Network Working Group
Request for Comments: 3952
Category: Experimental

A. Duric
Telio
S. Andersen
Aalborg University

December 2004

Real-time Transport Protocol (RTP) Payload Format for internet Low Bit Rate Codec (iLBC) Speech

Status of this Memo

Copyright Notice

Abstract

This document describes the Real-time Transport Protocol (RTP) payload format for the internet Low Bit Rate Codec (iLBC) Speech developed by Global IP Sound (GIPS). Also, within the document there are included necessary details for the use of iLBC with MIME and Session Description Protocol (SDP).

Table of Contents

1. Introduction

This document describes how compressed iLBC speech, as produced by the iLBC codec [1], may be formatted for use as an RTP payload type. Methods are provided to packetize the codec data frames into RTP packets. The sender may send one or more codec data frames per packet depending on the application scenario or based on the transport network condition, bandwidth restriction, delay requirements and packet-loss tolerance.

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 BCP 14, RFC 2119 [2].

2. Background

Global IP Sound (GIPS) has developed a speech compression algorithm for use in IP based communications [1]. The iLBC codec enables graceful speech quality degradation in the case of lost frames, which occurs in connection with lost or delayed IP packets.

This codec is suitable for real time communications such as, telephony and videoconferencing, streaming audio, archival and messaging.

The iLBC codec [1] is an algorithm that compresses each basic frame (20 ms or 30 ms) of 8000 Hz, 16-bit sampled input speech, into output frames with rate of 400 bits for 30 ms basic frame size and 304 bits for 20 ms basic frame size.

The codec supports two basic frame lengths: 30 ms at 13.33 kbit/s and 20 ms at 15.2 kbit/s, using a block independent linear-predictive coding (LPC) algorithm. When the codec operates at block lengths of 20 ms, it produces 304 bits per block which MUST be packetized in 38 bytes. Similarly, for block lengths of 30 ms it produces 400 bits per block which MUST be packetized in 50 bytes. This algorithm results in a speech coding system with a controlled response to packet losses similar to what is known from pulse code modulation (PCM) with a packet loss concealment (PLC), such as ITU-T G711 standard [7], which operates at a fixed bit rate of 64 kbit/s. At the same time, this algorithm enables fixed bit rate coding with a quality-versus-bit rate tradeoff close to what is known from code-excited linear prediction (CELP).

3. RTP Payload Format

The iLBC codec uses 20 or 30 ms frames and a sampling rate clock of 8 kHz, so the RTP timestamp MUST be in units of 1/8000 of a second. The RTP payload for iLBC has the format shown in the figure bellow. No addition header specific to this payload format is required.

This format is intended for the situations where the sender and the receiver send one or more codec data frames per packet. The RTP packet looks as follows:

```
 0                   1                   2                   3
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                      RTP Header [3]                           |
+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+
|                                                               |
+                 one or more frames of iLBC [1]                |
|                                                               |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Figure 1, Packet format diagram

The RTP header of the packetized encoded iLBC speech has the expected values as described in [3]. The usage of M bit SHOULD be as specified in the applicable RTP profile, for example, RFC 3551 [4] specifies that if the sender does not suppress silence (i.e., sends a frame on every frame interval), the M bit will always be zero. When more then one codec data frame is present in a single RTP packet, the timestamp is, as always, the oldest data frame represented in the RTP packet.

The assignment of an RTP payload type for this new 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 by the sender.

3.1. Bitstream definition

The total number of bits used to describe one frame of 20 ms speech is 304, which fits in 38 bytes and results in a bit rate of 15.20 kbit/s. For the case with a frame length of 30 ms speech the total number of bits used is 400, which fits in 50 bytes and results in a bit rate of 13.33 kbit/s. In the bitstream definition, the bits are distributed into three classes according to their bit error or loss sensitivity. The most sensitive bits (class 1) are placed first in the bitstream for each frame. The less sensitive bits (class 2) are placed after the class 1 bits. The least sensitive bits (class 3) are placed at the end of the bitstream for each frame.

Looking at the 20/30 ms frame length cases for each class: The class 1 bits occupy a total of 6/8 bytes (48/64 bits), the class 2 bits occupy 8/12 bytes (64/96 bits), and the class 3 bits occupy 24/30 bytes (191/239 bits). This distribution of the bits enables the use of uneven level protection (ULP). The detailed bit allocation is shown in the table below. When a quantization index is distributed between more classes the more significant bits belong to the lowest class.

Bitstream structure:

Parameter			Bits Class <1,2,3> 20 ms frame	Bits Class <1,2,3> 30 ms frame
		Split 1	6 <6,0,0>	6 <6,0,0>
	LSF 1	Split 2	7 <7,0,0>	7 <7,0,0>
LSF		Split 3	7 <7,0,0>	7 <7,0,0>
		Split 1	NA (Not Appl.)	6 <6,0,0>
	LSF 2	Split 2	NA	7 <7,0,0>
		Split 3	NA	7 <7,0,0>
	Sum		20 <20,0,0>	40 <40,0,0>
Block Class.			2 <2,0,0>	3 <3,0,0>
Position 22 sample segment			1 <1,0,0>	1 <1,0,0>
Scale Factor State Coder			6 <6,0,0>	6 <6,0,0>
	Sample 0		3 <0,1,2>	3 <0,1,2>
Quantized	Sample 1		3 <0,1,2>	3 <0,1,2>
Residual	:		: :	: :
State	:		: :	: :
Samples	:		: :	: :
	Sample 56		3 <0,1,2>	3 <0,1,2>
	Sample 57		NA	3 <0,1,2>
	Sum		171 <0,57,114>	174 <0,58,116>
		Stage 1	7 <6,0,1>	7 <4,2,1>
CB for 22/23		Stage 2	7 <0,0,7>	7 <0,0,7>
sample block		Stage 3	7 <0,0,7>	7 <0,0,7>
	Sum		21 <6,0,15>	21 <4,2,15>
		Stage 1	5 <2,0,3>	5 <1,1,3>
Gain for 22/23		Stage 2	4 <1,1,2>	4 <1,1,2>
sample block		Stage 3	3 <0,0,3>	3 <0,0,3>
	Sum		12 <3,1,8>	12 <2,2,8>
		Stage 1	8 <7,0,1>	8 <6,1,1>
	sub-block 1	Stage 2	7 <0,0,7>	7 <0,0,7>
		Stage 3	7 <0,0,7>	7 <0,0,7>

```
                            Stage 1 |   8 <0,0,8>   |   8 <0,7,1>   |
              sub-block 2   Stage 2 |   8 <0,0,8>   |   8 <0,0,8>   |
   Indices                  Stage 3 |   8 <0,0,8>   |   8 <0,0,8>   |
   for CB         ------------------+---------------+---------------+
   sub-blocks               Stage 1 |      NA       |   8 <0,7,1>   |
              sub-block 3   Stage 2 |      NA       |   8 <0,0,8>   |
                            Stage 3 |      NA       |   8 <0,0,8>   |
                  ------------------+---------------+---------------+
                            Stage 1 |      NA       |   8 <0,7,1>   |
              sub-block 4   Stage 2 |      NA       |   8 <0,0,8>   |
                            Stage 3 |      NA       |   8 <0,0,8>   |
                  ------------------+---------------+---------------+
                  Sum               |  46 <7,0,39>  |  94 <6,22,66> |
   ---------------------------------+---------------+---------------+
                            Stage 1 |   5 <1,2,2>   |   5 <1,2,2>   |
              sub-block 1   Stage 2 |   4 <1,1,2>   |   4 <1,2,1>   |
                            Stage 3 |   3 <0,0,3>   |   3 <0,0,3>   |
                  ------------------+---------------+---------------+
                            Stage 1 |   5 <1,1,3>   |   5 <0,2,3>   |
              sub-block 2   Stage 2 |   4 <0,2,2>   |   4 <0,2,2>   |
                            Stage 3 |   3 <0,0,3>   |   3 <0,0,3>   |
   Gains for      ------------------+---------------+---------------+
   sub-blocks               Stage 1 |      NA       |   5 <0,1,4>   |
              sub-block 3   Stage 2 |      NA       |   4 <0,1,3>   |
                            Stage 3 |      NA       |   3 <0,0,3>   |
                  ------------------+---------------+---------------+
                            Stage 1 |      NA       |   5 <0,1,4>   |
              sub-block 4   Stage 2 |      NA       |   4 <0,1,3>   |
                            Stage 3 |      NA       |   3 <0,0,3>   |
                  ------------------+---------------+---------------+
                  Sum               |  24 <3,6,15>  |  48 <2,12,34> |
   -------------------------------------------------------------------
   Empty frame indicator            |   1 <0,0,1>   |   1 <0,0,1>   |
   -------------------------------------------------------------------
   SUM                                304 <48,64,192> 400 <64,96,240>
```

Table 3.1 The bitstream definition for iLBC.

When packetized into the payload, all the class 1 bits MUST be sorted in order (from top and down) as they were specified in the table. Additionally, all the class 2 bits MUST be sorted (from top and down) and all the class 3 bits MUST be sorted in the same sequential order.

3.2. Multiple iLBC frames in a RTP packet

More than one iLBC frame may be included in a single RTP packet by a sender.

It is important to observe that senders have the following additional restrictions:

- o SHOULD NOT include more iLBC frames in a single RTP packet than will fit in the MTU of the RTP transport protocol.

- o Frames MUST NOT be split between RTP packets.

- o Frames of the different modes (20 ms and 30 ms) MUST NOT be included within the same packet.

It is RECOMMENDED that the number of frames contained within an RTP packet are consistent with the application. For example, in telephony and other real time applications where delay is important, the delay is lower depending on the amount of frames per packet (i.e., fewer frames per packet, the lower the delay). Whereas for bandwidth constrained links or delay insensitive streaming messaging application, one or more frames per packet would be acceptable.

Information describing the number of frames contained in an RTP packet is not transmitted as part of the RTP payload. The way to determine the number of iLBC frames is to count the total number of octets within the RTP packet, and divide the octet count by the number of expected octets per frame (32/50 per frame).

4. IANA Considerations

One new MIME sub-type as described in this section has been registered.

4.1. Storage Mode

The storage mode is used for storing speech frames (e.g., as a file or email attachment).

```
+------------------+
| Header           |
+------------------+
| Speech frame 1   |
+------------------+
:                  :
+------------------+
| Speech frame n   |
+------------------+
```

Figure 2, Storage format diagram

The file begins with a header that includes only a magic number to identify that it is an iLBC file.

The magic number for iLBC file MUST correspond to the ASCII character string:

- o for 30 ms frame size mode:"#!iLBC30\n", or "0x23 0x21 0x69 0x4C 0x42 0x43 0x33 0x30 0x0A" in hexadecimal form,

- o for 20 ms frame size mode:"#!iLBC20\n", or "0x23 0x21 0x69 0x4C 0x42 0x43 0x32 0x30 0x0A" in hexadecimal form.

After the header, follow the speech frames in consecutive order.

Speech frames lost in transmission MUST be stored as "empty frames", as defined in [1].

4.2. MIME Registration of iLBC

MIME media type name: audio

MIME subtype: iLBC

Optional parameters:

All of the parameters does apply for RTP transfer only.

maxptime: The maximum amount of media which can be encapsulated in each packet, expressed as time in milliseconds. The time SHALL be calculated as the sum of the time the media present in the packet represents. The time SHOULD be a multiple of the frame size. This attribute is probably only meaningful for audio data, but may be used with other media types if it makes sense. It is a media attribute, and is not dependent on charset. Note that this attribute was introduced after RFC 2327, and non updated implementations will ignore this attribute.

mode: The iLBC operating frame mode (20 or 30 ms) that will be encapsulated in each packet. Values can be 0, 20 and 30 (where 0 is reserved, 20 stands for preferred 20 ms frame size and 30 stands for preferred 30 ms frame size).

ptime: Defined as usual for RTP audio (see [5]).

Encoding considerations:
This type is defined for transfer via both RTP (RFC 3550) and stored-file methods as described in Section 4.1, of RFC

3952. Audio data is binary data, and must be encoded for non-binary transport; the Base64 encoding is suitable for email.

Security considerations:
See Section 6 of RFC 3952.

Public specification:
Please refer to RFC 3951 [1].

Additional information:
The following applies to stored-file transfer methods:

Magic number:
ASCII character string for:
o 30 ms frame size mode "#!iLBC30\n" (or 0x23 0x21 0x69 0x4C 0x42 0x43 0x33 0x30 0x0A in hexadecimal)
o 20 ms frame size mode "#!iLBC20\n" (or 0x23 0x21 0x69 0x4C 0x42 0x43 0x32 0x30 0x0A in hexadecimal)

File extensions: lbc, LBC
Macintosh file type code: none
Object identifier or OID: none

Person & email address to contact for further information:
alan.duric@telio.no

Intended usage: COMMON.
It is expected that many VoIP applications will use this type.

Author/Change controller:
alan.duric@telio.no
IETF Audio/Video transport working group

5. Mapping To SDP Parameters

The information carried in the MIME media type specification has a specific mapping to fields in the Session Description Protocol (SDP) [5], which is commonly used to describe RTP sessions. When SDP is used to specify sessions employing the iLBC codec, the mapping is as follows:

- o The MIME type ("audio") goes in SDP "m=" as the media name.

- o The MIME subtype (payload format name) goes in SDP "a=rtpmap" as the encoding name.

o The parameters "ptime" and "maxptime" go in the SDP "a=ptime" and "a=maxptime" attributes, respectively.

o The parameter "mode" goes in the SDP "a=fmtp" attribute by copying it directly from the MIME media type string as "mode=value".

When conveying information by SDP, the encoding name SHALL be "iLBC" (the same as the MIME subtype).

An example of the media representation in SDP for describing iLBC might be:

```
m=audio 49120 RTP/AVP 97
a=rtpmap:97 iLBC/8000
```

If 20 ms frame size mode is used, remote iLBC encoder SHALL receive "mode" parameter in the SDP "a=fmtp" attribute by copying them directly from the MIME media type string as a semicolon separated with parameter=value, where parameter is "mode", and values can be 0 and 20 (where 0 is reserved and 20 stands for preferred 20 ms frame size). An example of the media representation in SDP for describing iLBC when 20 ms frame size mode is used might be:

```
m=audio 49120 RTP/AVP 97
a=rtpmap:97 iLBC/8000
a=fmtp:97 mode=20
```

It is important to emphasize the bi-directional character of the "mode" parameter - both sides of a bi-directional session MUST use the same "mode" value.

The offer contains the preferred mode of the offerer. The answerer may agree to that mode by including the same mode in the answer, or may include a different mode. The resulting mode used by both parties SHALL be the lower of the bandwidth modes in the offer and answer.

That is, an offer of "mode=20" receiving an answer of "mode=30" will result in "mode=30" being used by both participants. Similarly, an offer of "mode=30" and an answer of "mode=20" will result in "mode=30" being used by both participants.

This is important when one end point utilizes a bandwidth constrained link (e.g., 28.8k modem link or slower), where only the lower frame size will work.

Parameter ptime can not be used for the purpose of specifying iLBC operating mode, due to fact that for the certain values it will be impossible to distinguish which mode is about to be used (e.g., when ptime=60, it would be impossible to distinguish if packet is carrying 2 frames of 30 ms or 3 frames of 20 ms, etc.).

Note that the payload format (encoding) names are commonly shown in upper case. MIME subtypes are commonly shown in lower case. These names are case-insensitive in both places. Similarly, parameter names are case-insensitive both in MIME types and in the default mapping to the SDP a=fmtp attribute

6. Security Considerations

RTP packets using the payload format defined in this specification are subject to the general security considerations discussed in [3] and any appropriate profile (e.g., [4]).

As this format transports encoded speech, the main security issues include confidentiality and authentication of the speech itself. The payload format itself does not have any built-in security mechanisms. Confidentiality of the media streams is achieved by encryption, therefore external mechanisms, such as SRTP [6], MAY be used for that purpose. The data compression used with this payload format is applied end-to-end; hence encryption may be performed after compression with 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 which are complex to decode and cause the receiver to become overloaded. However, the encodings covered in this document do not exhibit any significant non-uniformity.

7. References

7.1. Normative References

[1] Andersen, S., Duric, A., Astrom, H., Hagen, R., Kleijn, W., and J. Linden, "Internet Low Bit Rate Codec (iLBC)", RFC 3951, December 2004.

[2] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, March 1997.

[3] Schulzrinne, H., Casner, S., Frederick, R., and V. Jacobson, "RTP: A Transport Protocol for Real-Time Applications", STD 64, RFC 3550, July 2003.

[4] Schulzrinne, H. and S. Casner, "RTP Profile for Audio and Video Conferences with Minimal Control", STD 65, RFC 3551, July 2003.

[5] Handley, M. and V. Jacobson, "SDP: Session Description Protocol", RFC 2327, April 1998.

[6] Baugher, M., McGrew, D., Naslund, M., Carrara, E., and K. Norrman, "The Secure Real-time Transport Protocol", RFC 3711, March 2004.

7.2. Informative References

[7] ITU-T Recommendation G.711, available online from the ITU bookstore at http://www.itu.int.

8. Acknowledgements

Henry Sinnreich, Patrik Faltstrom, Alan Johnston and Jean-Francois Mule for great support of the iLBC initiative and for valuable feedback and comments.

Authors' Addresses

Alan Duric
Telio AS
Stoperigt. 2
Oslo, N-0250
Norway

Phone: +47 21673505
EMail: alan.duric@telio.no

Soren Vang Andersen
Department of Communication Technology
Aalborg University
Fredrik Bajers Vej 7A
9200 Aalborg
Denmark

Phone: ++45 9 6358627
EMail: sva@kom.auc.dk

Full Copyright Statement

Intellectual Property

Acknowledgement

Funding for the RFC Editor function is currently provided by the Internet Society.