

Network Working Group  
Request for Comments: 4587  
Obsoletes: 2032  
Category: Standards Track

R. Even  
Polycom  
August 2006

## RTP Payload Format for H.261 Video Streams

### 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 Internet Society (2006).

### 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.

The memo also describes the syntax and semantics of the Session Description Protocol (SDP) parameters needed to support the H.261 video codec. A media type registration is included for this payload format.

This specification obsoletes RFC 2032.

## Table of Contents

1. Introduction .....	3
2. Terminology .....	3
3. Structure of the Packet Stream .....	3
3.1. Overview of the ITU-T Recommendation H.261 .....	3
3.2. Considerations for Packetization .....	4
4. Specification of the Packetization Scheme .....	5
4.1. Usage of RTP .....	5
4.2. Recommendations for Operation with Hardware Codecs .....	8
5. Packet Loss Issues .....	9
6. IANA Considerations .....	10
6.1. Media Type Registrations .....	10
6.1.1. Registration of MIME Media Type video/H261 .....	10
6.2. SDP Parameters .....	12
6.2.1. Usage with the SDP Offer Answer Model .....	12
7. Backward Compatibility to RFC 2032 .....	13
7.1. Optional H.261-Specific Control Packets .....	13
7.2. New SDP Optional Parameters .....	13
8. Security Considerations .....	14
9. Acknowledgements .....	14
10. Changes from RFC 2032 .....	14
11. References .....	15
11.1. Normative References .....	15
11.2. Informative References .....	15

## 1. Introduction

ITU-T Recommendation H.261 [H261] specifies the encoding used by ITU-T-compliant video-conference codecs. Although this encoding was originally specified for fixed-data rate Integrated Services Digital Network (ISDN) circuits, experiments [INRIA], [MICE] 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 [RFC3550].

This document obsoletes RFC 2032 and updates the "video/h261" media type that was registered in RFC 3555.

## 2. 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.

## 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 to 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, whereas 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 ITU-R Recommendation 601 [BT601].

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 that 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, which is a word of 16 bits, 0000 0000 0000 0001. 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 zero bits followed by four bits with a value of one; 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 px 64 kbps circuits according to specification H.221 [H221].

For 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 suitable 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 hierarchical structure of the 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 decode its components properly.

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 an MB boundary; that is, an 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 macroblock address (MBA) encoded in the previous packet), the quantizer value in effect prior to the start of this packet (GQUANT, MQQUANT, or zero in the 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 3-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

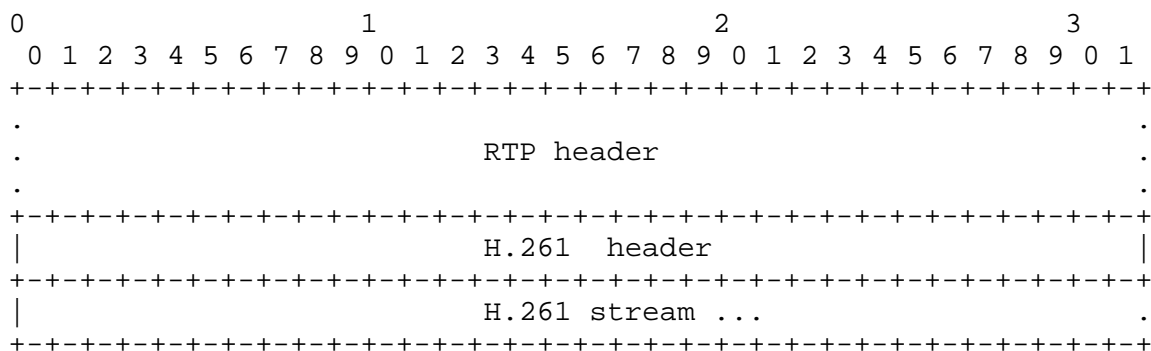
Each RTP packet starts with a fixed RTP header, as explained in RFC 3550 [RFC3550]. The following fields of the RTP fixed header used for H.261 video streams are further emphasized here:

- Payload type. The assignment of an RTP payload type for this packet format is outside the scope of this document and will not be specified here. It is expected that the RTP profile for a particular class of applications will assign a payload type for this encoding, or, if that is not done, then a payload type in the dynamic range shall be chosen.
- 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 SHALL be the same on all of those packets. Packets from different video images MUST have a different timestamp so that frames may be distinguished by the timestamp. For H.261 video streams, the RTP timestamp is based on a 90-kHz clock. This clock rate is a multiple of the natural H.261 frame rate (i.e., 30000/1001 or approximately 29.97 Hz). That way,

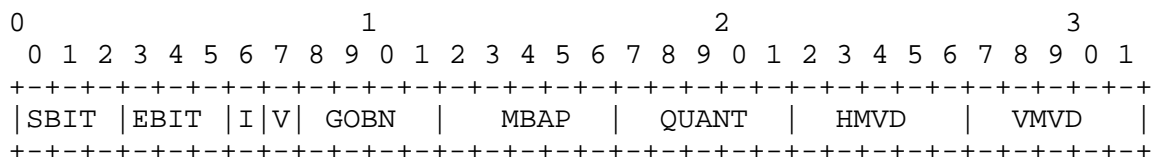
for each frame time, the clock is just incremented by the multiple, and this removes inaccuracy in calculating the timestamp. Furthermore, the initial value of the timestamp MUST be random (unpredictable) to make known-plaintext attacks on encryption more difficult; see RTP [RFC3550]. Note that if multiple frames are encoded in a packet (e.g., when there are very few 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 MUST be set to one in the last packet of a video frame; otherwise, it 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 SHALL follow the RTP header, as in the following:



The H.261 header is defined as follows:



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 meaning of this bit should not be changed 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 meaning of this bit should not be changed 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 shall not 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 motion compensation (MC). HMVD is encoded as a 2s 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 2s complement number, and '10000' corresponding to the value -16 SHALL not be used (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 the bit stream was decoded. Therefore, these bits cannot change for the duration of the stream. A conforming implementation can always set V=1 and I=0.

The H.261 stream SHALL be used without BCH error correction and without error correction framing.

#### 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 decompress the stream fully, so this requires relatively little processing; examples of implementations can be found in some public H.261 codecs, such as IVS [IVS] and VIC [VIC].) 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 to transmission errors. Using UDP, no mechanism is available at the sender to know whether 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 sequence number field that 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 an MB be 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). 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 H.261-specific control packets FIR and NACK, as described in RFC 2032, SHALL NOT be used to request image refreshment. Old implementations are encouraged to use the methods described in this section. Image refreshment may be needed due to packet loss or due to application requirements. An example of application requirement may be the change of the speaker in a voice-activated multipoint video switching conference. There are two methods that can be used for requesting image refreshment. The first method is by using the Extended RTP Profile for RTCP-based Feedback and sending RTCP generic

control packets, as described in RFC 4585 [RFC4585]. The second method is by using application protocol-specific commands, such as H.245 [ITU.H245] FastUpdateRequest.

## 6. IANA Considerations

This section updates the H.261 media type described in RFC 3555 [RFC3555].

This section specifies optional parameters that MAY be used to select optional features of the payload format. The parameters are specified here as part of the MIME subtype registration for the ITU-T H.261 codec. A mapping of the parameters into the Session Description Protocol (SDP) [RFC4566] is also provided for those applications that use SDP. Multiple parameters SHOULD be expressed as a media type string, in the form of a semicolon-separated list of parameters.

### 6.1. 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]. This registration uses the template defined in RFC 4288 [RFC4288]

#### 6.1.1. Registration of MIME Media Type video/H261

MIME media type name: video

MIME subtype name: H261

Required parameters: None

Optional parameters:

CIF. This parameter has the format of parameter=value. It describes the maximum supported frame rate for CIF resolution. Permissible values are integer values 1 to 4, and it means that the maximum rate is 29.97/specified value.

QCIF. This parameter has the format of parameter=value. It describes the maximum supported frame rate for QCIF resolution. Permissible values are integer values 1 to 4, and it means that the maximum rate is 29.97/specified value.

D. Specifies support for still image graphics according to H.261, annex D. If supported, the parameter value SHALL be "1". If not supported, the parameter SHOULD NOT be used 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 8

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.261 stream without annex D. Most decoders support at least QCIF resolutions, and they are expected to be available in almost every H.261-based video application.

Published specification: RFC 4587

Applications that use this media type:

Audio and video streaming and conferencing applications.

Additional information: None

Person and email address to contact for further information:

Roni Even: [roni.even@polycom.co.il](mailto: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.

## 6.2. SDP Parameters

The MIME media type video/H261 string is 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 H261 (the MIME subtype).
- o The clock rate in the "a=rtpmap" line MUST be 90000.
- o The optional parameters "CIF", "QCIF", and "D", if any, SHALL be included in the "a=fmtp" line of SDP. These parameters are expressed as a MIME media type string, in the form of as a semicolon-separated list of parameters

### 6.2.1. Usage with the SDP Offer Answer Model

When H.261 is offered over RTP using SDP in an Offer/Answer model [RFC3264] the following considerations are necessary.

Codec options: (D) This option MUST NOT appear unless the sender of this SDP message is able to decode this option. This option SHALL be considered a receiver's capability even when it is sent in a "sendonly" offer.

Picture sizes and MPI:

Supported picture sizes and their corresponding minimum picture interval (MPI) information for H.261 can be combined. All picture sizes may be advertised to the other party, or only a subset of it. Using the recvonly or sendrev direction attribute, a terminal SHOULD announce those picture sizes (with their MPIs) that it is willing to receive. For example, CIF=2 means that receiver can receive a CIF picture and that the frame rate SHALL be less than 15 frames per second.

When the direction attribute is sendonly, the parameters describe the capabilities of the stream that the sender can produce.

Implementations following this specification SHALL specify at least one supported picture size.

If the receiver does not specify the picture size/MPI parameter, then it is safe to assume that it is an implementation that follows RFC 2032. In that case, it is RECOMMENDED to assume that such a receiver is able to support reception of QCIF resolution with MPI=1.

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

An example of media representation in SDP is as follows CIF at 15 frames per second, QCIF at 30 frames per second and annex D

```
m=video 49170/2 RTP/AVP 31
a=rtpmap:31 H261/90000
a=fmtp:31 CIF=2;QCIF=1;D=1
```

This means that the sender of this message can decode an H.261 bit stream with the following options and parameters: preferred resolution is CIF (its MPI is 2), but if that is not possible, then QCIF size is also supported. Still image using annex D MAY be used.

## 7. Backward Compatibility to RFC 2032

The current document replaces RFC 2032. This section will address the major backward compatibility issues.

### 7.1. Optional H.261-Specific Control Packets

RFC 2032 defined two H.261-specific RTCP control packets, "Full INTRA-frame Request" and "Negative Acknowledgement". Support of these control packets was optional. 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 encoder. The destination port for these control packets is the same port that the encoder uses as a source port for transmitting RTP (data) packets. Therefore, these packets may be considered "reverse" control packets. This memo suggests generic methods to address the same requirement. The authors of the documents are not aware of products that support these control packets. Since these are optional features, new implementations SHALL ignore them, and they SHALL NOT be used by new implementations.

### 7.2. New SDP Optional Parameters

The document adds new optional parameters to the H261 payload type. Since these are optional parameters, we expect that old implementations ignore these parameters, whereas new implementations that receive the H261 payload type capabilities with no parameters will assume that it is an old implementation and will send H.261 at QCIF resolution and 30 frames per second.

## 8. 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 in any appropriate RTP profile (e.g., [RFC3551]). This implies that confidentiality of the media streams is achieved by encryption. SRTP [RFC3711] may be used to provide both encryption and integrity protection of RTP flow. Because the data compression used with this payload format is applied end to end, encryption will 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. H.261 is vulnerable to such attacks because it is possible for an attacker to generate RTP packets containing frames that affect the decoding process of future frames. Therefore, the usage of data origin authentication and data integrity protection of at least the RTP packet is RECOMMENDED; for example, with SRTP.

Note that the appropriate mechanism to ensure confidentiality and integrity of RTP packets and their payloads is very dependent on the application and on the transport and signaling protocols employed. Thus, although SRTP is given as an example above, other possible choices exist.

## 9. Acknowledgements

This is to acknowledge the authors of RFC 2032, Thierry Turletti and Christian Huitema. Special thanks for the work done by Petri Koskelainen from Nokia and Nermeen Ismail from Cisco, who helped with drafting the text for the new MIME types.

## 10. Changes from RFC 2032

The changes from the RFC 2032 are:

1. The H.261 MIME type is now in the payload specification.
2. Added optional parameters to the H.261 MIME type
3. Deprecated the H.261 specific control packets
4. Editorial changes to be in line with RFC editing procedures

## 11. References

### 11.1. Normative References

- [H261] International Telecommunications Union, "Video codec for audiovisual services at px 64 kbit/s", ITU Recommendation H.261, March 1993.
- [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.
- [RFC3264] Rosenberg, J. and H. Schulzrinne, "An Offer/Answer Model with Session Description Protocol (SDP)", RFC 3264, June 2002.
- [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.

### 11.2. Informative References

- [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.
- [ITU.H245] International Telecommunications Union, "CONTROL PROTOCOL FOR MULTIMEDIA COMMUNICATION", ITU Recommendation H.245, 2003.
- [INRIA] Turletti, T., "H.261 software codec for videoconferencing over the Internet", INRIA Research Report 1834, January 1993.

- [IVS] Turletti, T., "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>>.
- [BT601] International Telecommunications Union, "Studio encoding parameters of digital television for standard 4:3 and wide-screen 16:9 aspect ratios", ITU-R Recommendation BT.601-5, October 1995.
- [MICE] Sasse, MA., Bilting, U., Schultz, CD., and T. Turletti, "Remote Seminars through MultiMedia Conferencing: Experiences from the MICE project", Proc. INET'94/JENC5, Prague pp. 251/1-251/8, June 1994.
- [VIC] MacCanne, S., "VIC Videoconferencing tool, available by anonymous ftp from ee.lbl.gov in the "conferencing/vic" directory".
- [RFC3711] Baugher, M., McGrew, D., Naslund, M., Carrara, E., and K. Norrman, "The Secure Real-time Transport Protocol (SRTP)", RFC 3711, March 2004.
- [H221] International Telecommunications Union, "Frame structure for a 64 to 1920 kbit/s channel in audiovisual teleservices", ITU Recommendation H.221, May 1999.

#### Author's Address

Roni Even  
Polycom  
94 Derech Em Hamoshavot  
Petach Tikva 49130  
Israel

EMail: [roni.even@polycom.co.il](mailto:roni.even@polycom.co.il)

## Full Copyright Statement

Copyright (C) The Internet Society (2006).

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 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](mailto:ietf-ipr@ietf.org).

## Acknowledgement

Funding for the RFC Editor function is provided by the IETF Administrative Support Activity (IASA).

