Internet Engineering Task Force
INTERNET-DRAFT
draft-ietf-avt-profile-new-05.ps

AVT WG
Schulzrinne
Columbia U.

February 26, 1999
Expires: August 26, 1999

RTP Profile for Audio and Video Conferences with Minimal Control

Status of this Memo

This document is an Internet-Draft and is in full conformance with all provisions of Section 10 of RFC2026.

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF), its areas, and its working groups. Note that other groups may also distribute working documents as Internet-Drafts.

Internet-Drafts are draft documents valid for a maximum of six months and may be updated, replaced, or obsoleted by other documents at any time. It is inappropriate to use Internet-Drafts as reference material or to cite them other than as "work in progress."

The list of current Internet-Drafts can be accessed at http://www.ietf.org/ietf/1id-abstracts.txt

The list of Internet-Draft Shadow Directories can be accessed at http://www.ietf.org/shadow.html.

Abstract

This memorandum is a revision of RFC 1890 in preparation for advancement from Proposed Standard to Draft Standard status. Readers are encouraged to use the PostScript form of this draft to see where changes from RFC 1890 are marked by change bars.

This document describes a profile called "RTP/AVP" for the use of the real-time transport protocol (RTP), version 2, and the associated control protocol, RTCP, within audio and video multiparticipant conferences with minimal control. It provides interpretations of generic fields within the RTP specification suitable for audio and video conferences. In particular, this document defines a set of default mappings from payload type numbers to encodings.

This document also describes how audio and video data may be carried within RTP. It defines a set of standard encodings and their names when used within RTP. The descriptions provide pointers to reference implementations and the detailed standards. This document is meant as an aid for implementors of audio, video and other real-time multimedia applications.

Contents

Resolution of Open Issues

[Note to the RFC Editor: This section is to be deleted when this draft is published as an RFC but is shown here for reference during the Last Call. All RFC XXXX should be filled in with the number of the RTP specification RFC submitted for Draft Standard status, and all RFC YYYY should be filled in with the number of the draft specifying MIME registration of RTP payload types as it is submitted for Proposed Standard status.]

Readers are directed to Appendix 9, Changes from RFC 1890, for a listing of the changes that have been made in this draft. The changes from RFC 1890 are marked with change bars in the PostScript form of this draft.

The revisions in this draft are intended to be complete for Working Group last call. The following open issues from previous drafts have been addressed:

- The procedure for registering encoding names as MIME subtypes is outlined here and referenced in a separate RFC-to-be that may also serve to specify how (some of) the encodings here may be used with mail and other not-RTP transports.
- This profile follows the suggestion in the RTP spec that RTCP bandwidth may be specified separately from the session bandwidth and separately for active senders and passive receivers.
- No specific action is taken in this document to address generic payload formats; it is assumed that if any generic payload formats are developed, they can be specified in separate RFCs and that the session parameters they require for operation can be specified in the MIME registration of those formats.

1 Introduction

This profile defines aspects of RTP left unspecified in the RTP Version 2 protocol definition (RFC XXXX) [1]. This profile is intended for the use within audio and video conferences with minimal session control. In particular, no support for the negotiation of parameters or membership control is provided. The profile is expected to be useful in sessions where no negotiation or membership control are used (e.g., using the static payload types and the membership indications provided by RTCP), but this profile may also be useful in conjunction with a higher-level control protocol.

Use of this profile may be implicit in the use of the appropriate applications; there may be no explicit indication by port number, protocol identifier or the like. Applications such as session directories should refer to this profile as “RTP/AVP”.

Other profiles may make different choices for the items specified here.

This document also defines a set of encodings and payload formats for audio and video.

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 [2] and indicate requirement levels for implementations compliant with this RTP profile.

This draft defines the term *media type* as dividing encodings of audio and video content into three classes: audio, video and audio/video (interleaved).

2 RTP and RTCP Packet Forms and Protocol Behavior

The section “RTP Profiles and Payload Format Specification” of RFC XXXX enumerates a number of items that can be specified or modified in a profile. This section addresses these items. Generally, this profile follows the default and/or recommended aspects of the RTP specification.

RTP data header: The standard format of the fixed RTP data header is used (one marker bit).

Payload types: Static payload types are defined in Section 6.

RTP data header additions: No additional fixed fields are appended to the RTP data header.

RTP data header extensions: No RTP header extensions are defined, but applications operating under this profile MAY use such extensions. Thus, applications SHOULD NOT assume that the RTP header X bit is always zero and SHOULD be prepared to ignore the header extension. If a header extension is defined in the future, that definition MUST specify the contents of the first 16 bits in such a way that multiple different extensions can be identified.

RTCP packet types: No additional RTCP packet types are defined by this profile specification.

RTCP report interval: The suggested constants are to be used for the RTCP report interval calculation. Sessions operating under this profile MAY specify a separate parameter for the RTCP traffic bandwidth rather than using the default fraction of the session bandwidth. The RTCP traffic bandwidth may be divided into two separate session parameters for those participants which are active data senders and those which are not. Following the recommendation in the RTP specification [1] that 1/4 of the RTCP bandwidth be dedicated to data senders, the RECOMMENDED default values for these two parameters would be 1.25% and 3.75%, respectively. For a particular session, the RTCP bandwidth for non-data-senders MAY be set to zero when operating on unidirectional links or for sessions that don't require feedback on the quality of reception. The RTCP bandwidth for data senders SHOULD be kept non-zero so that sender reports can still be sent for inter-media synchronization and to identify the source by CNAME. The means by which the one or two session parameters for RTCP bandwidth are specified is beyond the scope of this memo.

SR/RR extension: No extension section is defined for the RTCP SR or RR packet.

SDES use: Applications MAY use any of the SDES items described in the RTP specification. While CNAME information MUST be sent every reporting interval, other items SHOULD only be sent every third reporting interval, with NAME sent seven out of eight times within that slot and the remaining SDES items cyclically taking up the eighth slot, as defined in Section 6.2.2 of the RTP specification. In other words, NAME is sent in RTCP packets 1, 4, 7, 10, 13, 16, 19, while, say, EMAIL is used in RTCP packet 22.

Security: The RTP default security services are also the default under this profile.

String-to-key mapping: A user-provided string ("pass phrase") is hashed with the MD5 algorithm to a 16-octet digest. An n-bit key is extracted from the digest by taking the first n bits from the digest. If several keys are needed with a total length of 128 bits or less (as for triple DES), they are extracted in order from that digest. The octet ordering is specified in RFC 1423, Section 2.2. (Note that some DES implementations require that the 56-bit key be expanded into 8 octets by inserting an odd parity bit in the most significant bit of the octet to go with each 7 bits of the key.)

It is RECOMMENDED that pass phrases be restricted to ASCII letters, digits, the hyphen, and white space to reduce the the chance of transcription errors when conveying keys by phone, fax, telex or email.

The pass phrase MAY be preceded by a specification of the encryption algorithm. Any characters up to the first slash (ASCII 0x2f) are taken as the name of the encryption algorithm. The encryption format specifiers SHOULD be drawn from RFC 1423 or any additional identifiers registered with IANA. If no slash is present, DES-CBC is assumed as default. The encryption algorithm specifier is case sensitive.

The pass phrase typed by the user is transformed to a canonical form before applying the hash algorithm. For that purpose, we define 'white space' to be the ASCII space, formfeed, newline, carriage return, tab, or vertical tab as well as all characters contained in the Unicode space characters table. The transformation consists of the following steps: (1) convert the input string to the ISO 10646 character set, using the UTF-8 encoding as specified in Annex P to ISO/IEC 10646-1:1993 (ASCII characters require no mapping, but ISO 8859-1 characters do); (2) remove leading and trailing white space characters; (3) replace one or more contiguous white space characters by a single space (ASCII or UTF-8 0x20); (4) convert all letters to lower case and replace sequences of characters and non-spacing accents with a single character, where possible. A minimum length of 16 key characters (after applying the transformation) SHOULD be enforced by the application, while applications MUST allow up to 256 characters of input.

Underlying protocol: The profile specifies the use of RTP over unicast and multicast UDP as well as TCP. (This does not preclude the use of these definitions when RTP is carried by other lower-layer protocols.)

Transport mapping: The standard mapping of RTP and RTCP to transport-level addresses is used.

Encapsulation: A minimal TCP encapsulation is defined.

3 Registering Additional Encodings with IANA

This profile defines a set of encodings and assigns names to them. These encodings are comprised of a particular media data compression or representation plus an payload format for encapsulation within RTP. It is expected that additional encodings beyond the set defined here will be created in the future. These additional encodings may be registered with the Internet Assigned Numbers Authority (IANA) as MIME subtype names under the "audio" and "video" MIME types through the MIME registration procedure as specified in RFC 2048 [3].

The MIME registration procedure was originally designed for transport of multimedia information via asynchronous Internet mail, but the MIME namespace now provides identification for other transport modes as well. Some additional parameters are required in the registration procedure to specify how a particular media type is transported over RTP. These extensions to the registration procedure, along with registrations for the encodings defined in this memo, are given in RFC YYYY [4].

The encoding definitions contained in this memo provide examples of the information needed:

- the name of the encoding; the names defined here are 3 or 4 characters long to allow a compact representation if needed;
- a description of encoding, including in particular the RTP timestamp clock rate (or multiple rates for audio encodings with multiple sampling rates), as may be provided by a separte RTP payload format specification;
- any additional or optional parameters, such as the number of channels for an audio encoding;
- a reference to a further description of the data compression format itself, if available;
- contact information and an indication of who has change control over the encoding (this is part of all MIME registrations).

In addition to assigning names to encodings, this profile also also assigns static RTP payload types to some of them. However, the payload type number space is relatively small and cannot accommodate assignments for all existing and future encodings. During the early stages of RTP development, it was necessary to use statically assigned payload types because no other mechanism had been specified to bind encodings to payload types. It was anticipated that non-RTP means beyond the scope of this memo (such as directory services or invitation protocols) would be specified to establish a dynamic mapping between a payload type and an encoding. Now, mechanisms for defining dynamic payload type bindings have been specified in the Session Description Protocol (SDP), RFC 2327 [5], and in other protocols such as ITU-T recommendation H.323/H.245. These mechanisms associate the registered name of the encoding/payload format, along with any additional required parameters such as the RTP timestamp clock rate and number of channels, to a payload type number. This association is effective only for the duration of the RTP session in which the dynamic payload type binding is made. This association applies only to the RTP session for which it is made, thus the numbers can be re-used for different encodings in different sessions so the number space limitation is avoided.

This profile reserves payload type numbers in the range 96-127 exclusively for dynamic assignment. Applications should first use values in this range for dynamic payload types. Only applications which need to define more than 32 dynamic payload types may bind codes below 96, in which case it is RECOMMENDED that unassigned payload type numbers be used first. However, the statically assigned payload types are default bindings and may be dynamically bound to new encodings if needed. Redefining payload types below 96 may cause incorrect operation if an attempt is made to join a session without obtaining session description information that defines the dynamic payload types.

Dynamic payload types SHOULD NOT be used without a well-defined mechanism to indicate the mapping. Systems that expect to interoperate with others operating under this profile SHOULD NOT make their own assignments of proprietary encodings to particular, fixed payload types.

This specification establishes the policy that no additional static payload types will be assigned beyond the ones defined in this document. Establishing this policy avoids the problem of trying to create a set of criteria for accepting static assignments and encourages the implementation and deployment of the dynamic payload type mechanisms.

4 Audio

4.1 Encoding-Independent Rules

For applications which send either no packets or comfort-noise packets during silence, the first packet of a talkspurt, that is, the first packet after a silence period, SHOULD be distinguished by setting the marker bit in the RTP data header to one. The marker bits in all other packets is zero. The beginning of a talkspurt MAY be used to adjust the playout delay to reflect changing network delays. Applications without silence suppression MUST set the marker bit to zero.

The RTP clock rate used for generating the RTP timestamp is independent of the number of channels and the encoding; it equals the number of sampling periods per second. For N-channel encodings, each sampling period (say, 1/8000 of a second) generates N samples. (This terminology is standard, but somewhat confusing, as the total number of samples generated per second is then the sampling rate times the channel count.)

If multiple audio channels are used, channels are numbered left-to-right, starting at one. In RTP audio packets, information from lower-numbered channels precedes that from higher-numbered channels. For more than two channels, the convention followed by the AIFF-C audio interchange format SHOULD be followed [6], using the following notation:

l	left
r	right
c	center
S	surround
F	front
R	rear

channels	description	channel 1	2	3	4	5	6
2	stereo	l	r				
3		l	r	c			
4	quadrophonic	Fl	Fr	Rl	Rr		
4		l	c	r	S		
5		Fl	Fr	Fc	Sl	Sr	
6		l	lc	c	r	rc	S

Samples for all channels belonging to a single sampling instant MUST be within the same packet. The interleaving of samples from different channels depends on the encoding. General guidelines are given in Section 4.3 and 4.4.

The sampling frequency SHOULD be drawn from the set: 8000, 11025, 16000, 22050, 24000, 32000, 44100 and 48000 Hz. (Older Apple Macintosh computers had a native sample rate of 22254.54 Hz, which can be converted to 22050 with acceptable quality by dropping 4 samples in a 20 ms frame.) However, most audio encodings are defined for a more restricted set of sampling frequencies. Receivers SHOULD be prepared to accept multi-channel audio, but MAY choose to only play a single channel.

4.2 Operating Recommendations

The following recommendations are default operating parameters. Applications SHOULD be prepared to handle other values. The ranges given are meant to give guidance to application writers, allowing a set of applications conforming to these guidelines to interoperate without additional negotiation. These guidelines are not intended to restrict operating parameters for applications that can negotiate a set of interoperable parameters, e.g., through a conference control protocol.

For packetized audio, the default packetization interval SHOULD have a duration of 20 ms or one frame, whichever is longer, unless otherwise noted in Table 1 (column "ms/packet"). The packetization interval determines the minimum end-to-end delay; longer packets introduce less header overhead but higher delay and make packet loss more noticeable. For non-interactive applications such as lectures or for links with severe bandwidth constraints, a higher packetization delay MAY be used. A receiver SHOULD accept packets representing between 0 and 200 ms of audio data. (For framed audio encodings, a receiver SHOULD accept packets with 200 ms divided by the frame duration, rounded up.) This restriction allows reasonable buffer sizing for the receiver.

4.3 Guidelines for Sample-Based Audio Encodings

In *sample-based* encodings, each audio sample is represented by a fixed number of bits. Within the compressed audio data, codes for individual samples may span octet boundaries. An RTP audio packet may contain any number of audio samples, subject to the constraint that the number of bits per sample times the number of samples per packet yields an integral octet count. *Fractional encodings* produce less than one octet per sample.

The duration of an audio packet is determined by the number of samples in the packet.

For sample-based encodings producing one or more octets per sample, samples from different channels sampled at the same sampling instant SHOULD be packed in consecutive octets. For example, for a two-channel encoding, the octet sequence is (left channel, first sample), (right channel, first sample), (left channel, second sample), (right channel, second sample), ... For multi-octet encodings, octets SHOULD be transmitted in network byte order (i.e., most significant octet first).

The packing of sample-based encodings producing less than one octet per sample is encoding-specific.

The RTP timestamp reflects the instant at which the first sample in the packet was sampled, that is, the oldest information in the packet.

4.4 Guidelines for Frame-Based Audio Encodings

Frame-based encodings encode a fixed-length block of audio into another block of compressed data, typically also of fixed length. For frame-based encodings, the sender MAY choose to combine several such frames into a single RTP packet. The receiver can tell the number of frames contained in an RTP packet, if all the frames have the same length, by dividing the RTP payload length by the audio frame size which is defined as part of the encoding. This does not work when carrying frames of different sizes unless the frame sizes are relatively prime. If not, the frames MUST indicate their size.

For frame-based codecs, the channel order is defined for the whole block. That is, for two-channel audio, right and left samples SHOULD be coded independently, with the encoded frame for the left channel preceding that for the right channel.

All frame-oriented audio codecs SHOULD be able to encode and decode several consecutive frames within a single packet. Since the frame size for the frame-oriented codecs is given, there is no need to use a separate designation for the same encoding, but with different number of frames per packet.

RTP packets SHALL contain a whole number of frames, with frames inserted according to age within a packet, so that the oldest frame (to be played first) occurs immediately after the RTP packet header. The

RTP timestamp reflects the instant at which the first sample in the first frame was sampled, that is, the oldest information in the packet.

4.5 Audio Encodings

name of encoding	sample/frame	bits/sample	sampling rate	ms/frame	default ms/packet
1016	frame	N/A	8,000	30	30
CN	frame	N/A	var.		
DVI4	sample	4	var.		20
G722	sample	8	16,000		20
G723	frame	N/A	8,000	30	30
G726-16	sample	2	8,000		20
G726-24	sample	3	8,000		20
G726-32	sample	4	8,000		20
G726-40	sample	5	8,000		20
G727-16	sample	2	8,000		20
G727-24	sample	3	8,000		20
G727-32	sample	4	8,000		20
G727-40	sample	5	8,000		20
G728	frame	N/A	8,000	2.5	20
G729	frame	N/A	8,000	10	20
GSM	frame	N/A	8,000	20	20
GSM-HR	frame	N/A	8,000	20	20
GSM-EFR	frame	N/A	8,000	20	20
L8	sample	8	var.	20	
L16	sample	16	var.	20	
LPC	frame	N/A	8,000	20	20
MPA	frame	N/A	var.	20	
PCMA	sample	8	var.	20	
PCMU	sample	8	var.	20	
QCELP	frame	N/A	8,000	20	
SX7300P	frame	N/A	8,000	15	30
SX8300P	frame	N/A	8,000	15	30
SX9600P	frame	N/A	8,000	15	30
VDVI	sample	var.	var.	20	

Table 1: Properties of Audio Encodings (N/A: not applicable; var.: variable)

The characteristics of the audio encodings described in this document are shown in Table 1; they are listed in order of their payload type in Table 4. While most audio codecs are only specified for a fixed sampling rate, some sample-based algorithms (indicated by an entry of "var." in the sampling rate column of Table 1) may be used with different sampling rates, resulting in different coded bit rates. When used with a sampling rate other than that for which a static payload type is defined, non-RTP means beyond the scope of this memo

MUST be used to define a dynamic payload type and MUST indicate the selected sampling rate.

4.5.1 1016

Encoding 1016 is a frame based encoding using code-excited linear prediction (CELP) and is specified in Federal Standard FED-STD 1016 [7, 8, 9, 10].

4.5.2 CN

The CN (comfort noise) packet contains a single-octet message to the receiver to play comfort noise at the absolute level specified. This message would normally be sent once at the beginning of a silence period (which also indicates the transition from speech to silence), but the rate of noise level updates is implementation specific. The magnitude of the noise level is packed into the least significant bits of the noise-level payload, as shown below.

The noise level is expressed in -dBov, with values from 0 to 127 representing 0 to -127 dBov. dBov is the level relative to the overload of the system. (Note: Representation relative to the overload point of a system is particularly useful for digital implementations, since one does not need to know the relative calibration of the analog circuitry.) For example, in a 16-bit linear PCM system (L16), a signal with 0 dBov represents a square wave with the maximum possible amplitude (+/-32767), and -63 dBov corresponds to -58 dBm0 in a standard telephone system. (dBm is the power level in decibels relative to 1 mW, with an impedance of 600 Ohms.)

```
 0 1 2 3 4 5 6 7
+-+-+-+-+-+-+-+-+
|0|   level     |
+-+-+-+-+-+-+-+-+
```

The RTP header for the comfort noise packet SHOULD be constructed as if the comfort noise were an independent codec. Thus, the RTP timestamp designates the beginning of the silence period. A static payload type is assigned for a sampling rate of 8,000 Hz; if other sampling rates are needed, they MUST be defined through dynamic payload types. The RTP packet SHOULD NOT have the marker bit set.

The CN payload type is primarily for use with L16, DVI4, PCMA, PCMU and other audio codecs that do not support comfort noise as part of the codec itself. G.723.1 and G.729 have their own comfort noise systems as part of Annexes A (G.723.1) and B (G.729), respectively.

4.5.3 DVI4

DVI4 is specified, with pseudo-code, in [11] as the IMA ADPCM wave type.

However, the encoding defined here as DVI4 differs in three respects from this recommendation:

- The RTP DVI4 header contains the predicted value rather than the first sample value contained the IMA ADPCM block header.
- IMA ADPCM blocks contain an odd number of samples, since the first sample of a block is contained just in the header (uncompressed), followed by an even number of compressed samples. DVI4 has an even number of compressed samples only, using the ‘predict’ word from the header to decode the first sample.

- For DVI4, the 4-bit samples are packed with the first sample in the four most significant bits and the second sample in the four least significant bits. In the IMA ADPCM codec, the samples are packed in the opposite order.

Each packet contains a single DVI block. This profile only defines the 4-bit-per-sample version, while IMA also specifies a 3-bit-per-sample encoding.

The "header" word for each channel has the following structure:

```
int16  predict;   /* predicted value of first sample
                     from the previous block (L16 format) */
u_int8 index;     /* current index into stepsize table */
u_int8 reserved;  /* set to zero by sender, ignored by receiver */
```

Each octet following the header contains two 4-bit samples, thus the number of samples per packet MUST be even because there is no means to indicate a partially filled last octet.

Packing of samples for multiple channels is for further study.

The document *IMA Recommended Practices for Enhancing Digital Audio Compatibility in Multimedia Systems (version 3.0)* contains the algorithm description. It is available from

Interactive Multimedia Association
48 Maryland Avenue, Suite 202
Annapolis, MD 21401-8011
USA
phone: +1 410 626-1380

4.5.4 G722

G722 is specified in ITU-T Recommendation G.722, "7 kHz audio-coding within 64 kbit/s".

4.5.5 G723

G723 is specified in ITU Recommendation G.723.1, "Dual-rate speech coder for multimedia communications transmitting at 5.3 and 6.3 kbit/s". The G.723.1 5.3/6.3 kbit/s codec was defined by the ITU-T as a mandatory codec for ITU-T H.324 GSTN videophone terminal applications. The algorithm has a floating point specification in Annex B to G.723.1, a silence compression algorithm in Annex A to G.723.1 and an encoded signal bit-error sensitivity specification in G.723.1 Annex C.

This Recommendation specifies a coded representation that can be used for compressing the speech signal component of multi-media services at a very low bit rate. Audio is encoded in 30 ms frames, with an additional delay of 7.5 ms due to look-ahead. A G.723.1 frame can be one of three sizes: 24 octets (6.3 kb/s frame), 20 octets (5.3 kb/s frame), or 4 octets. These 4-octet frames are called SID frames (Silence Insertion Descriptor) and are used to specify comfort noise parameters. There is no restriction on how 4, 20, and 24 octet frames are intermixed. The least significant two bits of the first octet in the frame determine the frame size and codec type:

bits	content	octets/frame
00	high-rate speech (6.3 kb/s)	24
01	low-rate speech (5.3 kb/s)	20
10	SID frame	4
11	reserved	

It is possible to switch between the two rates at any 30 ms frame boundary. Both (5.3 kb/s and 6.3 kb/s) rates are a mandatory part of the encoder and decoder. This coder was optimized to represent speech with near-toll quality at the above rates using a limited amount of complexity.

The packing of the encoded bit stream into octets and the transmission order of the octets is specified in G.723.1.

4.5.6 G726-16, G726-24, G726-32, G726-40

ITU-T Recommendation G.726 describes, among others, the algorithm recommended for conversion of a single 64 kbit/s A-law or mu-law PCM channel encoded at 8000 samples/sec to and from a 32 kbit/s channel. The conversion is applied to the PCM stream using an Adaptive Differential Pulse Code Modulation (ADPCM) transcoding technique. G.726 describes codecs operating at 16 kb/s (2 bits/sample), 24 kb/s (3 bits/sample), 32 kb/s (4 bits/sample), 40 kb/s (5 bits/sample). These encodings are labeled G726-16, G726-24, G726-32 and G726-40, respectively.

Note: In 1990, ITU-T Recommendation G.721 was merged with Recommendation G.723 into ITU-T Recommendation G.726. Thus, G726-32 designates the same algorithm as G721 in RFC 1890.

No payload-specific header information SHALL be included as part of the audio data. The 4-bit code words of the G726-32 encoding MUST be packed into octets as follows: the first code word is placed in the four least significant bits of the first octet, with the least significant bit of the code word in the least significant bit of the octet; the second code word is placed in the four most significant bits of the first octet, with the most significant bit of the code word in the most significant bit of the octet. Subsequent pairs of the code words SHALL be packed in the same way into successive octets, with the first code word of each pair placed in the least significant four bits of the octet. The number of samples per packet MUST be even because there is no means to indicate a partially filled last octet.

4.5.7 G727-16, G727-24, G727-32, G727-40

ITU-T Recommendation G.727, "5-, 4-, 3- and 2-bits sample embedded adaptive differential pulse code modulation (ADPCM)", specifies an embedded ADPCM algorithm which has the intrinsic capability of dropping bits in the encoded words to alleviate network congestion conditions. The algorithm, although not bitstream compatible with G.726, was based and has a structure similar to the G.726 ADPCM algorithm.

4.5.8 G728

G728 is specified in ITU-T Recommendation G.728, "Coding of speech at 16 kbit/s using low-delay code excited linear prediction".

A G.278 encoder translates 5 consecutive audio samples into a 10-bit codebook index, resulting in a bit rate of 16 kb/s for audio sampled at 8,000 samples per second. The group of five consecutive samples is called a vector. Four consecutive vectors, labeled V1 to V4 (where V1 is to be played first by the receiver), build one G.728 frame. The four vectors of 40 bits are packed into 5 octets, labeled B1 through B5. B1 SHALL be placed first in the RTP packet.

Referring to the figure below, the principle for bit order is “maintenance of bit significance”. Bits from an older vector are more significant than bits from newer vectors. The MSB of the frame goes to the MSB of B1 and the LSB of the frame goes to LSB of B5.

```
          1         2         3        3
0         0         0         0        9
++++++++++++++++++++++++++++++++++++++++
<---V1---><---V2---><---V3---><---V4---> vectors
<--B1--><--B2--><--B3--><--B4--><--B5--> octets
<------------- frame 1 ---------------->
```

In particular, B1 contains the eight most significant bits of V1, with the MSB of V1 being the MSB of B1. B2 contains the two least significant bits of V1, the more significant of the two in its MSB, and the six most significant bits of V2. B1 SHALL be placed first in the RTP packet and B5 last.

4.5.9 G729

G729 is specified in ITU-T Recommendation G.729, “Coding of speech at 8 kbit/s using conjugate structure-algebraic code excited linear prediction (CS-ACELP)”. A reduced-complexity version of the G.729 algorithm is specified in Annex A to Rec. G.729. The speech coding algorithms in the main body of G.729 and in G.729 Annex A are fully interoperable with each other, so there is no need to further distinguish between them. The G.729 and G.729 Annex A codecs were optimized to represent speech with high quality, where G.729 Annex A trades some speech quality for an approximate 50% complexity reduction [12].

A voice activity detector (VAD) and comfort noise generator (CNG) algorithm in Annex B of G.729 is recommended for digital simultaneous voice and data applications and can be used in conjunction with G.729 or G.729 Annex A. A G.729 or G.729 Annex A frame contains 10 octets, while the G.729 Annex B comfort noise frame occupies 2 octets:

```
 0                   1
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|L|  LSF1   |  LSF2 |   GAIN  |R|
|S|         |       |         |E|
|F|0 1 2 3 4|0 1 2 3|0 1 2 3 4|S|
|0|         |       |         |V|    RESV = Reserved (zero)
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

An RTP packet may consist of zero or more G.729 or G.729 Annex A frames, followed by zero or one G.729 Annex B payloads. The presence of a comfort noise frame can be deduced from the length of the RTP payload.

The transmitted parameters of a G.729/G.729A 10-ms frame, consisting of 80 bits, are defined in Recommendation G.729, Table 8/G.729.

The mapping of the these parameters is given below. Bits are numbered as Internet order, that is, the most significant bit is bit 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
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|L|      L1     |    L2   |    L3   |       P1      |P|    C1   |
|0|             |         |         |               |0|         |
| |0 1 2 3 4 5 6|0 1 2 3 4|0 1 2 3 4|0 1 2 3 4 5 6 7| |0 1 2 3 4|
| |             |         |         |               | |         |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+

               4                   5                   6
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 2 3
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|       C1      |  S1   | GA1 |  GB1  |    P2   |       C2      |
|               |       |     |       |         |               |
|5 6 7 8 9 1 1 1|0 1 2 3|0 1 2|0 1 2 3|0 1 2 3 4|0 1 2 3 4 5 6 7|
|          0 1 2|       |     |       |         |               |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+

           7
4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|   C2    |  S2   | GA2 |  GB2  |
|         |       |     |       |
|8 9 1 1 1|0 1 2 3|0 1 2|0 1 2 3|
|    0 1 2|       |     |       |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

4.5.10 GSM

GSM (group speciale mobile) denotes the European GSM 06.10 standard for full-rate speech transcoding, ETS 300 961, which is based on RPE/LTP (residual pulse excitation/long term prediction) coding at a rate of 13 kb/s [13, 14, 15]. The text of the standard can be obtained from

ETSI (European Telecommunications Standards Institute)
ETSI Secretariat: B.P.152
F-06561 Valbonne Cedex
France
Phone: +33 92 94 42 00
Fax: +33 93 65 47 16

Blocks of 160 audio samples are compressed into 33 octets, for an effective data rate of 13,200 b/s.

General Packaging Issues The GSM standard specifies the bit stream produced by the codec, but does not specify how these bits should be packed for transmission. Some software implementations of the GSM codec use a different packing than that specified here.

In the GSM packing used by RTP, the bits SHALL be packed beginning from the most significant bit. Every 160 sample GSM frame is coded into one 33 octet (264 bit) buffer. Every such buffer begins with a 4 bit signature (0xD), followed by the MSB encoding of the fields of the frame. The first octet thus contains 1101 in the 4 most significant bits (0-3) and the 4 most significant bits of F1 (0-3) in the 4 least significant bits (4-7). The second octet contains the 2 least significant bits of F1 in bits 0-1, and F2 in bits 2-7, and so on. The order of the fields in the frame is described in Table 2.

GSM variable names and numbers In the RTP encoding we have the bit pattern described in Table 3, where F.i signifies the ith bit of the field F, bit 0 is the most significant bit, and the bits of every octet are numbered from 0 to 7 from most to least significant.

4.5.11 GSM-HR

GSM-HR denotes GSM 06.20 half rate speech transcoding, specified in ETS 300 969 which is available from ETSI at the address given in Section 4.5.10. This codec has a frame length of 112 bits (14 octets). Packing of the fields in the codec bit stream into octets for transmission in RTP is done in a manner similar to that specified here for the original GSM 06.10 codec and is specified in ETSI Technical Specification TS 101 318.

4.5.12 GSM-EFR

GSM-EFR denotes GSM 06.60 enhanced full rate speech transcoding, specified in ETS 300 969 which is available from ETSI at the address given in Section 4.5.10. This codec has a frame length of 244 bits. For transmission in RTP, each codec frame is packed into a 31 octet (248 bit) buffer beginning with a 4-bit signature 0xC in a manner similar to that specified here for the original GSM 06.10 codec. The packing is specified in ETSI Technical Specification TS 101 318.

4.5.13 L8

L8 denotes linear audio data samples, using 8-bits of precision with an offset of 128, that is, the most negative signal is encoded as zero.

4.5.14 L16

L16 denotes uncompressed audio data samples, using 16-bit signed representation with 65535 equally divided steps between minimum and maximum signal level, ranging from -32768 to 32767. The value is represented in two's complement notation and transmitted in network byte order (most significant byte first).

4.5.15 LPC

LPC designates an experimental linear predictive encoding contributed by Ron Frederick, Xerox PARC, which is based on an implementation written by Ron Zuckerman, Motorola, posted to the Usenet group comp.dsp on June 26, 1992. The codec generates 14 octets for every frame. The framesize is set to 20 ms, resulting in a bit rate of 5,600 b/s.

field	field name	bits	field	field name	bits
1	LARc[0]	6	39	xmc[22]	3
2	LARc[1]	6	40	xmc[23]	3
3	LARc[2]	5	41	xmc[24]	3
4	LARc[3]	5	42	xmc[25]	3
5	LARc[4]	4	43	Nc[2]	7
6	LARc[5]	4	44	bc[2]	2
7	LARc[6]	3	45	Mc[2]	2
8	LARc[7]	3	46	xmaxc[2]	6
9	Nc[0]	7	47	xmc[26]	3
10	bc[0]	2	48	xmc[27]	3
11	Mc[0]	2	49	xmc[28]	3
12	xmaxc[0]	6	50	xmc[29]	3
13	xmc[0]	3	51	xmc[30]	3
14	xmc[1]	3	52	xmc[31]	3
15	xmc[2]	3	53	xmc[32]	3
16	xmc[3]	3	54	xmc[33]	3
17	xmc[4]	3	55	xmc[34]	3
18	xmc[5]	3	56	xmc[35]	3
19	xmc[6]	3	57	xmc[36]	3
20	xmc[7]	3	58	xmc[37]	3
21	xmc[8]	3	59	xmc[38]	3
22	xmc[9]	3	60	Nc[3]	7
23	xmc[10]	3	61	bc[3]	2
24	xmc[11]	3	62	Mc[3]	2
25	xmc[12]	3	63	xmaxc[3]	6
26	Nc[1]	7	64	xmc[39]	3
27	bc[1]	2	65	xmc[40]	3
28	Mc[1]	2	66	xmc[41]	3
29	xmaxc[1]	6	67	xmc[42]	3
30	xmc[13]	3	68	xmc[43]	3
31	xmc[14]	3	69	xmc[44]	3
32	xmc[15]	3	70	xmc[45]	3
33	xmc[16]	3	71	xmc[46]	3
34	xmc[17]	3	72	xmc[47]	3
35	xmc[18]	3	73	xmc[48]	3
36	xmc[19]	3	74	xmc[49]	3
37	xmc[20]	3	75	xmc[50]	3
38	xmc[21]	3	76	xmc[51]	3

Table 2: Ordering of GSM variables

Octet	Bit 0	Bit 1	Bit 2	Bit 3	Bit 4	Bit 5	Bit 6	Bit 7
0	1	1	0	1	LARc0.0	LARc0.1	LARc0.2	LARc0.3
1	LARc0.4	LARc0.5	LARc1.0	LARc1.1	LARc1.2	LARc1.3	LARc1.4	LARc1.5
2	LARc2.0	LARc2.1	LARc2.2	LARc2.3	LARc2.4	LARc3.0	LARc3.1	LARc3.2
3	LARc3.3	LARc3.4	LARc4.0	LARc4.1	LARc4.2	LARc4.3	LARc5.0	LARc5.1
4	LARc5.2	LARc5.3	LARc6.0	LARc6.1	LARc6.2	LARc7.0	LARc7.1	LARc7.2
5	Nc0.0	Nc0.1	Nc0.2	Nc0.3	Nc0.4	Nc0.5	Nc0.6	bc0.0
6	bc0.1	Mc0.0	Mc0.1	xmaxc00	xmaxc01	xmaxc02	xmaxc03	xmaxc04
7	xmaxc05	xmc0.0	xmc0.1	xmc0.2	xmc1.0	xmc1.1	xmc1.2	xmc2.0
8	xmc2.1	xmc2.2	xmc3.0	xmc3.1	xmc3.2	xmc4.0	xmc4.1	xmc4.2
9	xmc5.0	xmc5.1	xmc5.2	xmc6.0	xmc6.1	xmc6.2	xmc7.0	xmc7.1
10	xmc7.2	xmc8.0	xmc8.1	xmc8.2	xmc9.0	xmc9.1	xmc9.2	xmc10.0
11	xmc10.1	xmc10.2	xmc11.0	xmc11.1	xmc11.2	xmc12.0	xmc12.1	xcm12.2
12	Nc1.0	Nc1.1	Nc1.2	Nc1.3	Nc1.4	Nc1.5	Nc1.6	bc1.0
13	bc1.1	Mc1.0	Mc1.1	xmaxc10	xmaxc11	xmaxc12	xmaxc13	xmaxc14
14	xmax15	xmc13.0	xmc13.1	xmc13.2	xmc14.0	xmc14.1	xmc14.2	xmc15.0
15	xmc15.1	xmc15.2	xmc16.0	xmc16.1	xmc16.2	xmc17.0	xmc17.1	xmc17.2
16	xmc18.0	xmc18.1	xmc18.2	xmc19.0	xmc19.1	xmc19.2	xmc20.0	xmc20.1
17	xmc20.2	xmc21.0	xmc21.1	xmc21.2	xmc22.0	xmc22.1	xmc22.2	xmc23.0
18	xmc23.1	xmc23.2	xmc24.0	xmc24.1	xmc24.2	xmc25.0	xmc25.1	xmc25.2
19	Nc2.0	Nc2.1	Nc2.2	Nc2.3	Nc2.4	Nc2.5	Nc2.6	bc2.0
20	bc2.1	Mc2.0	Mc2.1	xmaxc20	xmaxc21	xmaxc22	xmaxc23	xmaxc24
21	xmaxc25	xmc26.0	xmc26.1	xmc26.2	xmc27.0	xmc27.1	xmc27.2	xmc28.0
22	xmc28.1	xmc28.2	xmc29.0	xmc29.1	xmc29.2	xmc30.0	xmc30.1	xmc30.2
23	xmc31.0	xmc31.1	xmc31.2	xmc32.0	xmc32.1	xmc32.2	xmc33.0	xmc33.1
24	xmc33.2	xmc34.0	xmc34.1	xmc34.2	xmc35.0	xmc35.1	xmc35.2	xmc36.0
25	Xmc36.1	xmc36.2	xmc37.0	xmc37.1	xmc37.2	xmc38.0	xmc38.1	xmc38.2
26	Nc3.0	Nc3.1	Nc3.2	Nc3.3	Nc3.4	Nc3.5	Nc3.6	bc3.0
27	bc3.1	Mc3.0	Mc3.1	xmaxc30	xmaxc31	xmaxc32	xmaxc33	xmaxc34
28	xmaxc35	xmc39.0	xmc39.1	xmc39.2	xmc40.0	xmc40.1	xmc40.2	xmc41.0
29	xmc41.1	xmc41.2	xmc42.0	xmc42.1	xmc42.2	xmc43.0	xmc43.1	xmc43.2
30	xmc44.0	xmc44.1	xmc44.2	xmc45.0	xmc45.1	xmc45.2	xmc46.0	xmc46.1
31	xmc46.2	xmc47.0	xmc47.1	xmc47.2	xmc48.0	xmc48.1	xmc48.2	xmc49.0
32	xmc49.1	xmc49.2	xmc50.0	xmc50.1	xmc50.2	xmc51.0	xmc51.1	xmc51.2

Table 3: GSM payload format

4.5.16 MPA

MPA denotes MPEG-1 or MPEG-2 audio encapsulated as elementary streams. The encoding is defined in ISO standards ISO/IEC 11172-3 and 13818-3. The encapsulation is specified in RFC 2250 [16].

Sampling rate and channel count are contained in the payload. MPEG-1 audio supports sampling rates of 32, 44.1, and 48 kHz (ISO/IEC 11172-3, section 1.1; "Scope"). MPEG-2 additionally supports sampling rates of 16, 22.05 and 24 kHz.

4.5.17 PCMA and PCMU

PCMA and PCMU are specified in ITU-T Recommendation G.711. Audio data is encoded as eight bits per sample, after logarithmic scaling. PCMU denotes mu-law scaling, PCMA A-law scaling. A detailed description is given by Jayant and Noll [17]. Each G.711 octet SHALL be octet-aligned in an RTP packet. The sign bit of each G.711 octet SHALL correspond to the most significant bit of the octet in the RTP packet (i.e., assuming the G.711 samples are handled as octets on the host machine, the sign bit SHALL be the most signficant bit of the octet as defined by the host machine format). The 56 kb/s and 48 kb/s modes of G.711 are not applicable to RTP, since PCMA and PCMU SHALL always be transmitted as 8-bit samples.

4.5.18 QCELP

The Electronic Industries Association (EIA) & Telecommunications Industry Association (TIA) standard IS-733, "TR45: High Rate Speech Service Option for Wideband Spread Spectrum Communications Systems," defines the QCELP audio compression algorithm for use in wireless CDMA applications. The QCELP CODEC compresses each 20 milliseconds of 8000 Hz, 16- bit sampled input speech into one of four different size output frames: Rate 1 (266 bits), Rate 1/2 (124 bits), Rate 1/4 (54 bits) or Rate 1/8 (20 bits). For typical speech patterns, this results in an average output of 6.8 k bits/sec for normal mode and 4.7 k bits/sec for reduced rate mode. The packetization of the QCELP audio codec is described in [18].

4.5.19 RED

The redundant audio payload format "RED" is specified by RFC 2198 [19]. It defines a means by which multiple redundant copies of an audio packet may be transmitted in a single RTP stream. Each packet in such a stream contains, in addition to the audio data for that packetization interval, a (more heavily compressed) copy of the data from a previous packetization interval. This allows an approximation of the data from lost packets to be recovered upon decoding of a subsequent packet, giving much improved sound quality when compared with silence substitution for lost packets.

4.5.20 SX*

The SX7300P, SX8300P and SX9600P codecs are part of the same compatible family and distinguished by the first octet in each frame, where "x" can be any value:

```
  0 1 2 3 4 5 6 7
 +-+-+-+-+-+-+-+-+
 |0 0 x          |  SX7300P bitstream (14 byte frame)
 |0 1 0          |  SX8300P bitstream (16 byte frame)
 |1 0 x          |  VAD bistream      ( 2 byte frame)
```

```
 |1 1 x           |  SX9600P bitstream (18 byte frame)
+-+-+-+-+-+-+-+-+
```

SX7300P The SX7300P is a low-complexity CELP-based audio codec operating at a sampling rate of 8000 Hz. It encodes blocks of 120 audio samples (15 ms) into an encoded frame of 14 octets, yielding an encoded bit rate of approximately 7467 b/s.

SX8300P The SX8300P is a low-complexity CELP-based audio codec operating at a sampling rate of 8000 Hz. It encodes blocks of 120 audio samples (15 ms) into an encoded frame of 16 octets, yielding an encoded bit rate of approximately 8533 b/s.

SX9600P The SX9600P is a low-complexity, toll-quality CELP-based audio codec operating at a sampling rate of 8000 Hz. It encodes blocks of 120 audio samples (15 ms) into an encoded frame of 18 octets, yielding an encoded bit rate of 9600 b/s.

4.5.21 VDVI

VDVI is a variable-rate version of DVI4, yielding speech bit rates of between 10 and 25 kb/s. It is specified for single-channel operation only. Samples are packed into octets starting at the most-significant bit.

It uses the following encoding:

DVI4 codeword	VDVI bit pattern
0	00
1	010
2	1100
3	11100
4	111100
5	1111100
6	11111100
7	11111110
8	10
9	011
10	1101
11	11101
12	111101
13	1111101
14	11111101
15	11111111

5 Video

The following sections describe the video encodings that are defined in this memo and give their abbreviated names used for identification. These video encodings and their payload types are listed in Table 5.

All of these video encodings use an RTP timestamp frequency of 90,000 Hz, the same as the MPEG presentation time stamp frequency. This frequency yields exact integer timestamp increments for the typical

24 (HDTV), 25 (PAL), and 29.97 (NTSC) and 30 Hz (HDTV) frame rates and 50, 59.94 and 60 Hz field rates. While 90 kHz is the RECOMMENDED rate for future video encodings used within this profile, other rates MAY be used. However, it is not sufficient to use the video frame rate (typically between 15 and 30 Hz) because that does not provide adequate resolution for typical synchronization requirements when calculating the RTP timestamp corresponding to the NTP timestamp in an RTCP SR packet. The timestamp resolution MUST also be sufficient for the jitter estimate contained in the receiver reports.

For most of these video encodings, the RTP timestamp encodes the sampling instant of the video image contained in the RTP data packet. If a video image occupies more than one packet, the timestamp is the same on all of those packets. Packets from different video images are distinguished by their different timestamps.

Most of these video encodings also specify that the marker bit of the RTP header SHOULD be set to one in the last packet of a video frame and otherwise set to zero. Thus, it is not necessary to wait for a following packet with a different timestamp to detect that a new frame should be displayed.

5.1 BT656

The encoding is specified in ITU-R Recommendation BT.656-3, “Interfaces for Digital Component Video Signals in 525-Line and 625-Line Television Systems operating at the 4:2:2 Level of Recommendation ITU-R BT.601 (Part A)”. The packetization and RTP-specific properties are described in RFC 2431 [20].

5.2 CelB

The CELL-B encoding is a proprietary encoding proposed by Sun Microsystems. The byte stream format is described in RFC 2029 [21].

5.3 JPEG

The encoding is specified in ISO Standards 10918-1 and 10918-2. The RTP payload format is as specified in RFC 2435 [22].

5.4 H261

The encoding is specified in ITU-T Recommendation H.261, “Video codec for audiovisual services at p x 64 kbit/s”. The packetization and RTP-specific properties are described in RFC 2032 [23].

5.5 H263

The encoding is specified in the 1996 version of ITU-T Recommendation H.263, “Video coding for low bit rate communication”. The packetization and RTP-specific properties are described in RFC 2190 [24].

5.6 H263-1998

The encoding is specified in the 1998 version of ITU-T Recommendation H.263, “Video coding for low bit rate communication”. The packetization and RTP-specific properties are described in RFC 2429 [25]. Because the 1998 version of H.263 is a superset of the 1996 syntax, this payload format can also be used with the 1996 version of H.263, and is recommended for this use by new implementations. This payload 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 MUST use the payload format described in RFC 2429.

5.7 MPV

MPV designates the use of MPEG-1 and MPEG-2 video encoding elementary streams as specified in ISO Standards ISO/IEC 11172 and 13818-2, respectively. The RTP payload format is as specified in RFC 2250 [16], Section 3.

5.8 MP2T

MP2T designates the use of MPEG-2 transport streams, for either audio or video. The RTP payoad format is described in RFC 2250 [16], Section 2.

5.9 MP1S

MP1S designates an MPEG-1 systems stream, encapsulated according to RFC 2250 [16].

5.10 MP2P

MP2P designates an MPEG-2 program stream, encapsulated according to RFC 2250 [16].

5.11 BMPEG

BMPEG designates an experimental payload format for MPEG-1 and MPEG-2 which specifies bundled (multiplexed) transport of audio and video elementary streams in one RTP stream as an alternative to the MP1S and MP2P formats. The packetization is described in RFC 2343 [26].

5.12 nv

The encoding is implemented in the program 'nv', version 4, developed at Xerox PARC by Ron Frederick. Further information is available from the author:

Ron Frederick
Xerox Palo Alto Research Center
3333 Coyote Hill Road
Palo Alto, CA 94304
United States
electronic mail: `frederic@parc.xerox.com`

6 Payload Type Definitions

Tables 4 and 5 define this profile's static payload type values for the PT field of the RTP data header. In addition, payload type values in the range 96-127 MAY be defined dynamically through a conference control protocol, which is beyond the scope of this document. For example, a session directory could specify that for a given session, payload type 96 indicates PCMU encoding, 8,000 Hz sampling rate, 2 channels. Entries in Tables 4 and 5 with payload type "dyn" have no static payload type assigned and are only used with a dynamic payload type. The payload type range marked 'reserved' has been set aside so that RTCP and RTP packets can be reliably distinguished (see Section "Summary of Protocol Constants" of the RTP protocol specification).

The payload types currently defined in this profile are assigned to exactly one of three categories or *media types*: audio only, video only and those combining audio and video. The media types are marked in Tables 4 and 5 as "A", "V" and "AV", respectively. Payload types of different media types SHALL NOT be interleaved or multiplexed within a single RTP session, but multiple RTP sessions MAY be used in parallel to send multiple media types. An RTP source MAY change payload types within the same media type during a session.

Session participants agree through mechanisms beyond the scope of this specification on the set of payload types allowed in a given session. This set MAY, for example, be defined by the capabilities of the applications used, negotiated by a conference control protocol or established by agreement between the human participants.

Audio applications operating under this profile SHOULD, at a minimum, be able to send and/or receive payload types 0 (PCMU) and 5 (DVI4). This allows interoperability without format negotiation and ensures successful negotation with a conference control protocol.

7 RTP over TCP and Similar Byte Stream Protocols

Under special circumstances, it may be necessary to carry RTP in protocols offering a byte stream abstraction, such as TCP, possibly multiplexed with other data. If the application does not define its own method of delineating RTP and RTCP packets, it SHOULD prefix each packet with a two-octet length field.

(Note: RTSP [27] provides its own encapsulation and does not need an extra length indication.)

8 Port Assignment

As specified in the RTP protocol definition, RTP data SHOULD be carried on an even UDP or TCP port number and the corresponding RTCP packets SHOULD be carried on the next higher (odd) port number.

Applications operating under this profile MAY use any such UDP or TCP port pair. For example, the port pair MAY be allocated randomly by a session management program. A single fixed port number pair cannot be required because multiple applications using this profile are likely to run on the same host, and there are some operating systems that do not allow multiple processes to use the same UDP port with different multicast addresses.

However, port numbers 5004 and 5005 have been registered for use with this profile for those applications that choose to use them as the default pair. Applications that operate under multiple profiles MAY use this port pair as an indication to select this profile if they are not subject to the constraint of the previous paragraph. Applications need not have a default and MAY require that the port pair be explicitly specified. The particular port numbers were chosen to lie in the range above 5000 to accommodate port number allocation practice within some versions of the Unix operating system, where port numbers below 1024 can only be used by privileged processes and port numbers between 1024 and 5000 are automatically assigned by the operating system.

PT	encoding name	media type	clock rate (Hz)	channels
0	PCMU	A	8000	1
1	1016	A	8000	1
2	G726-32	A	8000	1
3	GSM	A	8000	1
4	G723	A	8000	1
5	DVI4	A	8000	1
6	DVI4	A	16000	1
7	LPC	A	8000	1
8	PCMA	A	8000	1
9	G722	A	16000	1
10	L16	A	44100	2
11	L16	A	44100	1
12	QCELP	A	8000	1
13	unassigned	A		
14	MPA	A	90000	(see text)
15	G728	A	8000	1
16	DVI4	A	11025	1
17	DVI4	A	22050	1
18	G729	A	8000	1
19	CN	A	8000	1
20	unassigned	A		
21	unassigned	A		
22	unassigned	A		
23	unassigned	A		
dyn	GSM-HR	A	8000	1
dyn	GSM-EFR	A	8000	1
dyn	RED	A		

Table 4: Payload types (PT) for audio encodings

PT	encoding name	media type	clock rate (Hz)
24	unassigned	V	
25	CelB	V	90000
26	JPEG	V	90000
27	unassigned	V	
28	nv	V	90000
29	unassigned	V	
30	unassigned	V	
31	H261	V	90000
32	MPV	V	90000
33	MP2T	AV	90000
34	H263	V	90000
35-71	unassigned	?	
72-76	reserved	N/A	N/A
77-95	unassigned	?	
96-127	dynamic	?	
dyn	BT656	V	90000
dyn	H263-1998	V	90000
dyn	MP1S	V	90000
dyn	MP2P	V	90000
dyn	BMPEG	V	90000

Table 5: Payload types (PT) for video and combined encodings

9 Changes from RFC 1890

This RFC revises RFC 1890. It is fully backwards-compatible with RFC 1890 and codifies existing practice. The changes are listed below.

- Additional payload formats and/or expanded descriptions were included for CN, G723, G726, G727, G728, G729, GSM, GSM-HR, GSM-EFR, QCELP, RED, BT656, H263-1998, MP1S, MP2P and BM-PEG.
- Static payload types 4, 12, 16, 17, 18, 19 and 34 were added.
- The policy is established that no additional registration of static payload types for this Profile will be made beyond those included in Tables 4 and 5, but additional encoding names may be registered as MIME subtypes.
- In Section 4.1, the requirement level for setting of the marker bit on the first packet after silence for audio was changed from "is" to "SHOULD be".
- Similarly, text was added to specify that the marker bit SHOULD be set to one on the last packet of a video frame, and that video frames are distinguished by their timestamps.
- This profile follows the suggestion in the RTP spec that RTCP bandwidth may be specified separately from the session bandwidth and separately for active senders and passive receivers.
- RFC references are added for payload formats published after RFC 1890.
- A minimal TCP encapsulation is defined.
- The security considerations and full copyright sections were added.
- According to Peter Hoddie of Apple, only pre-1994 Macintosh used the 22254.54 rate and none the 11127.27 rate, so the latter was dropped from the discussion of suggested sampling frequencies.
- Small clarifications of the text have been made in several places, some in response to questions from readers. In particular:
  - A definition for "media type" is given in Section 1.1 to allow the explanation of multiplexing RTP sessions in Section 6 to be more clear regarding the multiplexing of multiple media.
  - The explanation of how to determine the number of audio frames in a packet from the length was expanded.
  - More description of the allocation of bandwidth to SDES items is given.
  - The terms MUST, SHOULD, MAY, etc. are used as defined in RFC 2119.

10 Security Considerations

Implementations using the profile defined in this specification are subject to the security considerations discussed in the RTP specification [1]. This profile does not specify any different security services other than giving rules for mapping characters in a user-provided pass phrase to canonical form. The primary function of this profile is to list a set of data compression encodings for audio and video media.

Confidentiality of the media streams is achieved by encryption. Because the data compression used with the payload formats described in this profile 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, the encodings described in this profile do 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 [28] and in multicast routing protocols to allow a receiver to select which sources are allowed to reach it.

11 Full Copyright Statement

12 Acknowledgements

The comments and careful review of Steve Casner, Simao Campos and Richard Cox are gratefully acknowledged. The GSM description was adopted from the *IMTC Voice over IP Forum Service Interoperability Implementation Agreement* (January 1997). Fred Burg and Terry Lyons helped with the G.729 description.

13 Address of Author

Henning Schulzrinne
Dept. of Computer Science
Columbia University
1214 Amsterdam Avenue
New York, NY 10027
USA
electronic mail: `schulzrinne@cs.columbia.edu`

References

[1] H. Schulzrinne, S. Casner, R. Frederick, and V. Jacobson, "RTP: A transport protocol for real-time applications," Internet Draft, Internet Engineering Task Force, Feb. 1999 Work in progress, revision to RFC 1889.

[2] S. Bradner, "Key words for use in RFCs to Indicate Requirement Levels," RFC 2119, Internet Engineering Task Force, Mar. 1997.

[3] N. Freed, J. Klensin, and J. Postel, "Multipurpose Internet Mail Extensions (MIME) Part Four: Registration Procedures," RFC 2048, Internet Engineering Task Force, Nov. 1996.

[4] P. Hoschka, "MIME Type Registration of RTP Payload Types," Internet Draft, Internet Engineering Task Force, Feb. 1999 Work in progress.

[5] M. Handley and V. Jacobson, "SDP: Session Description Protocol," Request for Comments (Proposed Standard) RFC 2327, Internet Engineering Task Force, Apr. 1998.

[6] Apple Computer, "Audio interchange file format AIFF-C," Aug. 1991. (also ftp://ftp.sgi.com/sgi/aiff-c.9.26.91.ps.Z).

[7] Office of Technology and Standards, "Telecommunications: Analog to digital conversion of radio voice by 4,800 bit/second code excited linear prediction (celp)," Federal Standard FS-1016, GSA, Room 6654; 7th & D Street SW; Washington, DC 20407 (+1-202-708-9205), 1990.

[8] J. P. Campbell, Jr., T. E. Tremain, and V. C. Welch, "The proposed Federal Standard 1016 4800 bps voice coder: CELP," *Speech Technology*, vol. 5, pp. 58–64, April/May 1990.

[9] J. P. Campbell, Jr., T. E. Tremain, and V. C. Welch, "The federal standard 1016 4800 bps CELP voice coder," *Digital Signal Processing*, vol. 1, no. 3, pp. 145–155, 1991.

[10] J. P. Campbell, Jr., T. E. Tremain, and V. C. Welch, "The DoD 4.8 kbps standard (proposed federal standard 1016)," in *Advances in Speech Coding* (B. Atal, V. Cuperman, and A. Gersho, eds.), ch. 12, pp. 121–133, Kluwer Academic Publishers, 1991.

[11] IMA Digital Audio Focus and Technical Working Groups, "Recommended practices for enhancing digital audio compatibility in multimedia systems (version 3.00)," tech. rep., Interactive Multimedia Association, Annapolis, Maryland, Oct. 1992.

[12] D. Deleam and J.-P. Petit, "Real-time implementations of the recent ITU-T low bit rate speech coders on the TI TMS320C54X DSP: results, methodology, and applications," in *Proc. of International Conference on Signal Processing, Technology, and Applications (ICSPAT)*, (Boston, Massachusetts), pp. 1656–1660, Oct. 1996.

[13] M. Mouly and M.-B. Pautet, *The GSM system for mobile communications*. Lassay-les-Chateaux, France: Europe Media Duplication, 1993.

[14] J. Degener, "Digital speech compression," *Dr. Dobb's Journal*, Dec. 1994.

[15] S. M. Redl, M. K. Weber, and M. W. Oliphant, *An Introduction to GSM*. Boston: Artech House, 1995.

[16] D. Hoffman, G. Fernando, V. Goyal, and M. Civanlar, "RTP payload format for MPEG1/MPEG2 video," Request for Comments (Proposed Standard) RFC 2250, Internet Engineering Task Force, Jan. 1998.

[17] N. S. Jayant and P. Noll, *Digital Coding of Waveforms—Principles and Applications to Speech and Video*. Englewood Cliffs, New Jersey: Prentice-Hall, 1984.

[18] K. McKay, "RTP Payload Format for PureVoice(tm) Audio", Internet Draft, Internet Engineering Task Force, Oct. 1998. Work in progress.

[19] C. Perkins, I. Kouvelas, O. Hodson, V. Hardman, M. Handley, J.C. Bolot, A. Vega-Garcia, and S. Fosse-Parisis, "RTP Payload for Redundant Audio Data," Request for Comments (Proposed Standard) RFC 2198, Internet Engineering Task Force, Sep. 1997.

[20] D. Tynan, "RTP payload format for BT.656 Video Encoding," Request for Comments (Proposed Standard) RFC 2431, Internet Engineering Task Force, Oct. 1998.

[21] M. Speer and D. Hoffman, "RTP payload format of sun's CellB video encoding," Request for Comments (Proposed Standard) RFC 2029, Internet Engineering Task Force, Oct. 1996.

[22] L. Berc, W. Fenner, R. Frederick, and S. McCanne, "RTP payload format for JPEG-compressed video," Request for Comments (Proposed Standard) RFC 2435, Internet Engineering Task Force, Oct. 1996.

[23] T. Turletti and C. Huitema, "RTP payload format for H.261 video streams," Request for Comments (Proposed Standard) RFC 2032, Internet Engineering Task Force, Oct. 1996.

[24] C. Zhu, "RTP payload format for H.263 video streams," Request for Comments (Proposed Standard) RFC 2190, Internet Engineering Task Force, Sep. 1997.

[25] C. Bormann, L. Cline, G. Deisher, T. Gardos, C. Maciocco, D. Newell, J. Ott, G. Sullivan, S. Wenger, C. Zhu, "RTP Payload Format for the 1998 Version of ITU-T Rec. H.263 Video (H.263+)," Request for Comments (Proposed Standard) RFC 2429, Internet Engineering Task Force, Oct. 1998.

[26] M. Civanlar, G. Cash, B. Haskell, "RTP Payload Format for Bundled MPEG," Request for Comments (Experimental) RFC 2343, Internet Engineering Task Force, May 1998.

[27] H. Schulzrinne, A. Rao, and R. Lanphier, "Real time streaming protocol (RTSP)," Request for Comments (Proposed Standard) RFC 2326, Internet Engineering Task Force, Apr. 1998.

[28] S. Deering, "Host Extensions for IP Multicasting," Request for Comments RFC 1112, STD 5, Internet Engineering Task Force, Aug. 1989.

Current Locations of Related Resources

Note: Several sections below refer to the *ITU-T Software Tool Library* (STL). It is available from the ITU Sales Service, Place des Nations, CH-1211 Geneve 20, Switzerland (also check http://www.itu.int. The ITU-T STL is covered by a license defined in ITU-T Recommendation G.191, *"Software tools for speech and audio coding standardization"*.

UTF-8

Information on the UCS Transformation Format 8 (UTF-8) is available at

http://www.stonehand.com/unicode/standard/utf8.html

1016

The U.S. DoD's Federal-Standard-1016 based 4800 bps code excited linear prediction voice coder version 3.2 (CELP 3.2) Fortran and C simulation source codes are available for worldwide distribution at no charge (on DOS diskettes, but configured to compile on Sun SPARC stations) from: Bob Fenichel, National Communications System, Washington, D.C. 20305, phone +1-703-692-2124, fax +1-703-746-4960.

An implementation is also available at

ftp://ftp.super.org/pub/speech/celp_3.2a.tar.Z

DVI4

An implementation is available from Jack Jansen at

ftp://ftp.cwi.nl/local/pub/audio/adpcm.shar

G722

An implementation of the G.722 algorithm is available as part of the ITU-T STL, described above.

G723

The reference C code implementation defining the G.723.1 algorithm and its Annexes A, B, and C are available as an integral part of Recommendation G.723.1 from the ITU Sales Service, address listed above. Both the algorithm and C code are covered by a specific license. The ITU-T Secretariat should be contacted to obtain such licensing information.

G726-16 through G726-40

G726-16 through G726-40 are specified in the ITU-T Recommendation G.726, "40, 32, 24, and 16 kb/s Adaptive Differential Pulse Code Modulation (ADPCM)". An implementation of the G.726 algorithm is available as part of the ITU-T STL, described above.

G727-16 through G727-40

G727-16 through G727-40 are specified in the ITU-T Recommendation G.727, "5-, 4-, 3-, and 2-bit/sample embedded adaptive differential pulse code modulation". An implementation of the G.727 algorithm will be available in a future release of the ITU-T STL, described above.

G729

The reference C code implementation defining the G.729 algorithm and its Annexes A and B are available as an integral part of Recommendation G.729 from the ITU Sales Service, listed above. Both the algorithm and the C code are covered by a specific license. The contact information for obtaining the license is listed in the C code.

GSM

A reference implementation was written by Carsten Borman and Jutta Degener (TU Berlin, Germany). It is available at

ftp://ftp.cs.tu-berlin.de/pub/local/kbs/tubmik/gsm/

Although the RPE-LTP algorithm is not an ITU-T standard, there is a C code implementation of the RPE-LTP algorithm available as part of the ITU-T STL. The STL implementation is an adaptation of the TU Berlin version.

LPC

An implementation is available at

ftp://parcftp.xerox.com/pub/net-research/lpc.tar.Z

PCMU, PCMA

An implementation of these algorithm is available as part of the ITU-T STL, described above. Code to convert between linear and mu-law companded data is also available in [11].