Network Working Group   D. Singer
Request for Comments: 5484   Apple Computer Inc.
Category: Standards Track   March 2009

Associating Time-Codes with RTP Streams

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.

Copyright Notice

Abstract

This document describes a mechanism for associating time-codes, as defined by the Society of Motion Picture and Television Engineers (SMPTE), with media streams in a way that is independent of the RTP payload format of the media stream itself.

Table of Contents

1. Introduction

First a brief background on time-codes [SMPTE-12M].

The time-code system in common use is defined by the Society of Motion Picture and Television Engineers (SMPTE); in it, time-codes count frames. A common form of the display looks like a normal clock value (hh:mm:ss.frame). When the frame rate is truly integral, then this can be a normal clock value, in that seconds tick by at the same rate as the seconds we know and love.

However, NTSC video infamously runs slightly slower than 30 frames per second (fps). Some people call it 29.97, which isn't quite right; to be accurate, a frame takes 1001 ticks of a 30000 tick/second clock. Be that as it may, SMPTE time-codes count 30 of these frames and deem that to make a second.

This causes an SMPTE time-code display to 'run slow' compared to real-time. To ameliorate this, sometimes a format called drop-frame is used. Some of the frame numbers are skipped, so that the counter periodically 'catches up' (so some time-code seconds actually only have 28 frames in them).

It is worth noting that in neither case is the SMPTE time-code an accurate clock; in the first case, it runs slow, and in the second, the adjustments are abrupt and periodic -- and still not quite accurate. Hence the rest of this document tries to be clear when referring to a second in a time-code as a 'time-code second'.

However, SMPTE time-codes do run in real-time when used with systems with integral fps (e.g., film content at 24 fps or PAL video).

This specification defines how to carry time-codes in RTP and RTCP (RTP Control Protocol), associate them with a media stream, and synchronize them with the RTP timestamps. It uses the general RTP header extension mechanism [RFC5285].

2. Requirements Notation

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].

3. Design Goals

What we desire is a system that allows us to associate an SMPTE time-code with some media in an RTP [RFC3550] stream. Since in RTP all media has a clock already, we can often leverage that fact. If we treat the media as having 'segments' of time in which the time-code is simply counting up, then the time-code anywhere within a segment can be calculated if you know:

- the RTP timestamp of the start of the segment;
- the time-code of the start of the segment;
- the counting rate and other parameters of the time-code;
- the RTP timestamp where you want to know the time-code.

There are two cases to consider:

1. the time-codes are piece-wise continuous with only occasional discontinuities;
2. the continuity of the time-codes is not certain (or not known).

The first can be handled by providing details of the time-code axis and an initial mapping from RTP time to time-code time as well as periodic mappings in RTCP packets. This is defined in Section 6.3.

The second requires in-band signaling within the RTP packets themselves. This is defined in Section 6.4.

There are applications where the transport of all 8 bytes of the SMPTE 12M time-code are important (e.g., when the date of the time-code must be known or when the RTP transport is used as a transparent pipe). On the other hand, there are cases (e.g., when time-codes are used with compressed audio) when bandwidth is also important. To support both use cases, provision is made for both compact and full forms of the time-code.

4. Requirements and Constraints

Receivers MUST support time-codes in both RTCP and RTP as well as both forms (compact and full) of the time-code. Senders, of course, are free to choose.

Note that the compact form allows frame numbers greater than the full form (a field of 6 bits vs. a full binary-coded decimal (BCD) digit and a 2-bit BCD digit, which gives a maximum transmitted value of 29). In some cases, the color frame flag (bit 11) is used to 'extend' the "tens of frames" field from 2 to 3 bits; however, such practices are outside the scope of this specification.

In the case that a presentation contains more than one stream, senders MUST continue to send the standard RTP synchronization information in RTCP, even if the streams carry SMPTE time-codes that could be used for synchronization. In fact, when time-codes are carried by more than one stream, this document does not constrain the time-codes: at a given point in time, they may be the same, or they may differ (e.g., if they carry the original time-codes of different source material that was edited together).

5. Signaling Information

If the recipient must ever calculate time-codes based on the RTP times, then some setup information is needed. This MUST be sent out-of-band -- for example, in a SIP offer/answer exchange [RFC3264]. Since this specification is a general header extension [RFC5285], when the Session Description Protocol (SDP) is used, the 'extmap' attribute defined by the extension mechanism is also used.

The setup information should include:

1. the duration, in the RTP timescale, of a single frame-count in the 'frames' portion of the time-code (frame_duration)

2. the number of those frames that make a time-code second (frames_per_tc_second); framecounter values may be between 0 and (frames_per_tc_second - 1)

3. the drop-frame indication, is-NTSC-drop-frame, which indicates whether the usual drop-frame behavior should be applied or not

Note that other information we need to do the calculation (e.g., the clock rate of the RTP timestamp) is supplied already and assumed to be available.

For example, if associated with a video stream with the common time-scale of 90000 ticks per second, then a frame_duration of 3003 and frames-per-tc-second of 30 would yield a 'normal' SMPTE time-code for NTSC video. Similarly, values of 3750 and 24 yield a time-code for 24 fps film content, and so on.

Note also that we supply explicitly the frame duration and fps, even though they are obviously closely related. This removes any ambiguity of what the counter values should be in the case of drop-frame counting. These three values MUST correspond with each other.

When the SDP is used, these three parameters are transmitted as extensionattributes, as defined in the header extension specification [RFC5285], with the following ABNF syntax [RFC5234]. The form of the extension attributes is 'owned' by the extension name. These parameters to the extension do not need registration action beyond their documentation here. Note that the parameters are supplied as extension attributes, suitable for in-line use in RTP, even if in a given stream only the RTCP mapping is used.

```
digit = "0"/"1"/"2"/"3"/"4"/"5"/"6"/"7"/"8"/"9"

integer = 1*digit

frame-duration-length = integer

timestamp-rate = integer

frame-duration = frame-duration-length "@" timestamp-rate

frames-per-tc-second = integer

drop = "/drop"

extensionattributes = frame-duration "/" frames-per-tc-second [drop]
```

The frame duration is specified as a count of ticks of a clock that has timestamp-rate ticks per second. It is recommended that the timestamp-rate be the same as the clock rate of the RTP stream in which the extension is embedded, to avoid the loss of accuracy in conversion of timestamps. If the payload type changes during a stream, especially between payloads with different clock rates, it is strongly recommended that the header extension be included on the first packet(s) of the new payload, to set the mapping for the new clock rate explicitly.

If '/drop' is specified, then the first two frame numbers are omitted from the count of each minute, except for minutes 00, 10, 20, 30, 40, and 50, as documented in Section 4.2.2 of SMPTE specification [SMPTE-12M]. (Note that this usually only applies to NTSC video.)

The URI used for the signaling is

"urn:ietf:params:rtp-hdrext:smpte-tc".

This URI signals the possible presence of associations in RTCP or RTP, as defined below.

An example in the SDP, for film material, on a stream with a timescale of 600, might be:

```
a=extmap:4 urn:ietf:params:rtp-hdrext:smpte-tc 25@600/24
```

Another example, for drop-frame NTSC, on a stream with a timescale of 600, might be:

```
a=extmap:4 urn:ietf:params:rtp-hdrext:smpte-tc 20@600/30/drop
```

6. In-Stream Information

6.1. Compact Format of the Time-Code

A compact binary SMPTE time-code in this design occupies 24 bits. It is NOT formatted in the BCD system, but uses binary fixed-width fields. It has the following structure:

sign(1) -- 1 for negative, 0 for positive

hours (5 bits) -- 0 to 23; the values 24-31 are reserved

minutes (6 bits) -- 0 to 59; 60-63 are reserved seconds (6 bits) -- 0 to 59; 60-63 are reserved

frames(6 bits) -- 0 to (frames-per-tc-second - 1)

Note that these fields are larger than the provision in SMPTE 12M, where BCD (binary-coded decimal) is used (and notably, where only two bits are provided for the tens digit of the frame-count, so frame numbers above 39 cannot be represented).

6.2. Full Format of the Time-Code

A full SMPTE time-code occupies 64 bits. It is formatted exactly as defined in Sections 7 and 8 of SMPTE 12M [SMPTE-12M], without the 16-bit syncword. The value of the "drop frame flag" MUST agree with the use of the "drop" indicator in the signaling.

Here are the bit assignments from SMPTE 12M, for information:

0--3 Units of frames

4--7 First binary group

8--9 Tens of frames

10 Drop frame flag

11 Color frame flag

12--15 Second binary group

16--19 Units of seconds

20--23 Third binary group

24--26 Tens of seconds

27 Polarity correction

28--31 Fourth binary group

32--35 Units of minutes

36--39 Fifth binary group

40--42 Tens of minutes

43 Binary group flag BGF0

44--47 Sixth binary group

48--51 Units of hours

52--55 Seventh binary group

56--57 Tens of hours

58 Binary group flag BGF1

59 Binary group flag BGF2

60--63 Eighth binary group

6.3. Associations in RTCP

When the time-codes are piece-wise continuous, we then supply in RTCP packets an RTP timestamp and an SMPTE time-code for the start of each run of calculable time-codes. This establishes the time-code for all RTP times greater than or equal to the one given, until a subsequent RTCP packet reestablishes the mapping.

Note that the RTP timestamp in the RTCP mapping may not match the timestamp of any frame in the media stream. For video, it normally would; but a timestamp transition may happen part-way through a decoded audio frame. Since they share the same clock, the timing of that transition and the timing of the audio stream itself have the same accuracy.

The RTCP packets need not use the same RTP timestamp as the sender report (or transmission time) in the same RTCP packet. They can be sent 'ahead of need' if possible (e.g., for stored content, when the server can look ahead) or 'just-in-time'. For example, packets sent 'just-in-time' may be sent as early feedback packets, following the rules in [RFC4585], after a discontinuity in the time-code is detected. Such packets allow media-buffering in the client the chance to 'catch' the RTCP before the matching RTP packet is processed and displayed.

The association is a new RTCP Control Packet Type, using the value 194 (see Section 10). This control packet has one of the two following forms, differentiated by its length.

```
    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|    SC   |PT=SMPTETC=194 |             length=3          |
   +=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+
   |                     SSRC of packet sender                     |
   +=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+
   |                         RTP timestamp                         |
   +=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+
   |S|  hours  |  minutes  |  seconds  |  frames   |  reserved=0   |
   +=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+
```

Figure 1: RTCP Short Form Packet

The fields S (sign), hours, minutes, seconds, and frames are defined in Section 6.1.

For this short form, the length takes the fixed value 3, indicating a control packet of 4 32-bit words.

```
    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|    SC   |PT=SMPTETC=194 |             length=4          |
   +=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+
   |                     SSRC of packet sender                     |
   +=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+
   |                         RTP timestamp                         |
   +=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+
   |                          Full 8-byte                          |
   |                      SMPTE 12M time-code                      |
   +=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+
```

Figure 2: RTCP Full Form Packet

For this full time-code (long form), the length takes the fixed value 4, indicating a control packet of 5 32-bit words.

6.4. Associations in RTP

When the time-codes are not known to be piece-wise continuous, or absolute surety of mapping is desired, then the mapping can be placed into some or all of the RTP packets. This is a less desirable route; it uses the RTP header extension [RFC5285], which some terminals may find problematic. And clearly placing mapping information in every packet uses more bandwidth.

In as many RTP packets as needed (possibly all), an RTP header extension is used [RFC5285] to associate an RTP time to an SMPTE time-code.

There are two forms of this header extension, again differentiated by their length. The short form associates a compact time-code with the RTP timestamp of the packet. The long form allows associates a full time-code with a timestamp offset from the RTP timestamp of the packet.

The short form has a length of 3 bytes (24 bits). The long form has a length of 12 bytes (96 bits) and consists of a full SMPTE 12M time-code, followed by a signed 32-bit offset D from the RTP timestamp. If the packet has timestamp T, this establishes an RTP to time-code association for the RTP time T+D.

7. Implementation Note (Informative)

This section contains a suggestion on how to calculate both a time-code for a time T2, given an initial code at time T1, and the frame duration.

It might seem that when drop-frame is used, there is a 'fence post' problem: how many minutes in which frame-numbers are dropped have passed since the initial time-code? However, this can be avoided if all calculations are 'zero-based'; then the number of 'fence posts' is known.

```
framesSinceTCzero := TimeCodeToFrameCount( initialTimeCode );
framesSinceMapping := floor( (T2-T1)/frameDuration );
totalFrames := framesSinceTCzero + framesSinceMapping;
timeCode := FrameCountToTimeCode( totalFrames );
```

The SMPTE engineering guideline [SMPTE-EG40] contains all the appropriate equations, constants, etc. for performing these and other conversions.

8. Discussion (Informative)

This design has the advantage of not requiring the introduction of new IP packets into the sessions or new data into the main data channel by using low-bandwidth (vanishingly low in the case of streams with no discontinuities), and it is independent of the design of the RTP packets themselves: the RTP profile (including possibly encryption) and the RTP payload format. SMPTE time-codes can be associated with any RTP stream, including those with existing payload formats.

It might be argued that we could set the initial mapping also in the SDP, since RTCP packets might get lost. But this means that the SDP now has to have knowledge of the RTP random offset, which is nasty; also, if one puts this RTCP packet into all sender reports, that's probably good enough. Then if you don't have time-codes, you don't have audio-video-sync either.

This specification associates the time-code with a particular media stream. An alternative would be to make it an RTP stream in its own right; however, the data rate is so low, this seems egregious. By packing it inline, we can do this backwards-compatible for gateways, etc., that already handle dual-stream.

There is no way described in this document to detect that an RTCP packet has been lost and that a mapping may be being used outside its intended range.

The design assumes that clients will hold mappings until they are superseded, and that a client may need to buffer some number of upcoming mappings.

9. Security Considerations

SMPTE time-codes are only informative and there are no known security considerations from associating them with media streams.

10. IANA Considerations

The RTCP packet type used for SMPTE time-codes has been registered, in accordance with Section 15 of [RFC3550]. IANA has added a new value to the RTCP Control Packet types sub-registry of the Real-Time Transport Protocol (RTP) Parameters registry, according to the following data:

abbrev.	name	value	Reference
SMPTETC	SMPTE time-code mapping	194	RFC 5484

Additionally, IANA has registered a new extension URI to the RTP Compact Header Extensions sub-registry of the Real-Time Transport Protocol (RTP) Parameters registry, according to the following data:

```
Extension URI: urn:ietf:params:rtp-hdrext:smpte-tc
Description:   SMPTE time-code mapping
Contact:       singer@apple.com
Reference:     RFC 5484
```

11. Acknowledgments

Both Brian Link and John Lazzaro provided helpful comments on an initial draft. Colin Perkins was helpful in reviewing and dealing with the details. Ladan Gharai provided a thoughtful final review.

12. References

12.1. Normative References

[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.

[RFC4585] Ott, J., Wenger, S., Sato, N., Burmeister, C., and J. Rey, "Extended RTP Profile for Real-time Transport Control Protocol (RTCP)-Based Feedback (RTP/AVPF)", RFC 4585, July 2006.

[RFC5234] Crocker, D. and P. Overell, "Augmented BNF for Syntax Specifications: ABNF", STD 68, RFC 5234, January 2008.

[RFC5285] Singer, D. and H. Desineni, "A General Mechanism for RTP Header Extensions", RFC 5285, July 2008.

12.2. Informative References

[SMPTE-12M] Society of Motion Picture and Television Engineers, "SMPTE Standard for Television -- Time and Control Code", SMPTE 12M-1-2008.

[SMPTE-EG40] SMPTE, "Conversion of Time Values Between SMPTE 12M Time Code, MPEG-2 PCR Time Base and Absolute Time", SMPTE EG40-2002, August 2002.

Author's Address

David Singer
Apple Computer Inc.
1 Infinite Loop
Cupertino, CA 95014
US

Phone: +1 408 996 1010
EMail: singer@apple.com