Sekr. FR 6-3
Franklinstr. 28/29
D-10587 Berlin, GERMANY
EMail: stewe@cs.tu-berlin.de
Phone: +49.30.314-73160
Fax: +49.30.314-25156

Chad Zhu
Intel Corp. M/S JF3-202
2111 NE 25th Avenue
Hillsboro, OR 97124, USA
EMail: czhu@ix.netcom.com
Phone: +1 503 264 6004
Fax: +1 503 264 1805

## 9. References

[1] Schulzrinne, H., Casner, S., Frederick, R., and V. Jacobson, "RTP : A Transport Protocol for Real-Time Applications", RFC 1889, January 1996.

[2] Schulzrinne, H., "RTP Profile for Audio and Video Conference with Minimal Control", RFC 1890, January 1996.

[3] "Video Coding for Low Bit Rate Communication," ITU-T Recommendation H.263, March 1996.

[4] "Video Coding for Low Bit Rate Communication," ITU-T Recommendation H.263, January 1998.

[5] Turletti, T. and C. Huitema, "RTP Payload Format for H.261 Video Streams", RFC 2032, October 1996.

[6] Zhu, C., "RTP Payload Format for H.263 Video Streams", RFC 2190, September 1997.

[7] S. Wenger, "Video Redundancy Coding in H.263+," Proc. Audio-Visual Services over Packet Networks, Aberdeen, U.K., September 1997.

## 10. Full Copyright Statement

Copyright (C) The Internet Society (1998). All Rights Reserved.

This document and translations of it may be copied and furnished to others, and derivative works that comment on or otherwise explain it or assist in its implementation may be prepared, copied, published and distributed, in whole or in part, without restriction of any kind, provided that the above copyright notice and this paragraph are included on all such copies and derivative works. However, this document itself may not be modified in any way, such as by removing the copyright notice or references to the Internet Society or other Internet organizations, except as needed for the purpose of developing Internet standards in which case the procedures for copyrights defined in the Internet Standards process must be followed, or as required to translate it into languages other than English.

The limited permissions granted above are perpetual and will not be revoked by the Internet Society or its successors or assigns.

This document and the information contained herein is provided on an "AS IS" basis and THE INTERNET SOCIETY AND THE INTERNET ENGINEERING TASK FORCE DISCLAIMS 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.