          RTP Payload Format for mU-law EMbedded Codec for Low-delay IP
                    Communication (UEMCLIP) Speech Codec

Abstract

   This document describes the RTP payload format of a mU-law EMbedded
   Coder for Low-delay IP communication (UEMCLIP), an enhanced speech
   codec of ITU-T G.711. The bitstream has a scalable structure with an
   embedded u-law bitstream, also known as PCMU, thus providing a handy
   transcoding operation between narrowband and wideband speech.

Status of This Memo

   This document specifies an Internet standards track protocol for the
   Internet community, and requests discussion and suggestions for
   improvements. Please refer to the current edition of the "Internet
   Official Protocol Standards" (STD 1) for the standardization state
   and status of this protocol. Distribution of this memo is unlimited.

not be created outside the IETF Standards Process, except to format
it for publication as an RFC or to translate it into languages other
than English.

Table of Contents

1. Introduction

   This document specifies the payload format for sending UEMCLIP-
   encoded (mU-law EMbedded Coder for Low-delay IP communication) speech
   using the Real-time Transport Protocol (RTP) [RFC3550]. UEMCLIP is a
   proprietary codec that enhances u-law ITU-T G.711 [ITU-T-G.711] and
   that is designed to help the market for smooth transition towards the
   forthcoming wideband communication environment while achieving a very
   small media transcoding load with the existing terminals, in which
   the implementation of G.711 is mandatory.

   It should be noted that, generally speaking, codecs are negotiated
   and changed using an SDP exchange. Also, [RFC3550] defines general
   RTP mixer and translator models, where media transcoding may not take
   place at the node. For those cases, the design concept of the
   embedded structure is not useful. However, there are other cases
   when costly transcoding is unavoidable in commonly deployed types of
   Multi-point Control Units (MCUs), which terminate media and RTCP

packets [RFC5117], and when narrowband and wideband terminals
coexist. This embedded bitstream structure can reduce the media
transcoding to a simple bitstream truncation.

The background and the basic idea of the media format is described in
Section 2. The details of the payload format are given in Section 3.
The transcoding issues with G.711 are discussed in Section 4, and the
considerations for congestion control are in Section 5. In
Section 6, the payload format parameters for a media type
registration for UEMCLIP RTP payload format and Session Description
Protocol (SDP) mappings are provided. The security considerations
and IANA considerations are dealt with in Section 7 and Section 8,
respectively.

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 [RFC2119].

2. Media Format Background

UEMCLIP is an enhanced version of u-law ITU-T G.711, otherwise known
as PCMU [RFC4856]. It is targeted at Voice over Internet Protocol
(VoIP) applications, and its main goal is to provide a wideband
communication platform that is highly interoperable with existing
terminals equipped with G.711 and to stimulate the market to
gradually shift to using wideband communication. In widely deployed
multi-point conferencing systems, the packets usually go through
RTCP-terminating (RTP Control Protocol) MCUs, "Topo-RTCP-terminating-
MCU" as defined in [RFC5117]. Because the G.711 bitstream is
embedded in the bitstream, costly media transcoding can be avoided in
this case.

This document does not discuss the implementation details of the
encoder and decoder, but only describes the bitstream format.

Because of its scalable nature, there are a number of sub-bitstreams
(sub-layer) in a UEMCLIP bitstream. By choosing appropriate sub-
layers, the codec can adapt to the following requirements:

o Sampling frequency,

o Number of channels,

o Speech quality, and

o Bit-rate.

The UEMCLIP codec operates at a 20-ms frame, and includes three sub-coders as shown in Table 1. The core layer is u-law G.711 at 64 kbit/s, and other two are quality and bandwidth enhancement layers with bit-rate of 16 kbit/s each.

Layer	Description	Bit-rate	Coding algorithm
a	G.711 core	64	u-law PCM
b	Lower-band enhancement	16	Time domain block quantization
c	Higher-band	16	MDCT block quantization

Table 1: Sub-Layer Description

Based on these sub-layers, the UEMCLIP codec operates in four modes as shown in Table 2. Here, "Ch" is the number of channels and "Fs" is the sampling frequency in kHz. It should be noted that the current version only supports single-channel operation and there might be future extensions with multi-channel capabilities. The absent Modes 2 and 5 are reserved for possible future extension to 32 kHz sampling modes. As the mode definition is expected to grow, any other modes not defined in this table MUST NOT be used for compatibility and interoperability reasons.

Mode	Ch	Fs	Layer a	Layer b	Layer c	Bit-rate w/o headers [kbit/s]	Total bit-rate [kbit/s]
0	1	8	x	-	-	64	67.2
1	1	16	x	-	x	80	84.0
2	-	-	-	-	-	-	-
3	1	8	x	x	-	80	84.0
4	1	16	x	x	x	96	100.8
5	-	-	-	-	-	-	-

Table 2: Mode Description

The UEMCLIP bitstream contains internal headers and other side-
information apart from the layer data. This results in total bit-
rate larger than the sum of the layers shown in the above table. The
detail of the internal headers and auxiliary information are
described in Section 3.3.1.

Defining the sampling frequency and the number of channels does not
result in a singular mode, i.e., there can be multiple modes for the
same sampling frequency or number of channels. The supported modes
would differ between implementations; thus, the sender and the
receiver must negotiate what mode to use for transmission.

3. Payload Format

   As an RTP payload, the UEMCLIP bitstream can contain one or more
   frames as shown in Figure 1.

```
 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                             |
+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+
|                                                               |
|              one or more frames of UEMCLIP                    |
|                                                               |
+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+
```
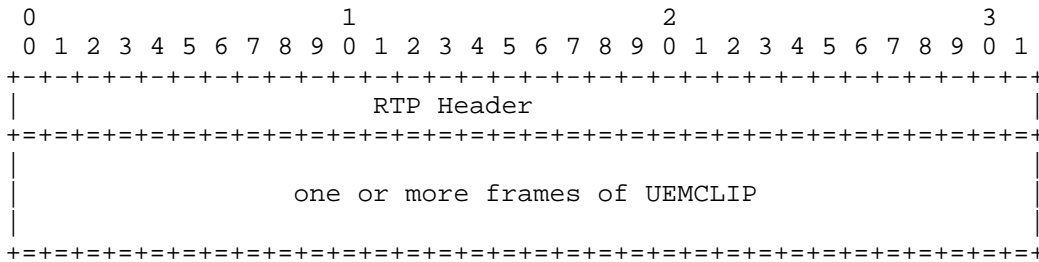
                    Figure 1: RTP Payload Format

   The UEMCLIP bitstream has a scalable structure; thus, it is possible
   to reconstruct the signal by decoding a part of it. A UEMCLIP frame
   is composed of a main header (MH) followed by one or more (up to
   three) sub-layers (SLs) as shown in Figure 2.

```
              +--+-------+//-+
              |MH| SL #1 |...|
              +--+-------+//-+
```
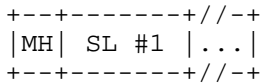
               Figure 2: A UEMCLIP Frame (Bitstream Format)

   As a sub-layer, the core layer, i.e., "Layer a", MUST always be
   included. It should be noted that the location of the core layer may
   or may not immediately follow MH field. The decoder MUST always
   refer to the layer indices for proper decoding because the order of
   the sub-layers is arbitrary.

The UEMCLIP bitstream does not explicitly include the following
information: mode and sampling frequency (Fs). As described before,
this information MUST be exchanged while establishing a connection,
for example, by means of SDP.

3.1. RTP Header Usage

Each RTP packet starts with a fixed RTP header, as explained in
[RFC3550]. The following fields of the RTP fixed header used
specifically for UEMCLIP streams are emphasized:

Payload type: The assignment of an RTP payload type for this packet
   format is outside the scope of this document; however, it is
   expected that a payload type in the dynamic range shall be
   assigned.

Timestamp: This encodes the sampling instant of the first speech
   signal sample in the RTP data packet. For UEMCLIP streams, the
   RTP timestamp MUST advance based on a clock either at 8000 or
   16000 (Hz). In cases where the audio sampling rate can change
   during a session, the RTP timestamp rate MUST be equal to the
   maximum rate (in Hz) given in the mode range (see Section 6.2.1).
   This implies that the RTP timestamp rate for UEMCLIP payload type
   MUST NOT change during a session. For example, for a UEMCLIP
   stream with 8-kHz audio sampling, where a transition to a 16-kHz
   audio sampling mode is allowed, the RTP time stamp must always
   advance using the 16-kHz clock rate. For a fixed audio sampling
   mode, the RTP timestamp rate should be either 8 or 16 kHz,
   depending on the sampling rate.

Marker bit: If the codec is used for applications with discontinuous
   transmission (DTX, or silence compression), the first packet after
   a silence period during which packets have not been transmitted
   contiguously SHOULD have the marker bit in the RTP data header set
   to one. The marker bit in all other packets MUST be zero.
   Applications without DTX MUST set the marker bit to zero.

3.2. Multiple Frames in an RTP Packet

More than one UEMCLIP frame may be included in a single RTP packet by
a sender. However, senders have the following additional
restrictions:

o A single RTP packet SHOULD NOT include more UEMCLIP frames than
   will fit in the path MTU.

o All frames contained in a single RTP packet MUST be of the same
   mode.

   o Frames MUST NOT be split between RTP packets.

   It is RECOMMENDED that the number of frames contained within an RTP
   packet be consistent with the application. Since UEMCLIP is designed
   for telephony applications where delay has a great impact on the
   quality, then fewer frames per packet for lower delay, is preferable.

3.3. Payload Data

   In a UEMCLIP bitstream, all numbers are encoded in a network byte
   order.

3.3.1. Main Header

   The main header (MH) is placed at the top of a frame and has a size
   of 6 bytes. The content of the main header is shown in Figure 3.

```
 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
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|     MX        |                   PC                          |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|         PC(cont'd)          |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```
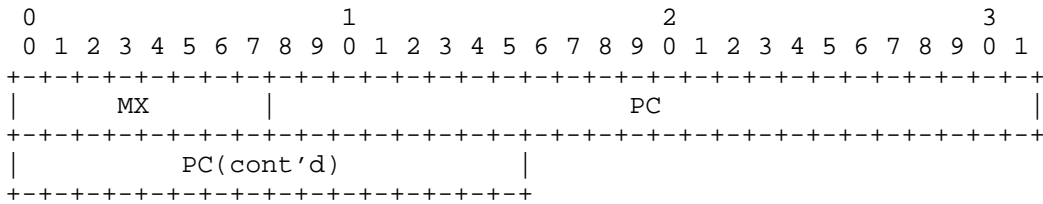
                  Figure 3: UEMCLIP Main Header Format (MH)

   Mixing information (MX): 8 bits

      Mixing information field. This field is only relevant when Topo-
      RTCP-terminating-MCUs are utilized to interpret these fields. See
      Section 3.3.1.1 for details of the fields.

   Packet-loss Concealment information (PC): 40 bits

      Packet-loss concealment (PLC) information field. See
      Section 3.3.1.2.

3.3.1.1. Mixing Information Field

```
           0 1 2 3 4 5 6 7
          +-+-+-+-+-+-+-+-+
          |C|R|V|   PW1   |
          |1|1|1|         |
          +-+-+-+-+-+-+-+-+
```
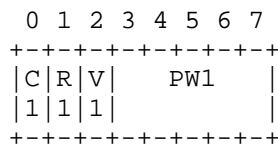
                  Figure 4: Mixing Information Field (MX)

   Check bit #1 (C1): 1 bit

      Validity flag of V1 and PW1. This bit being "1" indicates that
      both parameters are valid, and "0" indicates that the parameters
      should be ignored. If any of these parameters is invalid, this
      bit should be set to "0". This flag is mainly intended for a
      UEMCLIP-conscious Topo-RTCP-terminating-MCU. This flag should be
      set to "0" in case of upward transcoding from G.711 (see
      Section 4).

   Reserved bit #1 (R1): 1 bit

      This bit should be ignored. The default of this bit is 0.

   VAD flag #1 (V1): 1 bit

      Voice activity detection flag of the current frame, designed to be
   used for MCU operations. This flag being "1" indicates that the
      frame is an active (voice) segment, and "0" indicates that it is
      an inactive (non-voice) or a silent segment. This flag is
      specifically designed for mixing information. DTX judgment based
      this flag is not recommended.

   Power #1 (PW1): 5 bits

      Signal power code of the current frame. The code is obtained by
      calculating a root mean square (RMS) of "Layer a" and encoding
      this RMS using G.711 u-law [ITU-T-G.711]. Denoting the encoded
      RMS as R, then PW1 is obtained by PW1 = ((~R)>>2) & 0x1F, where
      "~", ">>", "&" are one's complement arithmetic, right SHIFT, and
      bitwise AND operators, respectively.

3.3.1.2. PLC Information Field

```
 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
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|C|R2 |V|   K   |U|    P1     |U|    P2     |      PW2          |
|2|   |2|       |1|           |2|           |                   |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|     R3        |
|              |
+-+-+-+-+-+-+-+-+
```
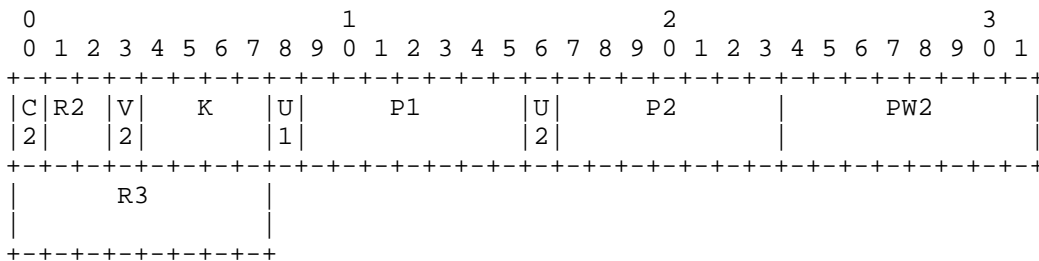
                  Figure 5: PLC Information Field (PC)

   Check bit #2 (C2): 1 bit

      Validity flag of V2, K, U1, P1, U2, P2, and PW2. If the flag is
      "1", it means that all these parameters are valid, and "0" means
      that the parameters should be ignored. If any of these parameters
      is invalid, this bit should be set to "0". Similarly to C1, this
      flag should be set to "0" in case of upward transcoding from G.711
      (see Section 4).

   Reserved bit #2 (R2): 2 bits

      These bits should be ignored. The default of these bits are 0.

   VAD flag #2 (V2): 1 bit

      Voice activity detection flag of the current frame, designed to be
      used for packet-loss concealment. This might not be the same as
      V1 in the mixing information, and might not be synchronous to the
      marker bit in the RTP header. DTX judgment based this flag is not
      recommended.

   Frame indicator (K): 4 bits

      This value indicates the frame offset of U2, P2, and PW2. Since
      it is a better idea to carry the speech feature parameters as PLC
      information in a different frame to maintain the speech quality,
      this frame offset value gives with which frame the parameters are
      to be associated. The value ranges between "0" and "15". If the
      current frame number is N, for example, the value K indicates that
      U2, P2, and PW2 are associated with the frame of N-K. The frame
      indicator is equal to the difference in the RTP sequence number
      when one UEMCLIP frame is contained in a single RTP packet.

   V/UV flag #1 (U1): 1 bit

      Voiced/Unvoiced signal indicator of the current frame. This flag
      being "0" indicates that the frame is a voiced signal segment, and
      "1" indicates that it is an unvoiced signal segment.

   Pitch lag #1 (P1): 7 bits

      Pitch code of the current frame. The actual pitch lag is
      calculated as P1+20 samples in 8-kHz sampling rate. Pitch lag
      must be 20 <= pitch length <= 120. Codes ranging between "0x65"
      and "0x7F" are not used. To obtain the pitch lag, any pitch
      estimation method can be used, such as the one used in G.711
      Appendix I [ITU-T-G.711Appendix1].

V/UV flag #2 (U2): 1 bit

   Voiced/Unvoiced signal indicator of the offset frame. This flag
   being "0" indicates that the frame is a voiced signal segment, and
   "1" indicates that it is an unvoiced signal segment. The offset
   value is defined as K.

Pitch lag #2 (P2): 7 bits

   Pitch code of the offset frame. The offset value is defined as K.
   The calculation method is identical to "P1", except that it is
   based on the signal of offset frame.

Power #2 (PW2): 8 bits

   Signal power code of the offset frame. The offset value is
   defined as K.

Reserved bits #3 (R3): 8 bits

   These bits should be ignored. The default of all bits are "0".

3.3.2. Sub-Layer

   Sub-layer (SL) is a sub-header followed by layer bitstreams, as shown
   in Figure 6. The sub-header indicates the layer location and the
   number of bytes.

```
    0                   1                   2
    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   . . .
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+//-+-+-+
   |CI |FI |QI |R4 |        SB         |              LD     ...  |
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+//-+-+-+
```

                   Figure 6: Sub-Layer Format (SL)
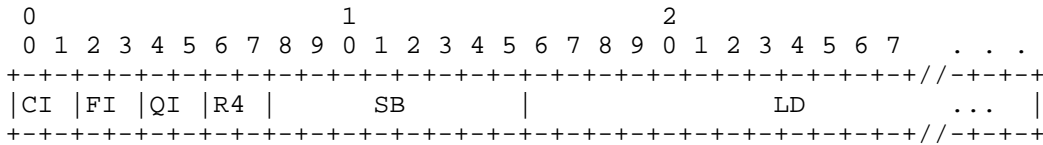
Channel index (CI): 2 bits

   Indicates the channel number. For all modes given in Table 2,
   this should be "0". The detail is given in Table 3.

Frequency index (FI): 2 bits

   Indicates the frequency number. "0" means that the layer is in the
   base frequency band, higher number means that the layer is in
   respective frequency band. The detail is given in Table 3.

Quality index (QI): 2 bits

   Indicates the quality layer number. "0" means that the layer is in
   the base layer, and higher number means that the layer is in
   respective quality layer. The detail is given in Table 3.

Reserved #4 (R4): 2 bits

   Not used (reserved). The default value is "0".

Sub-layer Size (SB): 8 bits

   Indicates the byte size of the following sub-layer data.

Layer Data (LD): SB*8 bits

   The actual sub-layer data.

For all the layers shown in Table 1, the layer indices are shown in
Table 3.

```
+-------+----+----+----+
| Layer | CI | FI | QI |
+-------+----+----+----+
|   a   | 0  | 0  | 0  |
|       |    |    |    |
|   b   | 0  | 0  | 1  |
|       |    |    |    |
|   c   | 0  | 1  | 0  |
+-------+----+----+----+
```

Table 3: Layer Indices

4. Transcoding between UEMCLIP and G.711

   As given in Section 2, the u-law-encoded G.711 bitstream (Layer a) is
   the core layer of a UEMCLIP bitstream, and is always embedded. This
   means that media transcoding from the UEMCLIP bitstream to G.711 does
   not have to undergo decoding and re-encoding procedures, but simple
   extraction would suffice. However, this does not apply for the
   reverse procedure, i.e., transcoding from G.711 to UEMCLIP, because
   the auxiliary information in the main header (MH) must be assigned
   separately. It should be noted that this media transcoding is useful
   for a Media Translator (Topo-Media-Translator) or a Point-to-
   Multipoint Using RTCP Terminating MCU (Topo-RTCP-terminating-MCU) in
   [RFC5117], and all the requirements apply. This means that a
   transcoding device of this sort MUST rewrite RTCP packets, together
   with the RTP media packets.

The transcoding from UEMCLIP to u-law G.711 can be done easily by
finding an appropriate sub-layer. Within a frame, the transcoder
should look for a sub-layer with a layer index of "0x00", and
subsequent LD that has a size of SB*8 bits (UEMCLIP has a 20-ms frame
thus, SB=160) are the actual G.711 bitstream data. It should be
noted that the transcoder should not always expect the core layer to
be located right after the main header.

On the other hand, the transcoding from G.711 to UEMCLIP is not
entirely straightforward. Since there are no means to generate
enhancement sub-layers, a G.711 bitstream can only be converted to
UEMCLIP Mode 0 bitstream. If the original G.711 bitstream is encoded
in A-law, it should first be converted to u-law to become the core
layer. Because a UEMCLIP frame size is 20 ms, a u-law-encoded G.711
bitstream MUST be a 160-sample chunk to become a core layer. For the
main header contents, when the UEMCLIP encoder is not available, it
should follow these guidelines:

o The check bits for mixing and PLC (C1 and C2) are set to 0.

o The reserved bits (R1 to R3) in MH are set to respective default
   values.

For the core layer (i.e., u-law G.711 bitstream), it should have the
following sub-layer header:

o All CI, FI, QI, and R4 MUST be 0.

o Sub-layer size (SB) MUST be 160 for a 20-ms frame.

5. Congestion Control Considerations

   The general congestion control considerations for transporting RTP
   data also apply to UEMCLIP over RTP [RFC3550] as well as any
   applicable RTP profile like Audio-Visual Profile (AVP) [RFC3551].

   The bandwidth of a UEMCLIP bitstream can be reduced by changing to
   lower-bit-rate modes. The embedded layer structure of UEMCLIP may
   help to control congestion, when dynamic mode changing (see
   Section 6.2.1) is available, and the range of modes is obtained by
   offer-answer negotiation as given in Section 6.3. It should be noted
   that this involves proper RTCP handling when the bit-rate is modified
   in an RTP translator or a mixer [RFC3550].

Packing more frames in each RTP payload can reduce the number of
packets sent, and hence the overhead from IP/UDP/RTP headers, at the
expense of increased delay and reduced error robustness against
packet losses. It should be treated with care because increased
delay means reduced quality.

6. Payload Format Parameters

6.1. Media Type Registration

This registration is done using the template defined in [RFC4288] and
following [RFC4855].

Media type name: audio

Media subtype name: UEMCLIP

Required parameters:

   Rate: Defines the sampling rate, and it MUST be either 8000 or
      16000. See Section 6.2.1 "Mode specification" of RFC 5686
      (this RFC) for details.

Optional parameters:

   ptime: See RFC 4566 [RFC4566].

   maxptime: See RFC 4566 [RFC4566].

   mode: Indicates the range of dynamically changeable modes during
      a session. Possible values are a comma-separated list of modes
      from the supported mode set: 0, 1, 3, and 4. If only one mode
      is specified, it means that the mode must not be changed during
      the session. When not specified, the mode transmission
      defaults to a singular mode as specified in Table 4. See
      Section 6.2.1 "Mode specification" of RFC 5686 (this RFC) for
      details.

Encoding considerations: This media type is framed and contains
   binary data. See Section 4.8 of RFC 4288.

Security considerations: See Section 7 "Security Considerations" of
   RFC 5686 (this RFC).

Interoperability considerations: This media may be readily
   transcoded to u-law-encoded ITU-T G.711. See Section 4
   "Transcoding between UEMCLIP and G.711" of RFC 5686 (this RFC).

   Published specification: RFC 5686 (this RFC)

   Applications that use this media type: Audio and video streaming and
      conferencing tools.

   Additional information: None

   Intended usage: COMMON

   Restrictions on usage: This media type depends on RTP framing, and
      hence is only defined for transfer via RTP.

   Person & email address to contact for further information:
      Yusuke Hiwasaki <hiwasaki.yusuke@lab.ntt.co.jp>

   Author: Yusuke Hiwasaki

   Change Controller: IETF Audio/Video Transport Working Group
      delegated from the IESG

6.2. Mapping to SDP Parameters

   The media types audio/UEMCLIP are mapped to fields in the Session
   Description Protocol (SDP) [RFC4566] as follows:

   Media name: The "m=" line of SDP MUST be audio.

   Encoding name: Registered media subtype name should be used for the
      "a=rtpmap" line.

   Sampling Frequency: Depending on the mode, clock rate (sampling
      frequency) specified in "a=rtpmap" MUST be selected from the ones
      defined in Table 2. See Section 6.2.1 for details.

   Encoding parameters: Since this is an audio stream, the encoding
      parameters indicate the number of audio channels, and this SHOULD
      default to "1", as selected from the ones defined in Table 2.
      This is OPTIONAL.

   Packet time: A frame length of any UEMCLIP is 20 ms, thus the
      argument of "a=ptime" SHOULD be a multiple of "20". When not
      listed in SDP, it should also default to the minimum size: "20".

   UMECLIP specific: Any description specific to UEMCLIP is defined in
      the Format Specification Parameters ("a=fmtp"). Each parameter
      MUST be separated with ";", and if any attribute (value) exists,
      it MUST be defined with "=". For compatibility reasons, any
      application/terminal MUST ignore any parameters that it does not

understand. This is to ensure the upper-compatibility with
parameters added in future enhancements. The mode specification
should be made here (see Section 6.2.1).

6.2.1. Mode Specification

Since UEMCLIP codec can operate in number of modes (bit-rates), it is
desirable to specify the range of modes at which an encoder or a
decoder can operate. When exchanging SDP messages, an offerer should
specify all possible combinations of mode numbers as arguments to
"mode=" in "a=fmtp" line, delimited by commas ",". In case of
specifying multiple modes, those SHOULD appear in the descending
priority order.

Although UEMCLIP decoders SHOULD accept bitstreams in any modes, an
implementation may fail to adapt to the dynamic mode changes during a
session. For this reason, an application may choose to operate
either with one fixed mode or with multiple modes that can be
dynamically changed. If the mode is to be fixed and changes are not
allowed, this can be indicated by specifying a single mode per
payload type.

The mode numbers that can be specified in a payload type as arguments
to "mode" are restricted by a combination of a clock rate and a
number of audio channels. This is because SDP binds a payload type
to a combination of a sampling frequency and a number of audio
channels. Table 4 gives selectable mode numbers that are attributed
with clock rates. When mode specifications are not given at all, a
payload type MUST default to a single mode using the default value
specified in this table.

Clock rate	Channels	Selectable modes	Default mode
8000	1	0,3	0
16000	1	0,1,3,4	1

Table 4: Default Modes

It should be noted that a mode attributed with a larger sampling
frequency (Fs) is not used in conjunction with smaller clock rates
specified in "a=rtpmap". This means that Modes 0 and 3 can be
specified in a payload type having a clock rate of both 8000 and
16000 in "a=rtpmap", but Modes 1 and 4 cannot be specified with one
having a clock rate of 8000.

6.3. Offer-Answer Model Considerations

6.3.1. Offer-Answer Guidelines

   The procedures related to exchanging SDP messages MUST follow
   [RFC3264]. The following is a detailed list on the semantics of
   using the UEMCLIP payload format in an offer-answer exchange.

   o An offerer SHOULD offer every possible combination of UEMCLIP
      payload type it can handle, i.e., sampling frequency, channel
      number, and fmtp parameters, in a preferred order. When the
      transmission bandwidth is restricted, it MUST be offered in
      accordance to the restriction.

   o When multiple UEMCLIP payload types are offered, it is RECOMMENDED
      that the answerer select a single UEMCLIP payload type and answer
      it back.

   o In a UEMCLIP payload type, an answerer MUST answer back suitable
      mode number(s) as a subset of what has been offered. This means
      that there is a symmetry assumption on sent and received streams,
      and the offerer MUST NOT send in modes that it does not offer.

   o In an offering/answering SDP, any fmtp parameters that are not
      known MUST be ignored. If any unknown/undefined parameters should
      be offered, an answerer MUST delete the entry from the answer
      message.

   o A receiver of an SDP message MUST only use specified payload types
      and modes. When a mode specification is missing, i.e., a mode is
      not specified at all, the session MUST default to one single mode
      without mode changes during a session. For this case, the default
      mode values, as shown in Table 4, MUST be used based on the
      sampling frequency and number of channels. This table must be
      looked up only when there are no mode specifications; thus, the
      offerer/answerer MUST NOT assume that the default modes are always
      available when it is not in the specified list of modes.

   o When an offered condition does not fit an answerer's capabilities,
      it naturally MUST NOT answer any of the conditions, and the
      session MAY proceed to re-INVITE, if possible. If a condition
      (mode) is decided upon, an offerer and an answerer MUST transmit
      on this condition.

6.3.2. Examples

   When an offerer indicates that he/she wishes to dynamically switch
   between modes (0,1,3, and 4) during a session, an example of an
   offered SDP could be:

     v=0
     o=john 51050101 51050101 IN IP4 offhost.example.com
     s=-
     c=IN IP4 offhost.example.com
     t=0 0
     m=audio 5004 RTP/AVP 96
     a=rtpmap:96 UEMCLIP/16000/1
     a=fmtp:96 mode=4,1,3,0

   It should be noted that the listed modes appears in the offerer's
   preference.

   When an answerer can only operate in Modes 1 and 0 but can
   dynamically switch between those modes during a session, an answerer
   MUST delete the entries of Mode 3 and 4, and answer back as:

     v=0
     o=lena 549947322 549947322 IN IP4 anshost.example.org
     s=-
     c=IN IP4 anshost.example.org
     t=0 0
     m=audio 5004 RTP/AVP 96
     a=rtpmap:96 UEMCLIP/16000/1
     a=fmtp:96 mode=1,0

   As a result, both would start communicating in either Mode 1 or 0,
   and can dynamically switch between those modes during the session.

   On the other hand, when the answerer is capable of communicating
   either in Modes 1 or 0, and cannot switch between modes during a
   session, an example of such answer is as follows:

     v=0
     o=lena 549947322 549947322 IN IP4 anshost.example.org
     s=-
     c=IN IP4 anshost.example.org
     t=0 0
     m=audio 5004 RTP/AVP 96
     a=rtpmap:96 UEMCLIP/16000/1
     a=fmtp:96 mode=1

As a result, both will start communicating in Mode 1. It should be
noted that mode change during this session is not allowed because the
answerer responded with a single mode, and answerer selected Mode 1
above Mode 0 according to the offered order.

If an offerer does not want a mode change during a session but is
capable of receiving either Modes 4 or 1 bitstreams, the SDP should
somewhat look like:

```
v=0
o=john 51050101 51050101 IN IP4 offhost.example.com
s=-
c=IN IP4 offhost.example.com
t=0 0
m=audio 5004 RTP/AVP 96 97
a=rtpmap:96 UEMCLIP/16000/1
a=fmtp:96 mode=4
a=rtpmap:97 UEMCLIP/16000/1
a=fmtp:97 mode=1
```

and if the answerer prefers to communicate in Mode 1, an answer would
be:

```
v=0
o=lena 549947322 549947322 IN IP4 anshost.example.org
s=-
c=IN IP4 anshost.example.org
t=0 0
m=audio 5004 RTP/AVP 97
a=rtpmap:97 UEMCLIP/16000/1
a=fmtp:97 mode=1
```

Please note that it is RECOMMENDED to select a single UEMCLIP payload
type for answers.

The "ptime" attribute is used to denote the desired packetization
interval. When not specified, it SHOULD default to 20. Since
UEMCLIP uses 20-ms frames, ptime values of multiples of 20 imply
multiple frames per packet. In the example below, the ptime is set
to 60, and this means that offerer wants to receive 3 frames in each
packet.

```
v=0
o=kosuke 2890844730 2890844730 IN IP4 anotherhost.example.com
s=-
c=IN IP4 anotherhost.example.com
t=0 0
m=audio 5004 RTP/AVP 96
a=ptime:60
a=rtpmap:96 UEMCLIP/16000/1
```

When mode specification is not present, it should default to a fixed
mode, and in this case, Mode 1 (see Section 6.2.1).

7. Security Considerations

   RTP packets using the payload format defined in this specification
   are subject to the security considerations discussed in the RTP
   specification [RFC3550] and any appropriate profiles. This implies
   that confidentiality of the media streams is achieved by encryption
   unless the applicable profile specifies other means.

   A potential denial-of-service threat exists for data encoding using
   compression techniques that have non-uniform receiver-end
   computational load. The attacker can inject pathological datagrams
   into the stream that are complex to decode and cause the receiver
   output to become overloaded. However, the UEMCLIP covered in this
   document do not exhibit any significant non-uniformity.

   Another potential threat is memory attacks by illegal layer indices
   or byte numbers. The implementor of the decoder should always be
   aware that the indicated numbers may be corrupted and not point to
   the right sub-layer, and they may force reading beyond the bitstream
   boundaries. It is advised that a decoder implementation reject
   layers of such indices.

8. IANA Considerations

   One new media subtype (audio/UEMCLIP) has been registered by IANA.
   For details, see Section 6.1.

9. References

9.1. Normative References

   [ITU-T-G.711]
             International Telecommunications Union, "Pulse code
             modulation (PCM) of voice frequencies", ITU-
             T Recommendation G.711, November 1988.

   [RFC2119] Bradner, S., "Key words for use in RFCs to Indicate
              Requirement Levels", BCP 14, RFC 2119, March 1997.

   [RFC3264] Rosenberg, J. and H. Schulzrinne, "An Offer/Answer Model
              with Session Description Protocol (SDP)", RFC 3264,
              June 2002.

   [RFC3550] Schulzrinne, H., Casner, S., Frederick, R., and V.
              Jacobson, "RTP: A Transport Protocol for Real-Time
              Applications", STD 64, RFC 3550, July 2003.

   [RFC3551] Schulzrinne, H. and S. Casner, "RTP Profile for Audio and
              Video Conferences with Minimal Control", STD 65, RFC 3551,
              July 2003.

   [RFC4288] Freed, N. and J. Klensin, "Media Type Specifications and
              Registration Procedures", BCP 13, RFC 4288, December 2005.

   [RFC4566] Handley, M., Jacobson, V., and C. Perkins, "SDP: Session
              Description Protocol", RFC 4566, July 2006.

   [RFC4855] Casner, S., "Media Type Registration of RTP Payload
              Formats", RFC 4855, February 2007.

   [RFC4856] Casner, S., "Media Type Registration of Payload Formats in
              the RTP Profile for Audio and Video Conferences",
              RFC 4856, February 2007.

   [RFC5117] Westerlund, M. and S. Wenger, "RTP Topologies", RFC 5117,
              January 2008.

9.2. Informative References

   [ITU-T-G.711Appendix1]
              International Telecommunications Union, "Pulse code
              modulation (PCM) of voice frequencies, Appendix I: A high
              quality low-complexity algorithm for packet loss
              concealment with G.711", ITU-T Recommendation G.711
              Appendix I, September 1999.

Authors' Addresses

    Yusuke Hiwasaki
    NTT Corporation
    3-9-11 Midori-cho,
    Musashino-shi
    Tokyo 180-8585
    Japan

    Phone: +81(422)59-4815
    EMail: hiwasaki.yusuke@lab.ntt.co.jp

    Hitoshi Ohmuro
    NTT Corporation
    3-9-11 Midori-cho,
    Musashino-shi
    Tokyo 180-8585
    Japan

    Phone: +81(422)59-2151
    EMail: ohmuro.hitoshi@lab.ntt.co.jp