Internet Draft
draft-rey-avt-3gpp-timed-text-02.txt

J. Rey
Y. Matsui
Matsushita

Expires: August 2004
February 2004

RTP Payload Format for 3GPP Timed Text

Abstract

This document specifies an RTP payload format for the transmission of 3GPP (3rd Generation Partnership Project) timed text. 3GPP timed text is time-lined decorated text media format whose defined storage format is the ISO (International Standardisation Organisation) Base Media File Format. As of today, 3GP files containing timed text contents can be downloaded via HTTP and be synchronised with audio/video contents. There is however no available mechanism for streaming 3GPP timed text contents. In the following sections the problems of streaming timed text are addressed and a payload format for streaming 3GPP timed text over RTP is specified.

Table of Contents

1. Change Log

1.1 Changes from draft-rey-rtp-avt-3gpp-tt-00.txt

Major changes:
- completed empty sections from -00 draft.
- abstract and introduction re-arranged. Moved section "Basics of the 3GP File Structure" to end of the document as Annex B.
- SLEN, SIDX and SDUR lengths fixed to 16, 16 and 24 bits, respectively.
- New OPTIONAL header, SPLDESC, added to transport sample description in-band.
- Section 4 on payload format expanded: text header, fragment header and sample description header are fully specified.
- SMIL usage section added.

1.2 Changes from draft-rey-rtp-avt-3gpp-tt-01.txt

Major changes:

- Terminology, some terms introduced to clarify text.
- Section 4
  - rules and recommendations on fragmentation are given.
  - payload headers were calssified into five types, with a common field section and specific fields for each type.

- header structure similar to RFC 3640 for easy transformation.

2. Introduction

The purpose of this draft is to provide a means to stream the 3GPP timed text using RTP.

3GPP timed text is a 3GP file format for time-lined decorated text specified in [1]. The 3GP file format itself follows the ISO Base Media File Format recommendation [2]. Besides plain text, the 3GPP timed text format allows the display of decorated text (e.g. blinking text, scrolling, hyperlinks) synchronised or not with other media.

The 3GPP timed text format was developed for 3GPP Transparent End-to-end Packet-switched Streaming Services (PSS) [1]. The scope of the 3GPP PSS includes downloading and streaming of multimedia content over 3G packet-switched networks. The PSS adopts multimedia codecs (such as MPEG-4 Visual, AMR, MPEG-4 AAC, and JPEG) and protocols like SMIL [9] for presentation layouts or RTP for streaming. The current usage of the 3GPP timed text file format is limited to downloading via HTTP (with or without audio contents) due to the lack of an appropriate RTP payload format.

In general, a multimedia presentation might consist of several audio/video/text streams or tracks (in 3GP file format jargon). Different tracks may have different contents and tracks of different media may be spatially synchronised using the information within the tracks or a scene description language like SMIL. An example of this would be a media session with three different media tracks: 1 audio, 1 video and 1 timed text that reproduces a music video with karaoke subtitles. The information contained in each track defines the regions where each media is displayed, how the media looks like and how it is synchronised, e.g., the song lyrics is displayed below the video and the words are highlighted and synchronised with the soundtrack.

Basically the 3GPP timed text format can be summarised as consisting of four differentiated functional components:

- initial setup information for text tracks: these are the height and width of the text region where the text track (contents) are displayed, the translation offsets tx and ty relative to the video track region and the layer or proximity of the text to the user. In the 3GPP timed text format, these pieces of information are extracted from Track Header Box, "tkhd".
- general formatting information about the text track: default font, default background colour, default horizontal and vertical justification, default line width, default scrolling, etcetera. In

the 3GPP timed text format, these pieces of information are extracted from the Sample Description Box, "stsd".

- the actual text, conveyed as plain text using either UTF-8 or UTF-16 encoding and,

- the "decoration": whether it is highlighted text, blinking text, karaoke, hypertext, scroll delay, other text styles/formatting than the defaults, etcetera. In the 3GPP timed text format, these pieces of information are extracted from the various Modifier Boxes: "hlit", "hclr", "blnk", "krok", "href", "dlay", "styl" or "tbox".

For details refer to Annex A that summarises the basics of the 3GP file format and to [1], where a more detailed description of the setup information, format parameters and modifiers is contained.

2.1 Requirements

In this section a set of requirements is listed. A justification for each of them is also given. An RTP Payload Format for 3GPP timed text SHALL:

1. Keep the 3GP text sample structure. A text sample consists of the text length, the text string (either UTF-8 or UTF-16 encoded), and one or several Modifier Boxes containing the text "decoration", as defined in [1]. This is important to foster interoperability of 3GP file and RTP payload formats.

2. Transmit the text sample size, sample duration and sample description index in-band. In RTP it is important to transmit it in-band because this information might change from sample to sample. This is also important for buffering purposes as described in Section 4.1.

3. Enable the transmission of the formatting information (contained in the Sample Description Box, "stsd") by out-of-band and in-band means. In general, a single sample description may be used by different text samples. Therefore, to save overhead it is sensible to transmit a default formatting once at the initialisation phase and update this on demand. On the other hand, these pieces of information may become large so that out-of-band transmission might not be the most appropriate method. Also, out-of-band channels might not be always available. For these reasons, the payload format SHALL enable also the in-band transmission of sample description information. This is especially useful for live streaming (where the contents are not known a priori) or to protect this information through other mechanisms like FEC [4] or retransmission [13]. RFC 2354 [8] discusses available mechanisms for packet loss resiliency.

4. Enable the aggregation of text samples in one single RTP packet. In a mobile communication environment a typical text sample size is around 100-200 bytes. Thus, transporting several text samples in one RTP packet makes the transport over RTP more efficient.

5. Enable the fragmentation and reassembly of a text sample into several RTP packets in order to cover a wide range of applications and network environments. In general, fragmentation is

a rare event given the low bit rates and text sample sizes. However, the 3GPP Timed Text media format allows for larger text samples and so SHALL the payload format cover this possibility.

6. Enable the use of resilient transport mechanisms, such as repetition, retransmissions and FEC.

3. Terminology

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in RFC 2119 [5].

Furthermore, the following definitions are used in this document:

- text sample or whole text sample: this refers to a unit of text data as contained in the source 3GP file. Its equivalent in audio/video would be an audio/video frame (see Section 14 for details). A text sample contains a text length indication, a text string, and zero or several modifier boxes (the decoration of the text).
- fragment or text sample fragment: a fraction of a text sample as defined above. A fragment may either text strings and modifiers or just one of these.
- sample contents: general term to identify timed text data transported when using this payload format. In addition to text samples and fragments, it may also be used to refer to the sample descriptions associated to the text samples.

4. RTP Payload Format for 3GPP Timed Text

The format of an RTP packet containing 3GPP timed text is shown below:

```
    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
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
   |V=2|P|X| CC    |M|    PT       |        sequence number        |
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
   |                           timestamp                           |
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
   |           synchronization source (SSRC) identifier            |
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
   |                                                               |
   +                      RTP payload                              |
   |                                                               |
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Marker bit (M): the marker bit must be set to 1 if the RTP packet includes the last fragment of a text sample; otherwise set to 0. For

RTP packets containing several text samples, being at least one of them non-fragmented, the marker bit MUST be set to 1.

Timestamp: the timestamp indicates the sampling instant of the timed text sample contained in the RTP packet. The initial value is randomly determined. If the RTP packet includes more than one text sample (i.e. text sample aggregation), the timestamp indicates the sampling instant of the oldest text sample in the RTP packet. The samples MUST be placed in playout order, whereas the oldest sample is placed first in the payload. The timestamp of the subsequent samples is obtained by adding the timed text sample duration (if present) to the timestamp value. For example, let sdur(0), sdur(1) and sdur(2) be the durations of three subsequent timed text samples included in an RTP packet. Let rtpts be the timestamp as present in the RTP header. The timestamp ts(i) for each sample (i=0,1,2) would be:

```
             ts(i)=rtpts + sum[sdur (i-1)];

                ts(0)=rtpts,
                ts(1)=rtpts +  sdur(0)
                ts(2)=rtpts + (sdur(0)+ sdur(1))
```

Some text samples may become large and have to be fragmented and so spread over several RTP packets. In this case, the receiver needs to associate fragments of the same text sample. This is done using the timestamp. The order of the fragments is resolved using the fields available in the following headers.

The value timestamp clockrate is copied directly from the 3GP file: the value of "timescale" in the Media Header Box is to be used.

Payload Type (PT): the payload type is set dynamically and sent by out-of-band means.

The usage of the remaining RTP header fields follows the rules of RTP [3] and the profile in use.

4.1 Fragmentation of Timed Text Samples

This section justifies why text samples MAY be fragmented and discusses some of the possible approaches to do it. A solution is proposed together with rules and recommendations for fragmenting and transporting text samples using this payload format.

3GPP Timed Text applications are expected to operate at low bit rates. This fact added to the small size of timed text samples (typically one or two hundred bytes) makes fragmentation of text samples a rare event. Samples should usually fit into the MTU size of the used network path.

Nevertheless, the text string (e.g. ending roll in a movie) and some modifier boxes, i.e. for hyperlinks ("href"), for karaoke ("krok") or for fonts ("styl") might become large and need fragmentation. This may also apply for future modifier boxes. While the text string is recommended as per [1] to take a maximum of 2048 bytes for maximum client interoperability, there is no recommendation on the amount of space occupied by modifier boxes.

In order to transport these larger text samples using RTP, it could be argued that a careful encoding be used to transform the original large sample into smaller self-contained text samples that fit into the transport MTU. This would comply with the ALF principle, as per RFC 2367 [14]. It would also need additional pre-processing previous to RTP encapsulation. Given the low probability of fragmentation, it is believed that the overhead of this pre-processing (careful encoding) is not worth. It appears more appropriate to encode text samples without taking the path MTU into account. In this manner, this payload format meets a trade-off by intentionally leaving out this pre-processing and making some text samples more sensitive to packet losses.

However, a minimum set of fragmentation rules and recommendations SHALL be observed to guarantee a minimum resiliency and guide in the task of fragmentation. Text samples and fragments thereof are aggregated in the RTP payload according to the rules and recommendations specified as follows:

- it is RECOMMENDED that text samples are fragmented as seldom as possible. E.g. if a previous packet has some free space and a new text sample fits in one MTU, a new RTP packet SHOULD be sent, instead of sending two or more fragments out of it. In order to fill up the remaining bits, piggybacking of sample descriptions MAY be performed.

- text strings MUST split at character boundaries. Otherwise, it is not possible to display the text of a fragment if the previous is lost.

- it is RECOMMENDED to include as fewer text sample fragments as possible in an RTP packet. This reduces the effects of packet loss. RTP packets using this payload format MAY include zero or more whole text samples, zero or more text sample fragments and zero or more sample descriptions.

- sample descriptions SHALL NOT be fragmented, since they contain important information that may affect several text samples.

- for enhanced resiliency against packet loss it is RECOMMENDED that fragments containing decoration are especially protected using FEC [4], retransmission [13], packet repetition or a similar technique.

- when fragmenting text samples, the start of the decoration (modifiers) MUST be indicated. Otherwise, if packets are lost, a client may be unable to identify where the modifiers start and the text ends.

Usually, RTP applications use the information on packet size from UDP or lower layers to find out the length of the RTP payload. This means that if several text samples (or fragments) are contained in the payload a length indication MUST be present for all fragments, but the last one. Similarly, those transported as unique payload do not need of a length indication.

However, some transport schemes for RTP, e.g. RFC 3640 [15], require that the length of each fragment is indicated. This payload format does not mandate to comply with such requirement, but OPTIONALLY allows to do so. Implementations subject to such requirement MUST include an explicit length indication,i.e. the LEN field, by setting the L bit in all cases.

Note also that the order of the text samples and fragments in the RTP payload is important. As described above in the definition of the RTP timestamp usage, these MUST be placed chronologically in the RTP payload, so that the SDUR field allows calculating the timestamp of the samples following. At the same time, other samples MAY follow if the current and following samples contain each a length indication. Otherwise, the sample is either placed at the end or as unique RTP payload. Fragments carrying modifier box contents and sample descriptions MAY be placed in any order (no timing requirements) and MAY be present as often as needed. Modifier box fragments SHOULD be placed as close as possible to the text strings, which they belong to.

4.2 Payload Header Definitions

An RTP packet using the payload headers defined in this document has the following format:

```
    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
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
   |V=2|P|X| CC    |M|    PT       |        sequence number        |
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
   |                           timestamp                           |
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
   |           synchronization source (SSRC) identifier            |
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
   | Res.|U|L|TYPE |                                               |
   +-+-+-+-+-+-+-+-+                                               |
   :        (variable payload header depending on TYPE value)      :
   :                                                               :
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
   |                                                               |
    : SAMPLE CONTENTS = Text Sample(s), Fragment(s), Sample         :
    : Description(s)                                                :
    :                                                               :
    +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

The payload headers specified in this document consist of a set of common fields followed by specific fields for each header type.

The structure of the payload headers resembles that of the 'access units' in RFC 3640. This similarity is intentional, in order to ease the transport using MPEG4 elementary streams. In this manner, the 'AU header' of that document finds an equivalent in the common header fields for all TYPE values: R, U, L, TYPE and LEN. The specific fields plus the sample contents would be similar to the 'AU data section'.

The payload header the following format:

```
    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
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
   | R   |U|L|TYPE |             LEN (if L=1)      |   specific    |
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
   |   fields (variable)...|
   +-+-+-+-+-+-+-+-+-+-+-+-+
```

where,

- R (3 bits) "Reserved bits": this field MUST be set to zero.

- U (1 bit) "UTF Transformation": indicates whether the text characters are encoded using UTF-8 (U=0) or UTF-16 (U=1). It MUST also be present in text sample fragments if they contain text strings. If the payload does not carry text strings, this bit has no meaning and SHALL be disregarded.

- L (1 bit) "Length Field" flag: if set indicates the presence of the OPTIONAL length field, LEN. By default this flag is reset. Where several 'access units' are included in an RTP packet, the length MUST be used to parse the payload. In this case, it SHALL be present in all 'access units' but the last one. In some cases, as in RFC 3640 [15], the length field is always needed, and so MUST L flag be always set.

- LEN (16 bits) "Length Field": indicates the size (in bytes) of this field and all the bits following, i.e. specific payload header fields for each TYPE (excluding initial byte) plus the sample contents. If present, LEN has the following values:

 - TYPE = 1, LEN >= 6,
 - TYPE = 2, LEN > 9,
 - TYPE = 3, LEN > 3,
 - TYPE = 4, LEN > 3,
 - TYPE = 5, LEN > 3.

For whole text samples (TYPE=1), the sample contents length corresponds to the entry value in the Sample Size Box, "stsz" (see also SLEN below). Otherwise the sample contents length MUST be calculated when fragmenting the sample, in order to obtain the correct LEN value.

Empty text samples do not have sample contents. For this case, TYPE 1 with a LEN of 6 (0x0006) MUST be used.

The following payload header compositions (including the specific fields) result for the different values of TYPE:

- TYPE = 0,6 and 7 are reserved.
- TYPE = 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
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
   | R   |U|L|TYPE |       LEN (if L=1, always >=6)|    SIDX       |
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
   |                      SDUR                     |
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

This header type is used to transport whole text samples. If several text samples are sent in an RTP packet, every sample has its own header.

The LEN field MUST be always equal (for empty text samples) or greater than 6 (0x0006).

The fields above have the following meaning:

- SIDX (8 bits) "Text Sample Entry Index": indicates the reference index for the text sample, which corresponds to the index field in the Sample to Chunk Box, "stsc", for the sample. This field is used to map the corresponding sample description information. The SIDX is unequivocally linked to one particular sample description. Therefore, sample descriptions SHALL not be modified during a streaming session. A maximum of 126 SIDX values is allowed per text stream. To allow SIDX wrap-up, clients SHALL keep as valid only those values of SIDX outside of the interval (X+128) modulo 255, where X is the last SIDX value received. The SIDX values 0 (0x00) and 255 (0xff) are reserved for possible future extensions.

- SDUR (24 bits) "Text Sample Duration": indicates the sample duration in timestamp units of the text sample, which corresponds to the entry value in the Decoding Time to Sample Box, "stts", for that sample. This field allows by a clockrate of 1000 Hz a maximum duration of approximately 279 hours (16 bits is would allow for just 65 seconds, which might be too short for some streams).

It is assumed that all text samples have a known duration at the time of transmission. In some cases however, e.g. live streaming, the SDUR value might not be known. To cover this exception, the value zero (0x000000) is reserved to signal unknown duration. For all other cases SDUR MUST be different from 0x000000.

The ordering of 'access units' in the RTP payload is important. Logically, samples of unknown duration SHALL NOT precede any text samples or text sample fragments in the RTP payload. Otherwise it would not be possible to find out the timestamp of these. However, samples of unknown duration MAY precede sample descriptions, as they have no duration.

In general, text sample contents expire when the next sample becomes valid. As an exception, samples of unknown duration (SDUR=0x000000) are valid until new packets arrive.

Note also, that samples of unknown duration SHALL NOT use features, such as scrolling or karaoke, which would need to know the duration of the sample up-front.

- TYPE = 2,

```
    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
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
   | R   |U|L|TYPE |        LEN(if L=1, always >9) |    SIDX       |
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
   |               SLEN            |            SDUR               |
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
   | SDUR (cont.)  |TOTAL  |  THIS |
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

This header type is used to transport text sample fragments containing text strings.

The LEN field (16 bits) has the same meaning as above. If present, the length of LEN MUST be greater than a value of nine (0x0009).

The SLEN field (16 bits) indicates the size (in bytes) of the (whole) text sample to which this fragment belongs. As seen above, the text sample length corresponds to the entry value in the Sample Size Box, "stsz". Clients MAY use SLEN to buffer space for the remaining fragments of the text sample.

The fields TOTAL (4 bits) and THIS (4 bits) indicate the total number of fragments in which the original text sample has been fragmented and which order occupies the current fragment in that sequence, respectively. The usual 'byte offset' field is not used here for two reasons: a) it would take one more byte and b) it does not provide any useful information on the character offset. UTF-8/16 text strings have, in general, a variable character length ranging from 1 to 6 bytes. Therefore, the TOTAL/THIS solution is preferred.

The R, U, L, TYPE, SIDX, and SDUR fields have identical interpretation as above. The U, SIDX and SDUR fields are useful since partial text strings MAY also be displayed with the corresponding decoration.

- TYPE = 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
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
   | R   |U|L|TYPE |        LEN(if L=1, always >3) |TOTAL  |  THIS |
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

This header type is used to transport either whole modifier boxes or just the first fragment of these. This depends on whether the modifier boxes fit into one RTP packet.

In case fragmentation is needed, this header type identifies first fragment. As explained above, the rules for fragmentation require that the start of the modifier boxes be signaled.

The R, U, L, TOTAL/THIS and LEN fields are used as above. If present, the LEN field MUST be greater than three (0x0003).

Note that the SLEN, SIDX and SDUR fields are not present. This is because: a) these fragments do not contain text strings and b) these types of fragments are applied over text string fragments, which already contain this information.

- TYPE = 4,

```
    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
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
   | R   |U|L|TYPE |        LEN(if L=1, always >3) |TOTAL  |  THIS |
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

This header type is used to transport modifier fragments, other than the first one.

The R, U, L, TOTAL/THIS and LEN fields are used as above. If present, the LEN field MUST be greater than three (0x0003).

Note that the SLEN, SIDX and SDUR fields are not present. This is because: a) these fragments do not contain text strings and b) these types of fragments are applied over text string fragments, which already contain this information.

- TYPE = 5

```
    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
```

```
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
   | R   |U|L|TYPE |      LEN(if L=1, always >3)   |   SIDX        |
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

This header type is used to transport sample descriptions. The L flag MUST be always set. Consequently, the LEN field MUST be greater than three (0x0003). Every sample description MUST have its own TYPE 5 header.

5. Resilient Transport

Apart from the basic fragmentation measures described in the section above, the simplest option for packet loss resilient transport is to send the same text samples (or fragments) again, i.e. repetition. A server MAY decide to send the same contents again as a measure for error resilience.

Repetition of text samples (or fragments) is only allowed if exactly the same RTP payloads are sent, so that a receiver can use original and repeated fragments together to reconstruct the text samples. If a text sample was originally sent as non-fragmented text sample, a repetition of that sample MUST be sent also as a single non-fragmented text sample. Likewise, if the original text sample was fragmented and spread over several RTP packets, repeated fragments SHALL also be observe the same byte boundaries and use the same headers and bytes per fragment. Finally, if several text samples resolve to the same timestamp, the receiver SHOULD use the one received in the RTP packet with the highest sequence number and discard the rest.

In repeated text samples (or fragments), all RTP header fields MUST keep their original values except the sequence number that MUST be increased to comply with RTP.

6. Congestion control

The RTP profile under which this payload format is used defines an appropriate congestion control mechanism in different environments. Following the rules under the profile, an RTP application can determine its acceptable bitrate and packet rate in order to be fair to other TCP or RTP flows.

If an RTP application using this payload format uses retransmission, the acceptable packet rate and bitrate includes both the original and retransmitted data. This guarantees that an application using retransmission achieves the same fairness as one that does not. Such a rule may translate in practice into the following actions:

If enhanced service is used, it should be made sure that the total bitrate and packet rate do not exceed that of the requested service. It should be further monitored that the requested services are actually delivered. In a best-effort environment, the sender SHOULD NOT send retransmission packets without ensuring first that enough bandwidth for retransmission is available. Other solutions like reducing the packet rate and bitrate of the original stream (for example by encoding the data at a lower rate) MAY be used.

Similar considerations apply, if an RTP application using this payload format implements forward error correction, FEC [4]. Hereby, the sender should take care that the amount of FEC does not actually worsen the problem.

Therefore, it is RECOMMENDED that applications implementing this payload format also implement congestion control. The actual mechanism for congestion control is out of the scope of this document but should be suitable for real-time flows. As an example, RFC 3448 [11] specifies an equation-based congestion control that fulfils this requirement.

7. SMIL usage

The SMIL recommendation [9] specifies a means for synchronising different media streams.

This payload format defines the spatial layout parameters for a timed text stream. These specify the location of the text display area relative to the top left corner of the video display area, when a text stream is played with a single video stream without SMIL. In cases where several media streams shall be synchronized, SMIL MAY be used to specify the spatial layout parameters.

It shall be noted that even if SMIL scene description is used the track header information pieces SHOULD be sent anyway as they represent the intrinsic media properties.

8. MIME Type usage Registration

8.1 3GPP Timed Text MIME Registration

MIME type: video

MIME subtype: 3gpp-tt

Required parameters:

rate: the RTP timestamp clockrate is equal to the clockrate of the media. The value timestamp clockrate is copied directly from the 3GP file: the value of "timescale" in the Media Header Box is to be used.

brand=<brand-name>, where <brand-name> identifies a Release specification of 3GPP Timed Text being transmitted over RTP. A brand indicates the "best use" of the contents: the brand value "3gp5" indicates Release 5 of 3GPP Technical Specification (TS) 26.245 [1].

spldesc=<flag>, where <flag> may take three different values:

- spldesc=in, for in-band transmission of sample descriptions
- spldesc=out, for out-of-band and,
- spldesc=both, when both methods are allowed.

tx3g=<base64-value-1>, <base64-value-2>,... where <base64-value-i> represents a list of sample description entries using base64 encoding. This parameter MAY be used to convey sample descriptions out-of-band. The list of sample entries is not required to follow any particular order. Each value <base64-value-i> represents the concatenation of the SIDX and sample descriptions contents for that SIDX. The LEN field is not needed.

width=<integer-value> indicates the width in pixels of the text track or area where the text is actually displayed.

height=<integer-value> indicates the height in pixels of the text track.

tx=<integer-value>, indicates the horizontal translation offset in pixels of the text track with respect to the origin of the video track.

ty=<integer-value>, indicates the vertical translation offset in pixels of the text track.

layer=<integer-value>, indicates the proximity of the text track to the viewer. Higher values means closer to the viewer. This parameter has no units.

Optional parameters:

mver=<version-value>, "Minor version" where <version-value> is a positive integer. It identifies the oldest compatible version. How the version is defined can be found in TS 26.245 "3GPP Transparent end-to-end packet switched streaming service (PSS); Timed Text Format (Release 6)".

cbrand=<value1,value2,...>, "List of Compatible Brands" where value1 is a brand. This list MUST at least contain the <brand-name> as in "brand".

Encoding considerations: this type is only defined for transfer via RTP.

Security considerations: please refer to Section 12 of RFCXXXX.

Interoperability considerations: the 3GPP Timed Text media format is specified in 3GPP Release 5 version of TS 26.245 "Transparent end-to-end packet switched streaming service (PSS); Timed Text Format (Release 6)". In later releases, 3GPP may specify extensions or updates to the media format in a backwards-compatible way, e.g. new modifier boxes or extensions to the sample descriptions. The payload format RFCXXXX allows for such extensions. For future 3GPP Releases of the Timed Text Format, the therein parameters "brand", "mver" and "cbrand" are used to identify the current version of the media format, the oldest compatible version and a list of compatible versions.

Published specification: RFC XXXX

Applications which use this media type: multimedia streaming applications.

Additional information: the 3GPP Timed Text media format is specified in 3GPP TS 26.245 "Transparent end-to-end packet switched streaming service (PSS); Timed Text Format (Release 6)". This document and future extensions to the 3GPP Timed Text format are publicly available at http://www.3gpp.org.

Person & email address to contact for further information:
rey@panasonic.de
matsui.yoshinori@jp.panasonic.com

Intended usage: COMMON

Author/Change controller:
Jose Rey
Yoshinori Matsui
IETF AVT WG

9. SDP usage

This document defines the MIME subtype name "3gpp-tt" and introduces several REQUIRED payload-format-specific parameters: "brand", "width", "height", "tx", "ty", "layer", "spldesc" and "tx3g" and two OPTIONAL parameters "mver" and "cbrand".

9.1 Mapping to SDP

The information carried in the MIME media type specification has a specific mapping to fields in SDP [4], which is commonly used to describe RTP sessions. When SDP is used to specify transmission using this payload format, the mapping is done as follows:

- The MIME type ("video") goes in the SDP "m=" as the media name. The "video" MIME Type is used as timed text is considered visual media.

- The MIME subtype ("3gpp-tt") goes in SDP "a=rtpmap" as the encoding name. The value timestamp clockrate is copied directly from the 3GP file, the value of "timescale" in the Media Header Box is to be used. Other values MAY be specified by out-of-band means.

- The REQUIRED payload-format-specific parameters "brand", "width", "height", "tx", "ty", "layer", "tx3g" and "spldesc" go in the SDP "a=fmtp" as a semicolon separated list of parameter=<value> (or parameter=<value1,value2,value3> for "tx3g") pairs.

- The OPTIONAL payload-format-specific parameters "mver", "cbrand" go in the SDP "a=fmtp" as a semicolon-separated list of parameter=<value> pairs.

- Any remaining parameters go in the SDP "a=fmtp" attribute by copying them directly from the MIME media type string as a semicolon separated list of parameter=value pairs.

In the following sections some example SDP descriptions are presented.

10. Examples of RTP packet structure

In this section, some examples of RTP packet structure are explained for better understanding of this payload format. The wrap-around of the long lines is indicated by the backslash character "\". The examples assume aggregate control of stream container files. The session descriptions are not complete but limited to the example purposes.

10.1 An RTP packet containing multiple text samples

<TODO>

11. IANA Considerations

This document introduces the MIME subtype name "3gpp-tt" in Section 8.

12. Security considerations

RTP packets using the payload format defined in this specification are subject to the security considerations discussed in the RTP specification [3]. This implies that confidentiality of the media streams is achieved by encryption.

Furthermore, the main security issues are confidentiality and authentication of the text itself. The payload format itself does not have any support for security. These issues have to be solved by a payload external mechanism, e.g. SRTP [10].

13. References

13.1 Normative References

1 3GPP, "Transparent end-to-end packet switched streaming service (PSS); Timed Text Format (Release 6)", TS 26.245 v 0.1.6, Working Draft, July 2003.

2 ISO/IEC 14496-1:2001/AMD5, "Information technology û Coding of audio-visual objects û Part 1: Systems, ISO Base Media File Format", 2003.

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

4 M. Handley, V. Jacobson, "SDP: Session Description Protocol", RFC 2327, April 1998.

5 S. Bradner, "Key words for use in RFCs to indicate requirement levels," BCP 14, RFC 2119, IETF, March 1997.

13.2 Informative References

6 C. Perkins, I. Kouvelas, O. Hodson, V. Hardman, M. Handley, J.C. Bolot, A. Vega-Garcia, S. Fosse-Parisis, "RTP Payload for Redundant Audio Data", September 1997.

7 J. Rosenberg, H. Schulzrinne, "An RTP Payload Format for Generic Forward Error Correction", RFC 2733, December 1999.

8 C. Perkins, O. Hodson, "Options for Repair of Streaming Media", RFC 2354, June 1998.

9 W3C, "Synchronised Multimedia Integration Language (SMIL 2.0)", August, 2001.

10 M. Baugher, D. A. McGrew, D. Oran, R. Blom, E. Carrara, M. Naslund, K. Norrman, "The Secure Real-Time Transport Protocol", draft-ietf-avt-srtp-05.txt, June 2002.

11 Handley, et al., "TCP Friendly Rate Control (TFRC): Protocol Specification ", RFC 3448, January 2003.

12 R. Hovey, S. Bradner, "The Organizations involved in the IETF Standards Process", BCP 11, RFC 2028, October 1996.

13 J. Rey et al., "RTP Retransmission Payload Format", draft-ietf-avt-rtp-retransmission-10.txt, work in progress, January 2004.

14 M. Handley, C. Perkins, "Guidelines for Writers of RTP Payload
  Format Specifications", RFC 2367, December 1999.

15 Van der Meer et al., "RTP Payload Format for Transport of MPEG-4 Elementary Streams ", RFC3640, November 2003.

14. Annex A Basics of 3GP File Structure

Each 3GP file consists of "Boxes". Boxes start with a header which indicates both size and type contained. The 3GP file contains the File Type Box (ftyp), the Movie Box (moov), and the Media Data Box (mdat). The Movie Box and the Media Data Box, serving as containers, include own boxes for each media. Similarly, each box type may include a number of boxes, see ISO Base Media file Format [2] for a complete list of possibilities.

In the following, only those boxes are mentioned, which are useful for the purposes of this payload format.

The File Type Box identifies the type and properties of a 3GP file. The File Type Box contents comprise the major brand, the minor version and the compatible brands. These are communicated via out-of-band means, such as SDP, when streamed with RTP. For the 3GPP timed text file format, the set of compatible-brands MUST include "3gp5".

The Movie Box contains one or more Track Boxes (trak) which include information about each track. A Track Box contains, among others, the Track Header Box (tkhd), the Media Header Box (mdhd) and the Media Information Box (minf). The Media Header Box contains the timescale or number of time units that pass in one second. The Media Information Box includes the Sample Table Box (stbl) which itself contains the Sample Description Box (stsd), the Decoding Time to Sample Box (stts), the Sample Size Box (stsz) and the Sample to Chunk Box (stsc). Sample descriptions for each text sample are encoded as "tx3g" sample entries in the Sample Description Box (stsd).

The Track Header Box specifies the characteristics of a single track, where a track is, in this case, the streamed text during a session. Exactly one Track Header Box is needed for a track. It contains information about the track, such as the spatial layout (width and height), the video transformation matrix and the layer number. Since these pieces of information are essential and static, i.e. constant for the duration of the session, they MUST be sent prior to the transmission of any text samples. See the ISO base media file format [2] for details about the definition of the conveyed information.

When using scene description in SMIL [9], it is possible to specify the layer and the position of the text track. However, in this case, the transmission of the Track Header Box (tkhd) is still RECOMMENDED, as the intrinsic track information is specified there. Otherwise, the Track Header Box information MUST be sent prior to the start of the text streaming.

The Sample Table Box (stbl) contains all the time and data indexing of the media samples in a track. Using the tables here, it is possible to locate samples in time, determine their type, and determine their size, container, and offset into that container. From the Sample Table Box (stbl) the following information is carried in each RTP packet using this payload format: the Sample Description Box (stsd), the Decoding Time to Sample Box (stts), the Sample Size Box (stsz) and the Sample to Chunk Box (stsc). The Decoding Time to Sample Box (stts) is mapped to the field SDUR (Text Sample Duration); the Sample Size Box (stsz) is mapped the field SLEN (Text Sample Length) and the Sample to Chunk Box is mapped to the field SIDX (Text Sample Entry Index). The Sample to Chunk Box (stsc) associates the text sample and its corresponding sample description entry in the Sample Description Box (stsd, see below). The Sample to Chunk Box can be used to associate a text sample with a sample description entry. Since the sample description may vary during the session, the association SDIX must be sent together with the text samples using this payload format.

The Sample Description Box (stsd) provides information on the basic characteristics of text samples. Each entry is a sample entry box of type "tx3g". An example of the information contained in a sample entry could be the font size or the background colour. Since these pieces of information are commonly used by many text samples during the session, it is sent by out-of-bands means. A complete list of text characteristics can be found in [1].

Finally, the Media Data Box contains the media data itself. In 3GPP timed text tracks this box contains text samples. Its equivalent to audio and video is audio and video frames, respectively. The text sample consists of the text length, the text string, and one or several Modifier Boxes. The text length is the size of the text in bytes. The text string is plain text to render. The Modifier Box is information to render in addition to the text such as colour, font, etc.

15. Author's Addresses

Jose Rey   rey@panasonic.de
Panasonic European Laboratories GmbH
Monzastr. 4c
D-63225 Langen, Germany
Phone: +49-6103-766-134
Fax: +49-6103-766-166

Yoshinori Matsui   matsui.yoshinori@jp.panasonic.com
Matsushita Electric Industrial Co., LTD.

1006 Kadoma
Kadoma-shi, Osaka, Japan
Phone: +81 6 6900 9689
Fax: +81 6 6900 9699

16. IPR Notices

The IETF takes no position regarding the validity or scope of any intellectual property or other rights that might be claimed to pertain to the implementation or use of the technology described in this document or the extent to which any license under such rights might or might not be available; neither does it represent that it has made any effort to identify any such rights. Information on the IETF's procedures with respect to rights in standards-track and standards-related documentation can be found in BCP 11 [12]. Copies of claims of rights made available for publication and any assurances of licenses to be made available, or the result of an attempt made to obtain a general license or permission for the use of such proprietary rights by implementers or users of this specification can be obtained from the IETF Secretariat.

The IETF invites any interested party to bring to its attention any copyrights, patents or patent applications, or other proprietary rights which may cover technology that may be required to practice this standard. Please address the information to the IETF Executive Director.

17. Full Copyright Statement