Network Working Group
Request for Comments: 2032
Category: Standards Track

T. Turletti
MIT
C. Huitema
Bellcore

October 1996

RTP Payload Format for H.261 Video Streams

Status of this Memo

Table of Contents

1. Abstract

This memo describes a scheme to packetize an H.261 video stream for transport using the Real-time Transport Protocol, RTP, with any of the underlying protocols that carry RTP.

This specification is a product of the Audio/Video Transport working group within the Internet Engineering Task Force. Comments are solicited and should be addressed to the working group's mailing list at rem-conf@es.net and/or the authors.

2. Purpose of this document

The ITU-T recommendation H.261 [6] specifies the encodings used by ITU-T compliant video-conference codecs. Although these encodings were originally specified for fixed data rate ISDN circuits, experiments [3],[8] have shown that they can also be used over packet-switched networks such as the Internet.

The purpose of this memo is to specify the RTP payload format for encapsulating H.261 video streams in RTP [1].

3. Structure of the packet stream

3.1. Overview of the ITU-T recommendation H.261

The H.261 coding is organized as a hierarchy of groupings. The video stream is composed of a sequence of images, or frames, which are themselves organized as a set of Groups of Blocks (GOB). Note that H.261 "pictures" are referred as "frames" in this document. Each GOB holds a set of 3 lines of 11 macro blocks (MB). Each MB carries information on a group of 16x16 pixels: luminance information is specified for 4 blocks of 8x8 pixels, while chrominance information is given by two "red" and "blue" color difference components at a resolution of only 8x8 pixels. These components and the codes representing their sampled values are as defined in the ITU-R Recommendation 601 [7].

This grouping is used to specify information at each level of the hierarchy:

- At the frame level, one specifies information such as the delay from the previous frame, the image format, and various indicators.

- At the GOB level, one specifies the GOB number and the default quantifier that will be used for the MBs.

- At the MB level, one specifies which blocks are present and which did not change, and optionally a quantifier and motion vectors.

Blocks which have changed are encoded by computing the discrete cosine transform (DCT) of their coefficients, which are then quantized and Huffman encoded (Variable Length Codes).

The H.261 Huffman encoding includes a special "GOB start" pattern, composed of 15 zeroes followed by a single 1, that cannot be imitated by any other code words. This pattern is included at the beginning of

each GOB header (and also at the beginning of each frame header) to mark the separation between two GOBs, and is in fact used as an indicator that the current GOB is terminated. The encoding also includes a stuffing pattern, composed of seven zeroes followed by four ones; that stuffing pattern can only be entered between the encoding of MBs, or just before the GOB separator.

3.2. Considerations for packetization

H.261 codecs designed for operation over ISDN circuits produce a bit stream composed of several levels of encoding specified by H.261 and companion recommendations. The bits resulting from the Huffman encoding are arranged in 512-bit frames, containing 2 bits of synchronization, 492 bits of data and 18 bits of error correcting code. The 512-bit frames are then interlaced with an audio stream and transmitted over px64 kbps circuits according to specification H.221 [5].

When transmitting over the Internet, we will directly consider the output of the Huffman encoding. All the bits produced by the Huffman encoding stage will be included in the packet. We will not carry the 512-bit frames, as protection against bit errors can be obtained by other means. Similarly, we will not attempt to multiplex audio and video signals in the same packets, as UDP and RTP provide a much more efficient way to achieve multiplexing.

Directly transmitting the result of the Huffman encoding over an unreliable stream of UDP datagrams would, however, have poor error resistance characteristics. The result of the hierachical structure of H.261 bit stream is that one needs to receive the information present in the frame header to decode the GOBs, as well as the information present in the GOB header to decode the MBs. Without precautions, this would mean that one has to receive all the packets that carry an image in order to properly decode its components.

If each image could be carried in a single packet, this requirement would not create a problem. However, a video image or even one GOB by itself can sometimes be too large to fit in a single packet. Therefore, the MB is taken as the unit of fragmentation. Packets must start and end on a MB boundary, i.e. a MB cannot be split across multiple packets. Multiple MBs may be carried in a single packet when they will fit within the maximal packet size allowed. This practice is recommended to reduce the packet send rate and packet overhead.

To allow each packet to be processed independently for efficient resynchronization in the presence of packet losses, some state information from the frame header and GOB header is carried with each

packet to allow the MBs in that packet to be decoded. This state information includes the GOB number in effect at the start of the packet, the macroblock address predictor (i.e. the last MBA encoded in the previous packet), the quantizer value in effect prior to the start of this packet (GQUANT, MQUANT or zero in case of a beginning of GOB) and the reference motion vector data (MVD) for computing the true MVDs contained within this packet. The bit stream cannot be fragmented between a GOB header and MB 1 of that GOB.

Moreover, since the compressed MB may not fill an integer number of octets, the data header contains two three-bit integers, SBIT and EBIT, to indicate the number of unused bits in the first and last octets of the H.261 data, respectively.

4. Specification of the packetization scheme

4.1. Usage of RTP

The H.261 information is carried as payload data within the RTP protocol. The following fields of the RTP header are specified:

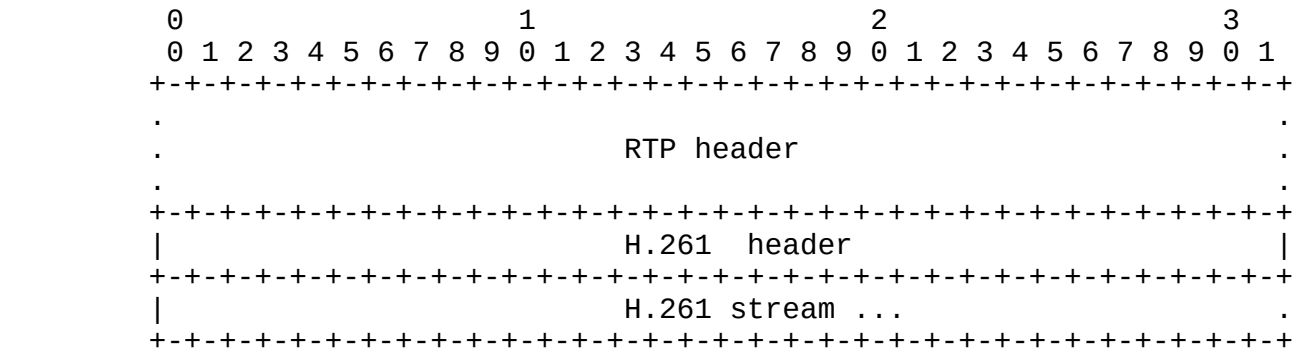
- The payload type should specify H.261 payload format (see the companion RTP profile document RFC 1890).

- The RTP timestamp encodes the sampling instant of the first video image contained in the RTP data packet. If a video image occupies more than one packet, the timestamp will be the same on all of those packets. Packets from different video images must have different timestamps so that frames may be distinguished by the timestamp. For H.261 video streams, the RTP timestamp is based on a 90kHz clock. This clock rate is a multiple of the natural H.261 frame rate (i.e. 30000/1001 or approx. 29.97 Hz). That way, for each frame time, the clock is just incremented by the multiple and this removes inaccuracy in calculating the timestamp. Furthermore, the initial value of the timestamp is random (unpredictable) to make known-plaintext attacks on encryption more difficult, see RTP [1]. Note that if multiple frames are encoded in a packet (e.g. when there are very little changes between two images), it is necessary to calculate display times for the frames after the first using the timing information in the H.261 frame header. This is required because the RTP timestamp only gives the display time of the first frame in the packet.

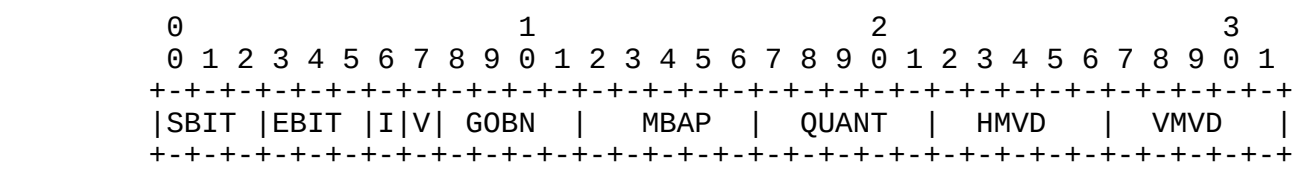
- The marker bit of the RTP header is set to one in the last packet of a video frame, and otherwise, must be

zero. Thus, it is not necessary to wait for a following packet (which contains the start code that terminates the current frame) to detect that a new frame should be displayed.

The H.261 data will follow the RTP header, as in:



The H.261 header is defined as following:



The fields in the H.261 header have the following meanings:

Start bit position (SBIT): 3 bits
Number of most significant bits that should be ignored in the first data octet.

End bit position (EBIT): 3 bits
Number of least significant bits that should be ignored in the last data octet.

INTRA-frame encoded data (I): 1 bit
Set to 1 if this stream contains only INTRA-frame coded blocks. Set to 0 if this stream may or may not contain INTRA-frame coded blocks. The sense of this bit may not change during the course of the RTP session.

Motion Vector flag (V): 1 bit
Set to 0 if motion vectors are not used in this stream. Set to 1 if motion vectors may or may not be used in this stream. The sense of this bit may not change during the course of the session.

GOB number (GOBN): 4 bits
Encodes the GOB number in effect at the start of the packet. Set to 0 if the packet begins with a GOB header.

Macroblock address predictor (MBAP): 5 bits
Encodes the macroblock address predictor (i.e. the last MBA encoded in the previous packet). This predictor ranges from 0-32 (to predict the valid MBAs 1-33), but because the bit stream cannot be fragmented between a GOB header and MB 1, the predictor at the start of the packet can never be 0. Therefore, the range is 1-32, which is biased by -1 to fit in 5 bits. For example, if MBAP is 0, the value of the MBA predictor is 1. Set to 0 if the packet begins with a GOB header.

Quantizer (QUANT): 5 bits
Quantizer value (MQUANT or GQUANT) in effect prior to the start of this packet. Set to 0 if the packet begins with a GOB header.

Horizontal motion vector data (HMVD): 5 bits
Reference horizontal motion vector data (MVD). Set to 0 if V flag is 0 or if the packet begins with a GOB header, or when the MTYPE of the last MB encoded in the previous packet was not MC. HMVD is encoded as a 2’s complement number, and ‘10000’ corresponding to the value -16 is forbidden (motion vector fields range from +/-15).

Vertical motion vector data (VMVD): 5 bits
Reference vertical motion vector data (MVD). Set to 0 if V flag is 0 or if the packet begins with a GOB header, or when the MTYPE of the last MB encoded in the previous packet was not MC. VMVD is encoded as a 2’s complement number, and ‘10000’ corresponding to the value -16 is forbidden (motion vector fields range from +/-15).

Note that the I and V flags are hint flags, i.e. they can be inferred from the bit stream. They are included to allow decoders to make optimizations that would not be possible if these hints were not provided before bit stream was decoded. Therefore, these bits cannot change for the duration of the stream. A conformant implementation can always set V=1 and I=0.

4.2. Recommendations for operation with hardware codecs

Packetizers for hardware codecs can trivially figure out GOB boundaries using the GOB-start pattern included in the H.261 data. (Note that software encoders already know the boundaries.) The

cheapest packetization implementation is to packetize at the GOB level all the GOBs that fit in a packet. But when a GOB is too large, the packetizer has to parse it to do MB fragmentation. (Note that only the Huffman encoding must be parsed and that it is not necessary to fully decompress the stream, so this requires relatively little processing; example implementations can be found in some public H.261 codecs such as IVS [4] and VIC [9].) It is recommended that MB level fragmentation be used when feasible in order to obtain more efficient packetization. Using this fragmentation scheme reduces the output packet rate and therefore reduces the overhead.

At the receiver, the data stream can be depacketized and directed to a hardware codec's input. If the hardware decoder operates at a fixed bit rate, synchronization may be maintained by inserting the stuffing pattern between MBs (i.e., between packets) when the packet arrival rate is slower than the bit rate.

5. Packet loss issues

On the Internet, most packet losses are due to network congestion rather than transmission errors. Using UDP, no mechanism is available at the sender to know if a packet has been successfully received. It is up to the application, i.e. coder and decoder, to handle the packet loss. Each RTP packet includes a a sequence number field which can be used to detect packet loss.

H.261 uses the temporal redundancy of video to perform compression. This differential coding (or INTER-frame coding) is sensitive to packet loss. After a packet loss, parts of the image may remain corrupt until all corresponding MBs have been encoded in INTRA-frame mode (i.e. encoded independently of past frames). There are several ways to mitigate packet loss:

(1) One way is to use only INTRA-frame encoding and MB level conditional replenishment. That is, only MBs that change (beyond some threshold) are transmitted.

(2) Another way is to adjust the INTRA-frame encoding refreshment rate according to the packet loss observed by the receivers. The H.261 recommendation specifies that a MB is INTRA-frame encoded at least every 132 times it is transmitted. However, the INTRA-frame refreshment rate can be raised in order to speed the recovery when the measured loss rate is significant.

(3) The fastest way to repair a corrupted image is to request an INTRA-frame coded image refreshment after a packet loss is detected. One means to accomplish this is for the

decoder to send to the coder a list of packets lost. The coder can decide to encode every MB of every GOB of the following video frame in INTRA-frame mode (i.e. Full INTRA-frame encoded), or if the coder can deduce from the packet sequence numbers which MBs were affected by the loss, it can save bandwidth by sending only those MBs in INTRA-frame mode. This mode is particularly efficient in point-to-point connection or when the number of decoders is low. The next section specifies how the refresh function may be implemented.

Note that the method (1) is currently implemented in the VIC videoconferencing software [9]. Methods (2) and (3) are currently implemented in the IVS videoconferencing software [4].

5.1. Use of optional H.261-specific control packets

This specification defines two H.261-specific RTCP control packets, "Full INTRA-frame Request" and "Negative Acknowledgement", described in the next section. Their purpose is to speed up refreshment of the video in those situations where their use is feasible. Support of these H.261-specific control packets by the H.261 sender is optional; in particular, early experiments have shown that the usage of this feature could have very negative effects when the number of sites is very large. Thus, these control packets should be used with caution.

The H.261-specific control packets differ from normal RTCP packets in that they are not transmitted to the normal RTCP destination transport address for the RTP session (which is often a multicast address). Instead, these control packets are sent directly via unicast from the decoder to the coder. The destination port for these control packets is the same port that the coder uses as a source port for transmitting RTP (data) packets. Therefore, these packets may be considered "reverse" control packets.

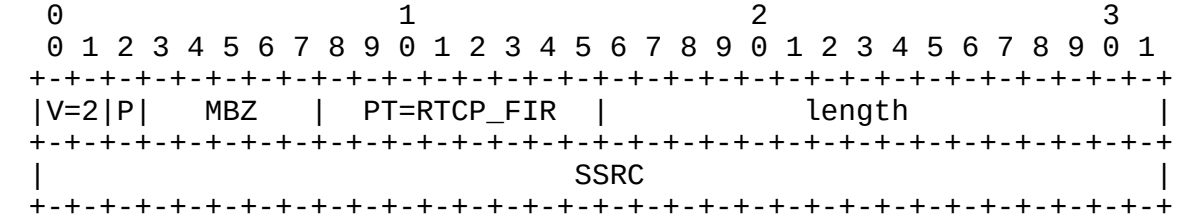
As a consequence, these control packets may only be used when no RTP mixers or translators intervene in the path from the coder to the decoder. If such intermediate systems do intervene, the address of the coder would no longer be present as the network-level source address in packets received by the decoder, and in fact, it might not be possible for the decoder to send packets directly to the coder.

Some reliable multicast protocols use similar NACK control packets transmitted over the normal multicast distribution channel, but they typically use random delays to prevent a NACK implosion problem [2]. The goal of such protocols is to provide reliable multicast packet delivery at the expense of delay, which is appropriate for applications such as a shared whiteboard.

On the other hand, interactive video transmission is more sensitive to delay and does not require full reliability. For video applications it is more effective to send the NACK control packets as soon as possible, i.e. as soon as a loss is detected, without adding any random delays. In this case, multicasting the NACK control packets would generate useless traffic between receivers since only the coder will use them. But this method is only effective when the number of receivers is small. e.g. in IVS [4] the H.261 specific control packets are used only in point-to-point connections or in point-to-multipoint connections when there are less than 10 participants in the conference.

5.2. H.261 control packets definition

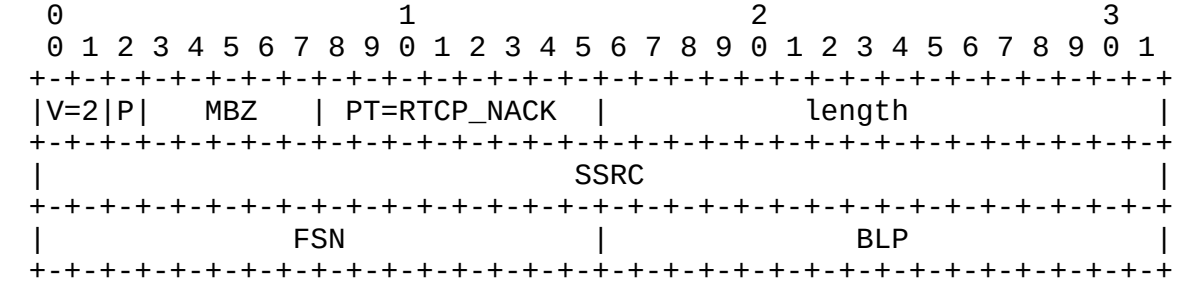
5.2.1. Full INTRA-frame Request (FIR) packet



This packet indicates that a receiver requires a full encoded image in order to either start decoding with an entire image or to refresh its image and speed the recovery after a burst of lost packets. The receiver requests the source to force the next image in full "INTRA-frame" coding mode, i.e. without using differential coding. The various fields are defined in the RTP specification [1]. SSRC is the synchronization source identifier for the sender of this packet. The value of the packet type (PT) identifier is the constant RTCP_FIR (192).

5.2.2. Negative ACKnowledgements (NACK) packet

The format of the NACK packet is as follow:



The various fields T, P, PT, length and SSRC are defined in the RTP specification [1]. The value of the packet type (PT) identifier is the constant RTCP_NACK (193). SSRC is the synchronization source identifier for the sender of this packet.

The two remaining fields have the following meanings:

First Sequence Number (FSN): 16 bits
Identifies the first sequence number lost.

Bitmask of following lost packets (BLP): 16 bits
A bit is set to 1 if the corresponding packet has been lost, and set to 0 otherwise. BLP is set to 0 only if no packet other than that being NACKed (using the FSN field) has been lost. BLP is set to 0x00001 if the packet corresponding to the FSN and the following packet have been lost, etc.

6. Security Considerations

Security issues are not discussed in this memo.

Authors' Addresses

Thierry Turletti
INRIA - RODEO Project
2004 route des Lucioles
BP 93, 06902 Sophia Antipolis
FRANCE

EMail: turletti@sophia.inria.fr

Christian Huitema
MCC 1J236B Bellcore
445 South Street
Morristown, NJ 07960-6438

EMail: huitema@bellcore.com

Acknowledgements

This memo is based on discussion within the AVT working group chaired by Stephen Casner. Steve McCanne, Stephen Casner, Ronan Flood, Mark Handley, Van Jacobson, Henning G. Schulzrinne and John Wroclawski provided valuable comments. Stephen Casner and Steve McCanne also helped greatly with getting this document into readable form.

References

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

[2] Sridhar Pingali, Don Towsley and James F. Kurose, A comparison of sender-initiated and receiver-initiated reliable multicast protocols, IEEE GLOBECOM ’94.

[3] Thierry Turletti, H.261 software codec for videoconferencing over the Internet INRIA Research Report no 1834, January 1993.

[4] Thierry Turletti, INRIA Videoconferencing tool (IVS), available by anonymous ftp from zenon.inria.fr in the "rodeo/ivs/last_version" directory. See also URL <http://www.inria.fr/rodeo/ivs.html>.

[5] Frame structure for Audiovisual Services for a 64 to 1920 kbps Channel in Audiovisual Services ITU-T (International Telecommunication Union - Telecommunication Standardisation Sector) Recommendation H.221, 1990.

[6] Video codec for audiovisual services at p x 64 kbit/s ITU-T (International Telecommunication Union - Telecommunication Standardisation Sector) Recommendation H.261, 1993.

[7] Digital Methods of Transmitting Television Information ITU-R (International Telecommunication Union - Radiocommunication Standardisation Sector) Recommendation 601, 1986.

[8] M.A Sasse, U. Bilting, C-D Schulz, T. Turletti, Remote Seminars through MultiMedia Conferencing: Experiences from the MICE project, Proc. INET’94/JENC5, Prague, June 1994, pp. 251/1-251/8.

[9] Steve MacCanne, Van Jacobson, VIC Videoconferencing tool, available by anonymous ftp from ee.lbl.gov in the "conferencing/vic" directory.