Internet Engineering Task Force
INTERNET-DRAFT
draft-ietf-avt-encodings-02.txt

Audio-Video Transport Working Group
H. Schulzrinne
AT&T Bell Laboratories
September 17, 1993
Expires: 10/01/93

Media Encodings

Status of this Memo

Abstract

This document describes a possible structure of the media content for audio and video for Internet applications. The definitions are independent of the particular transport mechanism used. 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.

1 Audio

1.1 Encoding-independent recommendations

The following recommendations are default operating parameters. An

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 application 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, unless otherwise noted in Table 1. 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 links with severe bandwidth constraints, a higher packetization delay may be appropriate. For frame-based encodings (marked as F in the table 1 below) such as LPC, CELP and GSM, the sender may choose to combine several frame intervals into a single message. The receiver can tell the number of frames contained in a message since the frame duration is defined as part of the encoding.

If multiple channels are used, the left channel information always precedes the right-channel information. For more than two channels, the convention followed by the AIFF-C audio interchange format should be followed. It is listed in the table below. (The AIFF-C specification is available by anonymous ftp at ftp.sgi.com in the file sgi/aiff-c.9.26.91.ps.)

type	channels				
stereo	left	right			
3 channel	left	right	center		
quad	front left	front right	rear left	rear right	
4 channel	left	center	right	surround	
6 channel	left	left center	center	right	right center
surround					

The sampling frequency should be drawn from the set: 8, 11.025, 16, 22.05, 44.1 and 48 kHz.

1.2 Recommended Audio Encodings

The table 1 shows the names, types (sample vs. frame oriented), per-channel bit rates and default sampling frequencies of recommended encodings. The list is partially drawn from the document ``Recommended practices for

enhancing digital audio compatibility in multimedia systems'', published by the Interactive Multimedia Assocation, Version 3.00, Oct. 1992 (referenced as [IMA]). The names are for identification only; they correspond to the names used within the Real-Time Transport Protocol (RTP). Other applications may choose different namings. Note that the L16 encoding may be used with different sampling rates.

name	nom. sampling kHz	rate kb/s	type S/F	frame ms	description
L16	48	768	S		16-bit linear, 2's complement
	44.1	705.6	S		
	22.05	352.8	S		
	11.025	176.4	S		
G722	16	64	S		CCITT subband ADPCM
PCMU	8	64	S		CCITT mu-law PCM
PCMA	8	64	S		CCITT A-law PCM
G721	8	32	S		CCITT ADPCM
IDVI	8	32	S		Intel/DVI ADPCM [IMA]
G723	8	24	S		CCITT ADPCM
GSM	8	13	F	20	RTE/LTP GSM 06.10
1016	8	4.8	F	30	CELP

Table 1: Audio encodings

For multi-octet encodings, octets are transmitted in network byte order (i.e., most significant octet first).

A detailed description of the encodings is given below. The names shown (L16, PCMU, etc.) are limited to four characters and suitable to be used for identification in protocols such as RTP (RFC TBD).

L16: denotes uncompressed audio data, 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.

PCMU: specified in CCITT recommendation G.711. Audio data is encoded as eight bits per sample, after companding. Code to convert between linear and mu-law companded data is available in the IMA document.

PCMA: specified in CCITT recommendation G.711. Audio data is encoded as eight bits per sample, after companding. Code to convert between linear and A-law companded data is available in the IMA document.

G721 through G729: specified in the corresponding CCITT recommendations. Reference implementations for G.721 and G.723 are available as part of

the CCITT Software Tool Library (STL) from the ITU General Secretariat, Sales Service, Place du Nations, CH-1211 Geneve 20, Switzerland. The

library is covered by a license and is available for anonymous ftp on gaia.cs.umass.edu, file pub/ccitt/ccitt_tools.tar.Z.

GSM: (group speciale mobile) denotes the European GSM 06.10 provisional standard for full-rate speech transcoding, prI-ETS 300 036, which is based on RPE/LTP (residual pulse excitation/long term prediction) coding at a rate of 13 kb/s. A reference implementation was written by Carsten Borman and Jutta Degener (TU Berlin, Germany) and is available for anonymous ftp from tub.cs.tu-berlin.de, directory tub/tubmik.

1016: uses code-excited linear prediction (CELP) and is specified in Federal Standard FED-STD 1016, published by the Office of Technology and Standards, Washington, DC 20305-2010.

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.

Example input and processed speech files, a technical information bulletin, and the official standard ``Federal Standard 1016, Telecommunications: Analog to Digital Conversion of Radio Voice by 4,800 bit/second Code Excited Linear Prediction (CELP)'' are included at no charge. According to Vincent Cate (Carnegie Mellon), the distribution is also available for anonymous ftp at furmint.nectar.cs.cmu.edu (128.2.209.111) in directory celp.audio.compression.

The following articles describes the Federal-Standard-1016 4.8-kbps CELP coder:

Campbell, Joseph P. Jr., Thomas E. Tremain and Vanoy C. Welch, ``The Proposed Federal Standard 1016 4800 bps Voice Coder: CELP,'' S_p_e_e_c_h_ T_e_c_h_n_o_l_o_g_y_ M_a_g_a_z_i_n_e_, April/May 1990, p. 58-64.

Campbell, Joseph P. Jr., Thomas E. Tremain and Vanoy C. Welch, ``The Federal Standard 1016 4800 bps CELP Voice Coder,'' D_i_g_i_t_a_l_ S_i_g_n_a_l_ P_r_o_c_e_s_s_i_n_g_, Academic Press, 1991, Vol. 1, No. 3, p. 145-155.

Campbell, Joseph P. Jr., Thomas E. Tremain and Vanoy C. Welch, ``The DoD 4.8 kbps Standard (Proposed Federal Standard 1016),'' in A_d_v_a_n_c_e_s_ i_n_ S_p_e_e_c_h_ C_o_d_i_n_g_, ed. Atal, Cuperman and Gersho, Kluwer Academic Publishers, 1991, Chapter 12, p. 121-133.

Campbell, Joseph P. Jr., Thomas E. Tremain and Vanoy C. Welch, ``The

Proposed Federal Standard 1016 4800 bps Voice Coder: CELP,'' S_p_e_e_c_h_ T_e_c_h_n_o_l_o_g_y_ M_a_g_a_z_i_n_e_, April/May 1990, p. 58-64.

Copies of the FS-1016 document are available for $2.50 each from:

GSA Rm 6654
7th & D St SW
Washington, D.C. 20407
1-202-708-9205

DVI: is specified in the ``Recommended Practices for Enhancing Digital Audio Compatibility in Multimedia Systems'', published by the Interactive Multimedia Association (IMA), Annapolis, MD. The document also contains reference implementations for mu-law to 16-bit, ADPCM and sample rate conversions.

For sample-based encodings, a receiver should accept packets representing between 0 and 200 ms of audio data.(1) Receivers should be prepared to accept multi-channel audio, but may choose to only play a single channel.

1.3 Application Programming Interface for Audio Codecs

The application programming interface (API) for audio codecs described here is suggested, but not required for interoperability. The API shown here is similar to the one used by SunOS 4.1. The encoding types are drawn from the standard names defined here.

```
typedef {AE_PCMU = 1, AE_PCMA, AE_L16} encoding_t;

typedef struct {
  unsigned sample_rate;            /* samples per second */
  unsigned samples_per_unit;       /* samples per unit */
  unsigned bytes_per_unit;         /* bytes per sample unit */
  unsigned channels;               /* # of interleaved channels */
  encoding_t encoding;             /* data encoding format */
  unsigned data_size;              /* length of data (optional) */
} audio_descr_t;

void *x_init(void *state, double period);

int x_encode(void *in_buf, int in_size, audio_descr_t *in_descr,
   void *out_buf, int *out_size, void *state);
int x_decode(void *in_buf, int in_size, audio_descr_t *out_descr,
   void *out_buf, int *out_size, void *state);
```

---

1. This restriction allows reasonable buffer sizing for the receiver.

x_init initializes a particular instance of a codec. If the argument state is zero, a memory area sufficient to hold the encoder or decoder state is allocated; if that argument is non-zero, the existing area is reinitialized. The function returns a pointer to the area, zero if the state area could not be allocated. The argument period refers to the amount of audio data in each block, measured in seconds. It is typically only used for block-oriented codecs.

The generic pointer to state refers to an area of storage whose structure is opaque to the application program. In the functions, 'x' is replaced by the appropriate codec name, appropriately modified to conform to C syntax (e.g., g711, g721, etc).

The encoder and decoder transform the data contained in the input buffer in_buf (in_size bytes) and deposit the result into the output buffer area out_buf. The variable out_size is set to the number of bytes actually contained in the output buffer. The ah arguments points to a structure of type audio_hdr_t, which defines the given input data format for the encoder and the desired output data format for the decoder. The functions return 0 on success, a negative number if a failure occurred.

All block-oriented audio codecs should be able to encode and decode several consecutive blocks.

2 Video

The following video encodings are defined, with their abbreviated names used for identification:

Bolt: The encoding is implemented by the Bolter video codec [ED: need more info on company, designation].

JPEG: The encoding is specified in ISO Standards DIS 10918-1 and DIS 10918-2. The data is formatted according to the JFIF (JPEG File Interchange Format) defined by C-Cube Microsystems.

H261: The encoding is specified in ITU-T (formerly CCITT) standard H.261. The packetization and RTP-specific properties are described in RFC TBD.

nv: The encoding is implemented in the program 'nv' developed at Xerox PARC by Ron Frederick.

CUSM: The encoding is implemented in the program CU-SeeMe developed at Cornell University by Dick Cogger, Scott Brim, Tim Dorcey and John Lynn.

PicW: The encoding is implemented in the program PictureWindow developed at Bolt, Beranek and Newman (BBN).

3 Address of Author

Henning Schulzrinne
AT&T Bell Laboratories
MH 2A244
600 Mountain Avenue
Murray Hill, NJ 07974-0636
telephone: +1 908 582 2262
facsimile: +1 908 582 5809
electronic mail: hgs@research.att.com