IPR Notice

By submitting this Internet-Draft, each author represents that any applicable patent or other IPR claims of which he or she is aware have been or will be disclosed, and any of which he or she becomes aware will be disclosed, in accordance with Section 6 of BCP 79.

Status of This Memo

This document is an Internet-Draft and is in full conformance with all provisions of Section 10 of RFC2026.

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF), its areas, and its working groups. Note that other groups may also distribute working documents as Internet-Drafts. __

Internet-Drafts are draft documents valid for a maximum of six months and may be updated, replaced, or obsoleted by other documents at any time. It is inappropriate to use Internet-Drafts as reference material or to cite them other than as "work in progress." __

The list of current Internet-Drafts can be accessed at http://www.ietf.org/1d-abstracts.html __The list of Internet-Draft Shadow Directories can be accessed at http://www.ietf.org/shadow.html

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.

Distribution of this document is unlimited.

Copyright Notice

Copyright (C) The Internet Society (2005).
This document is subject to the rights, licenses and restrictions contained in BCP 78, and except as set forth therein, the authors retain all their rights.

Abstract
This document describes a mechanism for associating SMPTE time-codes with media streams, in a way that is independent of the RTP payload format of the media stream itself.

1 Introduction

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

SMPTE time-codes count frames. There are two common forms of display: either a simple counter, or what looks like a normal clock value (hh:mm:ss.frame). When the frame rate is truly integer, 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/second. Some people call it 29.97 (which isn’t quite right) and some say that a frame takes 1001 ticks of a 30000 tick/second clock (which is closer). Be that as it may, SMPTE time codes count 30 of these frames and deem that to make a second.

This causes a 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 or 29 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 in the rest of this document I try 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 frames/second (e.g. film content at 24 frames/second, or PAL video). The ‘drift’ issue is (I believe) unique to NTSC video.

2 Design Goals

What we desire is a system that allows us to associate a SMPTE time-code with some media in an RTP [RTP] 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:

1. the RTP timestamp of the start of the segment;
2. the time-code of the start of the segment;
3. the counting rate and other parameters of the time-code;
4. 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, and periodic mappings in RTCP packets.

The second requires in-band signaling within the RTP packets themselves.

Both cases are covered by this specification.

3 Signaling (setup) information

If the recipient must ever calculate time-codes based on the RTP times, then some setup information is needed. This is sent out-of-band (e.g. in SDP; the SDP mapping is TBD).

The setup information includes:
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)
3. the following booleans:
   3.1 is-NTSC-drop-frame: should the usual ‘left out numbers’ of drop-frame be applied or not?
   3.2 display-time-code-as-counter: should the display be an integer frame-count, or hh:mm:ss.fr format?
   3.3 time-code-displayed: is it intended that this time-code be displayed somehow?

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 track with the common timescale of 90000, then 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.
4 In-stream information

4.1 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. If the SMPTE time-code has been signaled as a simple counter (see above), then the 24 bits are a signed integer frame-count. If it is a ‘classic’ time-code, 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 binary-coded decimal is used (and notably, only two bits are provided for the tens digit of the frame count, so frame numbers above 39 cannot be represented).

Open question: should we allow for a full 8-byte SMPTE time-code formatted exactly as in SMPTE 12M? We are currently missing the 6 flag bits and the 8 4-bit binary groups.

4.2 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 APP packet reestablishes the mapping.

Note that the RTP time-stamp in the RTCP mapping may not match the time-stamp of any frame in the media stream. For video, it normally would; but a time-stamp 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.

4.3 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, 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), a named header extension is used to associate an RTP time to a SMPTE time-code. (See related specification of named header extensions for RTP).

There are two forms of this header extension. The first (‘implicit’) form associates the time-code with the RTP timestamp of the packet. The second (‘explicit’) form allows associates the time-code with a timestamp offset from the RTP timestamp of the packet.

The implicit form has the name "org.smpte.082005.12M.implicit", and contains solely a 24-bit time-code as defined above.

The explicit form has the name "org.smpte.082005.12M.explicit", and contains the 24-bit time-code as defined above, 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.

5 Discussion

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, using low-bandwidth (vanishingly low in the case of streams with no discontinuities), and 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; and if one puts this APP packet into all sender reports, it’s probably good enough. Then if you don’t have time-codes, you don’t have audio-video-sync either.

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

The APP packets (or the in-band codes) 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 -
send an RTCP immediately a discontinuity in the time-code is detected, and allow media-buffering in the client the chance to 'catch' the RTCP before the matching RTP packet is processed and displayed.

There is no way in this draft to detect that an RTCP packet has been lost, and that a mapping may be being used outside its intended range. The likelihood of this happening can be reduced, however, by permitting a pair of RTP times in the mapping, and defining that the mapping is only valid between those times. This only works for stored media, when look-ahead is possible, of course. It is a discussion item whether it is worthwhile.

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. It may be necessary to introduce explicit statements about the amount of buffering needed.

For trick modes, it may be desirable to signal that a given section of media has the time-code running in reverse; this would require a new sign bit in the mapping record.

6 Security Considerations

 a) SMPTE time-codes are only informative and it is hard to see security considerations from associating them with media streams.

7 IANA Considerations

 a) None, unless the domain-reversed names for the time-codes should be centrally documented somewhere.

8 RFC Editor Considerations

 a) None.

9 Full Copyright Statement

 a) Copyright (C) The Internet Society (2005).

 b) This document is subject to the rights, licenses and restrictions contained in BCP 78, and except as set forth therein, the authors retain all their rights.

 c) This document and the information contained herein are provided on an "AS IS" basis and THE CONTRIBUTOR, THE ORGANIZATION HE/SHE REPRESENTS OR IS SPONSORED BY (IF ANY), THE INTERNET SOCIETY AND THE INTERNET ENGINEERING TASK FORCE DISCLAIM ALL WARRANTIES, EXPRESS OR IMPLIED.
INCLUDING BUT NOT LIMITED TO ANY WARRANTY THAT THE USE OF THE
INFORMATION HEREBIN WILL NOT INFRINGE ANY RIGHTS OR ANY IMPLIED
WARRANTIES OF MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE.

10 Intellectual Property Notice

The IETF takes no position regarding the validity or scope of any
Intellectual Property Rights 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; nor does it represent that it has
made any independent effort to identify any such rights. Information
on the procedures with respect to rights in RFC documents can be
found in BCP 78 and BCP 79.

Copies of IPR disclosures made to the IETF Secretariat 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 on-line IPR repository at

The IETF invites any interested party to bring to its attention any
copyrights, patents or patent applications, or other proprietary
rights that may cover technology that may be required to implement
this standard. Please address the information to the IETF at ietf-
ipr@ietf.org.

Acknowledgments

Both Brian Link and John Lazzaro provided helpful comments on an
initial draft.
Authors’ Contact Information
David Singer
Apple Computer, Inc.
One Infinite Loop, MS:302-3MT
Cupertino CA 95014
USA
Email: singer@apple.com
Tel: +1 408 974 3162

6. References
[RTP]
RFC3550, STD0064, RTP: A Transport Protocol for Real-Time
Applications, H. Schulzrinne, S. Casner, R. Frederick, V. Jacobson,
July 2003

[SMPTE-12M]
SMPTE 12M-1999, Television, Audio and Film – Time and Control Code

Change History

August 2005: Draft-avt-smpte-rtp made from draft-singer-smpte-rtp;
added question on full time-code option

Dates
Written: Aug 11 2005
Expires: Feb 11 2006