Internet Engineering Task Force
Audio Video Transport WG
INTERNET-DRAFT
July 14, 2000
Expires: January 14, 2001

Tim Fingscheidt, Siemens AG
Bernhard Wimmer, Siemens AG
Germany

RTP Payload Format for AMR

<draft-fingscheidt-avt-rtp-amr-00.txt>

Status of this memo

This document is an individual submission to the IETF. Comments should be directed to the authors.

Abstract

This document proposes a real-time transport protocol (RTP) [1] payload format for AMR speech encoded [2] signals. It supports all 8 modes of the AMR speech codec and is as well prepared for future extensions, such as AMR wideband. Mode adaptation and discontinuous transmission (DTX) are supported as well.

The proposed payload format allows large flexibility with a minimum of bitrate overhead. One or multiple speech frames can be transmitted in a single packet. Redundant transmission of previously transmitted frames (or parts thereof) is possible as well as parity code transmission. With one speech frame per packet the additional parity code transmission allows reconstruction of N previous lost speech frames when N consecutive correct packets are buffered in the receiver. This means a very high robustness while the receiver buffer size can be chosen according to the application.

For implementation of this draft, please consider also the requirements of [12].

1. Conventions used

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 [11].

2. Introduction

The European Telecommunications Standards Institute (ETSI) as well as the Third Generation Partnership Project (3GPP) standardized the adaptive multi-rate (AMR) speech codec. In third generation systems the AMR codec will be mandatory. Three of the AMR modes are earlier standards like the 6.7 kbps mode (PDC-EFR [3]), the 7.4 kbps mode (IS-641 codec in TDMA [4]), and the 12.2 kbps mode (GSM-EFR [5]).

The AMR codec comprises 8 modes with different bit rates ranging from 4.75 to 12.2 kbps. In systems with a fixed gross bit rate like e.g. GSM, this allows assigning different amounts of error protection in order to preserve high speech quality over a wide range of channel qualities. The sampling frequency is 8 kHz, speech frames are processed in 20 ms frames. The AMR modes are closely related to each other and use the same coding framework.

AMR implementations must support all 8 speech coding modes, and mode switching can occur to any mode at any speech frame boundary. The mode information must therefore be transmitted together with the speech encoded bits to indicate the mode. Furthermore, the decoder may give an indication to the encoder of what mode it prefers to receive. This is called a codec mode request (CMR) and is useful to adjust the ratio of speech coder bits to error protection bits in order to ensure a certain speech quality.

Along with the AMR codec, voice activity detection (VAD) and comfort noise generation (CNG) have been standardized. This allows a reduction of the number of transmitted bits in silence periods.
The three earlier codec standards [3-5] however have different DTX/VAD/CNG schemes if they are not used in the AMR framework. For Interoperability reasons the proposed payload format supports also these CNG formats.

To address the transmission over networks with high packet loss rates extra redundancy is built into the RTP payload format for AMR This is done in a very flexible manner by the optional transmission of parity bit blocks generated from previously transmitted AMR encoded frames. Dependent on how many previous frames are covered by this parity bit computation, a certain number of consecutive past lost frames can be reconstructed at the receiver. Since this may require buffering, the AMR payload format allows flexible tradeoff between robustness, bit rate, and receiver delay.

The speech encoded bits have different perceptual sensitivity to bit errors. Accordingly, unequal error protection (UEP) is employed in cellular systems. A frame is considered as lost or damaged if errors are detected in the most sensitive bits. Unequal error detection (UED) can also be employed on RTP if e.g. UDP lite is used as transport layer protocol (UDP lite [6] is work in progress). The payload then has to be ordered in sensitivity order. The sensitivity order for the AMR encoded bits are defined in [7]. The different sensitivity can also be exploited by a parity check covering only the most sensitive bits, as is proposed as an option for the AMR payload format.

To improve quality in circuit-switched GSM networks connected to IP networks also frames disturbed on the wireless GSM link should be transmitted to the decoder in the IP network. Consequently, such frames must be accompanied by a frame quality information in the IP network.

This proposal of an RTP payload format for AMR is the third in a series of internet drafts (works in progress) related to this topic. In [8] the transmission of multiple speech frames in a single RTP packet is supported. The advantage of [9] as compared to [8] is mainly the possibility to transmit redundant speech frames (or parts thereof).

The present proposal incorporates the abilities of [8,9] with the addition that there is an option for reconstruction of a larger number of past lost frames. For the purpose of clarity and simpler comparison, in the sequel we will follow the structure and the notation of [9] as far as possible.

3. Requirements

The AMR payload format for RTP was designed to meet the following requirements:

- o Different levels of robustness must be supported:
  - no redundancy at all
  - past frames (partly) repeated
  - parity bits generated over several past frames to yield extreme robustness capable of handling very high packet loss rates with no or small speech quality degradation.

- o Fast, frame-wise AMR mode adaptation must be supported. This means that it must be possible to send codec mode requests (CMRs) back from the receiving side to the transmitting side with information on the preferred mode. Slower AMR mode adaptation may also be accomplished with external signaling.

- o Discontinuous transmission (DTX) and comfort noise generation (CNG) as specified in AMR must be supported.

4. RTP Payload Format Specification

This RTP payload format is designed to be flexible, ranging from very low overhead (minimal) to an extended format with room for future AMR extensions, e.g. wide band modes, and the possibility to send extra redundancy information and several speech frames in one RTP payload packet.

Each RTP payload consists of an
- RTP payload header followed by the
- RTP payload data.

The RTP payload data is generated by the interleaving of one or several RTP payload frames, see section 4.4. An RTP payload frame may be generated from
- AMR frames or
- redundancy frames.

Each RTP payload frame must not be octet-aligned, however the RTP payload shall be octet-aligned. If the last octet of an RTP payload covers unused bits, these bits shall be set to zero.

4.1. The RTP Payload Header

The payload header has dynamic length, 3 or 8 bits. The bits in the Header are specified as follows:

Q (1 bit): The payload quality bit indicates, if not set, that the Payload is severely damaged and the receiver should set the RX_TYPE, see [10], to SPEECH_BAD or SID_BAD depending on the frame type (FT).

I (1 bit): If I=1, it indicates the existence LEN/DEPTH indicator bit (L) in each RTP payload frame. If I=0 the LEN/DEPTH indicator do not exist.

R (1 bit): Indicates if the codec mode request (CMR) is sent or not.

CMR (5 bits): OPTIONAL field, depending on the R bit. Requested codec mode for the other communication direction. The interpretation is equal to the FT field, see Table 1.

```
 0
 0 1 2
+-+-+-+
|Q|I|R|
+-+-+-+
```

Figure 1: RTP payload header, R=0

```
 0
 0 1 2 3 4 5 6 7
+-+-+-+-+-+-+-+-+
|Q|I|R|   CMR   |
+-+-+-+-+-+-+-+-+
```

Figure 2: RTP payload header, R=1

4.2. RTP Payload AMR Frame

The RTP payload AMR frame is designed for covering AMR encoded speech data and is generated by
- AMR frame header that is followed by the
- AMR frame payload.

The AMR frame must not be octet-aligned.

4.2.1. AMR Frame Header Format

Each AMR frame header includes several specified fields as follows:

F (1 bit): Indicates if this frame is followed by further frames. F=1 further frames follow, F=0 last frame.

L (1 bit): (OPTIONAL) If the RTP payload header bit I=1 this field exists. If I=0 this field is not existing. If set to L=1 the AMR frame header includes the LEN field. If L=0 no LEN field exists in this AMR frame header.

FT (5 bits): Frame type indicator, indicating the AMR speech coding mode or comfort noise (CN) mode. The mapping of existing AMR modes is given in Table 1. This implies that the number of bits of the AMR frame payload can be derived from Table 1. If FT=15 (No transmission) L for both AMR and redundancy frames SHOULD be set to 0.

LEN (7 bits): OPTIONAL field, exists if the AMR header bit L is set, L=1. LEN specifies the number of octets in the current AMR frame payload. The following situations may occur and shall be treated as follows:

- If LEN*8 <= number of speech bits indicated by FT, as shown in Table. 1,
  the number of bits of the AMR frame payload shall be derived by
  8*LEN and not by the FT field. This implies that the encoded AMR
  data was shortend to 8*LEN.
- otherwise the LEN field SHOULD be ignored.

```
 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
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|F|L|   FT    |     LEN     |                                   |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+                                 +
|                                                               |
+                                                               +
```

```
   /                    AMR frame payload                          /
   /                                                               /
   +                                                 +-+-+-+-+-+-+-+-+
   |                                                 |
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Figure 3: AMR frame format, I=1 and L=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
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|F|L|   FT    |                                                 |
+-+-+-+-+-+-+-+                                                 +
|                                                               |
+                                                               +
/                    AMR frame payload                          /
/                                                               /
+                                                +-+-+-+-+-+-+-+-+
|                                                |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Figure 4: AMR frame format, I=1 and L=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
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|F|   FT    |                                                   |
+-+-+-+-+-+-+                                                   +
|                                                               |
+                                                               +
/                    AMR frame payload                          /
+                                            +-+-+-+-+-+-+-+-+-+-+
|                                            |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Figure 5: AMR frame format, I=0

4.2.2. AMR Frame Payload Format

The AMR speech encoder produces AMR speech frames, as defined by [2]. The currently defined AMR speech frame types can be found in Table 1.

Index	Mode	speech bits
0	AMR 4.75	95
1	AMR 5.15	103
2	AMR 5.9	118
3	AMR 6.7	134
4	AMR 7.4	148
5	AMR 7.95	159
6	AMR 10.2	204
7	AMR 12.2	244
8	AMR CNG	39
9	GSM EFR CNG	43

10	IS-641 CNG	38
11	PDC-EFR CNG	37
12 - 14	For future use	-
15	No transmission	0
16 - 31	For future use	-

Table 1: AMR speech frame types (taken from [9])

The bit order of frame type 0 - 11 is given in [7]. Frame type 15, no transmission, is needed to indicate not transmitted frames or lost frames, e.g. when multiple frames are sent in each payload and comfort noise starts. A frame type sequence in a payload with 8 frames, AMR mode 7, and CNG starts in the fifth frame, could look like: {7,7,7,7,8,15,15,8}. The AMR DTX (also called "source controlled rate operation", SCR) is described in [10]. Another reason for the no transmission frame type is a possible need to send an urgent codec mode request in a silence period with comfort noise.

Before the AMR encoded speech frames are copied to the AMR frame payload the speech bits shall be ordered to the descending bit-error sensitivity. This re-ordering process is defined in [7].

After this re-ordering process the AMR encoded speech frame is copied to the AMR frame payload, according to the particular setting of the AMR frame header, e.g. copying of the first 8*LEN bits, see section 4.2.1.

4.3. RTP Payload - Redundancy Frame

The RTP payload redundancy frame is designed for covering redundancy data for error-correction of lost AMR frames. The redundancy frame is generated by
- redundancy frame header that is followed by the
- redundancy frame payload.

The redundancy frame must not be octet-aligned.

4.3.1. Redundancy Frame Header Format

Each redundancy frame header includes several specified fields as follows:

F (1 bit): Indicates if this frame is followed by further frames. F=1 further frames follow, F=0 last frame.

L (1 bit): (OPTIONAL) If the RTP payload header bit I=1 this field exists. If I=0 this field is not existing. If set to L=1 the redundancy frame header includes the LEN field. If L=0 no R_LEN field exists in this redundancy frame header.

R_FT (5 bits): This field indicates the FT-fields of the past DEPTH AMR frame headers by the following coding rule.

```
   R_FT(n) = FT(n-1) EXOR ... EXOR FT(n-DEPTH(n))     (Eq. 1)
   whereby
   n        is set to the current AMR frame number.
   FT(n)    is defined as the AMR frame header field FT of
```

```
          frame n.
R_FT(n)   denotes the redundancy frame header field R_FT of
          frame n.
EXOR      is defined as the bit-wise exclusive OR operation.
DEPTH(n)  denotes the redundancy frame header field DEPTH of
          frame n.
```

R_LEN (7 bits): OPTIONAL field, exists if the redundancy header bit L is set, L=1. R_LEN specifies the number of octets in the current redundancy frame payload. Depending on R_LEN several different operational modes are used that will be described in section 4.3.2. R_LEN may be changed from redundancy frame to redundancy frame. If L=0 or/and I=0, R_LEN(n) is set to FT(n), whereby n denotes the current AMR frame number.

DEPTH (4 bits): OPTIONAL field, exists if the redundancy header bit L is set, L=1. DEPTH specifies the number of previous AMR frame payload pakets that are used for the generation of the redundancy frame payload. The detailed description can be found in section 4.3.2. DEPTH = 0 is currently unused and may be used for future extension. If L=0 or/and I=0 then DEPTH is set to the default value 15.

```
 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
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|F|L|  R_FT   |     R_LEN    | DEPTH |                          |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+                         +
|                                                               |
+                                                               +
/                     redundancy frame payload                  /
/                                                               /
+                                                 +-+-+-+-+-+-+-+
|                                                 |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Figure 6: Redundancy frame format, I=1 and L=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
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|F|L|   R_FT  |                                                 |
+-+-+-+-+-+-+-+                                                 +
|                                                               |
+                                                               +
/                     redundancy frame payload                  /
/                                                               /
+                                                 +-+-+-+-+-+-+-+
|                                                 |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Figure 7: Redundancy frame format, I=1 and L=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
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

```
|F|   R_FT  |                                                     |
+-+-+-+-+-+-+                                                     +
|                                                                 |
+                                                                 +
/                    redundancy frame payload                     /
+                                             +-+-+-+-+-+-+-+-+-+-+
|                                             |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Figure 8: Redundancy frame format, I=0

4.3.2. Redundancy Frame Payload Format

The generation of the redundancy payload is based on parity bit calculation of one or several previous AMR frame payload pakets. This number of AMR frames is determined by the redundancy frame header field DEPTH.

The general rules for generating of the parity bits can be found in section 4.3.3.

The value of R_LEN can in principle be changed during transmission. Let's assume R_LEN changes from R_LEN1 to R_LEN2, with DEPTH being constant. In that case for a number of DEPTH AMR frame packets only min(R_LEN1,R_LEN2) AMR frame payload bits can be reconstructed. Although adaptation of R_LEN for redundancy frames works seamlessly, it is RECOMMENDED not to perform such an adaptation on a frame-by-frame basis.

The value of DEPTH can also be adapted during transmission. Let's assume DEPTH changes from DEPTH1 to DEPTH2. It is RECOMMENDED to choose a maximum value of DEPTH dependent on the application (e.g. streaming services: large DEPTH, VoIP: low DEPTH) and to adapt it only on a long term basis, since reconstruction capabilities are reduced in transition regions for a number of min(DEPTH1,DEPTH2) AMR frames.

4.3.3. Encoding Rules for the Parity Bits

This section describes the encoding rules for the parity bits.

```
Notation:
n        : number of the current AMR frame; n is increased for each
           sent AMR frame packet. n denotes also the current
           redundancy frame number.
o        : number of AMR frame that covers less AMR frame payload
           bits than required by current redundancy frame header
           field R_LEN(n) > LEN(o).
g(n,m)   : bit m in the AMR frame payload of frame n
p(n,m)   : bit m in the redundancy frame payload of frame n
XOR      : exclusive OR operation
R_LEN(n) : denotes the R_LEN field of the redundancy frame header of
           frame n
```

The parity bits SHALL be calculated by the following equation:

```
  p(n,m) = g(n-1,m) EXOR ... EXOR g(n-DEPTH+1, m) EXOR g(n-DEPTH, m)
                                                               (eq.2)
  for m = 0 ... R_LEN(n)-1;
```

Eq. 2 requires that all LEN(i) with i = (1, ... , DEPTH) of the AMR frames are at least as large as R_LEN(n). In the event that this is not valid the missing AMR frame payload bits SHALL be virtually generated by the following rule.

```
if (o = n-DEPTH)

   g(o, LEN(o)+i) = 0, for i=0...(R_LEN(n)-LEN(o)-1);

else

   if (R_LEN(n)-LEN(o) <= LEN(o-1))

      g(o, LEN(o)+i) = g(o-1, i),  for i=0...(R_LEN(n)-LEN(o)-1);

   else  {

      g(o, LEN(o)+i) = g(o-1, i),  for i = 0 ... (LEN(o-1)-1);
      g(o, LEN(o)+LEN(o-1)+i) = 0,
         for i = 0 ... (R_LEN(n)-LEN(o)-LEN(o-1)-1);
   }
```

This rule implies that virtuell data SHALL be copied from the most sensitive bits of the previous AMR frame payload of the AMR frame o. However if the previous AMR frame number (o-1) is outside the window defined by the DEPTH parameter of the current redundancy frame the virtual data is set to 0. In the case that the AMR frame payload (o-1) contains less bits than required to achieve all virtual bits of AMR frame payload (o) then first all AMR frame payload bits of (o-1) SHALL be taken and then the missing virtual bits of AMR frame payload (o) SHALL be set to 0.

Example:

In this example, see Figure 9, it can be seen that the AMR frame payload contains not enough bits. Therefore the most sensitive bits of AMR frame payload (n-3) are virtually appended to AMR frame payload (n-2) until the desired length is reached.

```
time: n-3               n-2                n-1              n

+----------+       +-----------+       +----------+     +--------+
|          |- XOR -| g(n-2,m), |- XOR -|          |  =  |        |
| g(n-3,m) |- XOR -| fill with |- XOR -| g(n-1,m) |  =  | p(n,m) |
|          |- XOR -| g(n-3,m)  |- XOR -|          |  =  |        |
+----------+       +-----------+       +----------+     +--------+
```

Figure 9: Example of parity bit generation for p(n,m) with DEPTH=3 and the number of AMR frame payload bits in frame n-2 being smaller than 8*R_LEN(n).

4.3.4. Decoding of Redundancy Frame Payload

Decoding of these parity codes is intended in the following manner. Imagine one frame of AMR encoded bits and one parity bit block per frame. Every value of DEPTH >= 1 allows the reconstruction of a single lost frame among the last DEPTH frames. DEPTH = 2 allows the reconstruction of two consecutive lost frames, once two good frames are received. In general, a number of DEPTH buffered packets allows for the reconstruction of a number of DEPTH lost frames preceding them. The set of equations given by the XOR operations is solved at first for the last (!) lost frame (unknowns), using the DEPTH buffered frames as knowns. Then everything is solved for the last but first lost frame, taking into account the already reconstructed last lost frame's bits. And so forth.

Here the tremenduous strength of using parity codes instead of frame repetition becomes obvious: Especially for streaming applications a large value of DEPTH allows to reconstruct error bursts of the same large number of DEPTH consecutive frames.

4.3.5. Implications for DTX and the choice of DEPTH

For delay reasons it is not advisable to store a large number (DEPTH) of CNG frames in the receiver buffer before previous lost CNG AMR frames or AMR frame payload packets, containing speech data, can be reconstructed.

Thus the follwing rules SHALL apply:

- o Starting with the second AMR frame containing one/several CNG frames, DEPTH SHALL be set maximally to 1 for all consecutive redundancy frames containing CNG AMR frames.
- o In the first and the second AMR frame containing no CNG after a speech pause, DEPTH SHALL be set maximally to 1.

These rules allow optimal recovery of lost AMR frames in DTX operation, while keeping delay at a minimum.

4.4. Payload Block Sorting

In general a bit
error in a more sensitive bit is subjectively more annoying than in a less sensitive bit. To be able to protect the most sensitive bits in a AMR and redundancy frames with a forward error detection code, e.g. a CRC outside RTP, the full RTP payload data MUST be sorted in sensitivity order. The protection MAY then cover an appropriate number of octets from the beginning of the AMR and/or redundancy frames. How many octets depends on the channel and application. This can for example be accomplished by UDP lite [6] (work in progress). To maintain sensitivity ordering inside the AMR payload when more than one speech frame is transmitted in one packet reordering of the data is needed.

The reordering to maintain the sensitivity ordered AMR payload SHALL be performed on bit level. The AMR payload header SHALL still be placed unchanged in the beginning of the payload. Thereafter, the payload frames are sorted with one bit alternating from each payload frame.

```
+-------------+
| h(0)-h(H-1) |
+------------------------+
| f(0,0) _ f(0,F(0))     |
+----------------------------+
| f(1,0) _ f(1,F(1))         |
+----------------------------+
| f(2,0) _ f(2,F(2))   |
+----------------------+
\                          \
+-------------------------------+
| f(N-1,0) _ f(N-1,F(N-1))      |
+-------------------------------+
```

Figure 10: The payload header and N AMR/redundancy frames before sorting.

The sorting algorithm can be described in C-code.

```
b(m)     : bit m of RTP final payload
f(n,m)   : bit m in AMR/redundancy frame payload of frame n
F(n)     : number of bits in AMR/redundancy frame n, defined by FT
           or by LEN/R_LEN
h(m)     : bit m of RTP payload header
H        : number of RTP payload header bits, 3 or 8 bits
N        : number of AMR/redundancy frames in the RTP payload
S        : number of unused bits
```

Payload frames f(n,m) are ordered in consecutive order, where frame n=1 is preceding frame n=2.

The sorting algorithm is defined in C-style as:

```
for (i = 0; i < H; i++)
  b(i) = h(i);
max = max(F(0),..,F(N-1));
k = H;
for (i = 0; i < max; i++){
  for (j = 0; j < N; j++){
    if (i < F(j)){
      b(k++) = f(j,i);
    }
  }
}
S = 8 - k%8;
if (S < 8){
  for (i = 0; i < S; i++)
    b(k++) = 0;
```

```
   }
```

5. RTP header usage

The RTP header marker bit (M) is used to mark (M=1) the packages containing the first speech frame after CN. In all other packages the marker bit is set to 0 (M=0).

The time-stamp corresponds to the sampling time of the first sample encoded for the first encoded speech frame in the AMR frame. The timestamp unit is in samples, i.e. one AMR speech frame is 20 ms and sampling frequency is 8 kHz corresponds to 160 encoded speech samples per frame, i.e. the timestamp is increased by 160 for each AMR speech consecutive frame.

Due to DTX functionality each RTP packet SHALL contain the appropriate time-stamp of the first AMR frame, covered by the RTP payload. Each AMR frame containg CNG data or the first AMR frame containing speech data after CNG SHALL start with a new RTP packet. This is required to achieve the correct timing information.

Please consider also [12] for setting of particular parameters.

6. Examples

6.1. Simple example

In the simple example we just send one full (I=0) frame in each RTP packet, no codec mode request CMR is sent (R=0), the payload was not damaged at IP origin (Q=1). In this example we transmit one frame encoded with the 5.9 kbps mode (FT=2). The speech encoded bits are put into f(0) to f(117) in descending sensitivity order according to [7].

```
    |                          Bit no.                              |
Oct.|   0       1       2       3       4       5       6       7   |
----+-------+-------+-------+-------+-------+-------+-------+-------+
  0 |  Q=1  |  I=0  |  R=0  |  F=0  |   0   |   0   |   0   |   1   |
----+-------+-------+-------+-------+-------+-------+-------+-------+
  1 |   0   | f(0)  | f(1)  | f(2)  |  ...  |  ...  |  ...  |  ...  |
----+-------+-------+-------+-------+-------+-------+-------+-------+
 16 |  ...  |  ...  |  ...  |  ...  | f(115)| f(116)| f(117)|   0   |
----+-------+-------+-------+-------+-------+-------+-------+-------+
```

Figure 11: One frame per packet example.

6.2. Example with parity bits

In this example a AMR frame with 6.7 kbps mode (FT=3) is sent with one redundancy frame packet.

- The RTP payload header is set to Q=1, I=1, R=1 and CMR = 6. A mode request is sent(R=1), requesting the 10.2 kbps mode for the other link (CMR=6).

- The AMR frame header uses F=1, L=0 (this implies NO LEN field) and FT = 3. The AMR frame header is followed by the AMR frame payload, denoted by f(0) to f(133).

- The redundancy frame header is set to
  - F = 0 (no following frames),
  - L = 1 (R_LEN and DEPTH exist)
  - R_FT = 3 (the 3 previous AMR frame header fields FT were 3),
  - R_LEN = 2 (number of redundancy frame payload bits = 2*8 = 16)
  - DEPTH = 3 (the 3 previous AMR frame payload packets are taken for redundancy frame payload calculation)

  The redundancy frame paylaod covers 16 bits and is denoted by the value r(.).

	Bit no.							
Oct.	0	1	2	3	4	5	6	7
0	Q=1	I=1	R=1	0	0	1	1	0
1	F=1	F=0	L=0	L=1	0	0	0	0
2	0	0	1	1	1	1	f(0)	0
3	f(1)	0	f(2)	0	f(3)	0	f(4)	0
4	f(5)	1	f(6)	0	f(7)	0	f(8)	0
5	f(9)	1	f(10)	1	f(11)	r(0)	f(12)	r(1)
6	f(13)	r(2)	f(14)	r(3)	...	...	...	...
..	...	...	...	...	...	...	...	...
9	...	...	...	r(15)	f(27)	r(16)	f(28)	f(29)
..	...	...	...	...	...	...	...	...
33	...	...	...	...	f(130)	f(131)	f(132)	f(133)

Figure 12: Example with 1 AMR frame and 1 redundancy frame

7. References

[1] IETF RFC1889, "RTP: A Transport Protocol for Real-Time Applications"

[2] GSM 06.90, "Adaptive Multi-Rate (AMR) speech transcoding"

[3] ARIB, RCR STD-27H, Section 5.4, "ACELP Speech CODEC"

[4] TIA/EIA IS-641-A, "TDMA Cellular/PCS _Radio interface, Enhanced Full-Rate Voice Codec"

[5] GSM 06.60, "Enhanced Full Rate (EFR) speech transcoding"

[6] IETF draft-larzon-udplite-02.txt, "The UDP Lite Protocol"

[7] 3G TS 26.101, "AMR Speech Codec Frame Structure"

[8] IETF draft-lakaniemi-avt-rtp-amr-00.txt, "RTP Payload Format for AMR"

[9] IETF draft-sjoberg-avt-rtp-amr-00.txt, "RTP payload format for AMR"

[10] 3G TS 26.093, "AMR Speech Codec; Source Controlled Rate Operation"

[11] RFC 2119, "Key words for use in RFCs to Indicate Requirement Levels"

[12] IETF draft-wimmer-amr-01.txt, "MIME Type Registration for AMR Speech Codec"

8. Authors' addresses

Tim Fingscheidt
Siemens AG, ICP CD
Grillparzerstrasse 10-18
D - 81675 Munich
Germany
Phone: ++49 89 722 57658
Fax: ++49 89 722 46489
E-mail: Tim.Fingscheidt@mch.siemens.de

Bernhard Wimmer (contact person)
Siemens AG, ICP CD
Grillparzerstrasse 10-18

D - 81675 Munich
Germany
Phone: ++49 89 722 23247
Fax: ++49 89 722 46489
E-mail: Bernhard.Wimmer@mch.siemens.de

This Internet-Draft expires January, 14, 2001.