                    Rapid Synchronisation of RTP Flows

Abstract

   This memo outlines how RTP sessions are synchronised, and discusses
   how rapidly such synchronisation can occur. We show that most RTP
   sessions can be synchronised immediately, but that the use of video
   switching multipoint conference units (MCUs) or large source-specific
   multicast (SSM) groups can greatly increase the synchronisation
   delay. This increase in delay can be unacceptable to some
   applications that use layered and/or multi-description codecs.

   This memo introduces three mechanisms to reduce the synchronisation
   delay for such sessions. First, it updates the RTP Control Protocol
   (RTCP) timing rules to reduce the initial synchronisation delay for
   SSM sessions. Second, a new feedback packet is defined for use with
   the extended RTP profile for RTCP-based feedback (RTP/AVPF), allowing
   video switching MCUs to rapidly request resynchronisation. Finally,
   new RTP header extensions are defined to allow rapid synchronisation
   of late joiners, and guarantee correct timestamp-based decoding order
   recovery for layered codecs in the presence of clock skew.

Status of This Memo

   This is an Internet Standards Track document.

   This document is a product of the Internet Engineering Task Force
   (IETF). It represents the consensus of the IETF community. It has
   received public review and has been approved for publication by the
   Internet Engineering Steering Group (IESG). Further information on
   Internet Standards is available in Section 2 of RFC 5741.

   Information about the current status of this document, any errata,
   and how to provide feedback on it may be obtained at
   http://www.rfc-editor.org/info/rfc6051.

Table of Contents

1. Introduction

   When using RTP to deliver multimedia content it's often necessary to
   synchronise playout of audio and video components of a presentation.
   This is achieved using information contained in RTP Control Protocol
   (RTCP) sender report (SR) packets [RFC3550]. These are sent
   periodically, and the components of a multimedia session cannot be
   synchronised until sufficient RTCP SR packets have been received for
   each RTP flow to allow the receiver to establish mappings between the
   media clock used for each RTP flow, and the common (NTP-format)
   reference clock used to establish synchronisation.

   Recently, concern has been expressed that this synchronisation delay
   is problematic for some applications, for example those using layered
   or multi-description video coding. This memo reviews the operations
   of RTP synchronisation, and describes the synchronisation delay that
   can be expected. Three backwards compatible extensions to the basic
   RTP synchronisation mechanism are proposed:

   o The RTCP transmission timing rules are relaxed for source-specific
      multicast (SSM) senders, to reduce the initial synchronisation
      latency for large SSM groups. See Section 3.1.

   o An enhancement to the extended RTP profile for RTCP-based feedback
      (RTP/AVPF) [RFC4585] is defined to allow receivers to request
      additional RTCP SR packets, providing the metadata needed to
      synchronise RTP flows. This can reduce the synchronisation delay
      when joining sessions with large RTCP reporting intervals, in the
      presence of packet loss, or when video switching MCUs are
      employed. See Section 3.2.

   o Two RTP header extensions are defined, to deliver synchronisation
      metadata in-band with RTP data packets. These extensions provide
      synchronisation metadata that is aligned with RTP data packets,
      and so eliminate the need to estimate clock skew between flows
      before synchronisation. They can also reduce the need to receive
      RTCP SR packets before flows can be synchronised, although it does
      not eliminate the need for RTCP. See Section 3.3.

   The immediate use-case for these extensions is to reduce the delay
   due to synchronisation when joining a layered video session (e.g., an
   H.264/SVC (Scalable Video Coding) session in Non-Interleaved
   Timestamp-based (NI-T) mode [AVT-RTP-SVC]). The extensions are not
   specific to layered coding, however, and can be used in any
   environment when synchronisation latency is an issue.

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 [RFC2119].

2. Synchronisation of RTP Flows

   RTP flows are synchronised by receivers based on information that is
   contained in RTCP SR packets generated by senders (specifically, the
   NTP-format timestamp and the RTP timestamp). Synchronisation
   requires that a common reference clock MUST be used to generate the
   NTP-format timestamps in a set of flows that are to be synchronised
   (i.e., when synchronising several RTP flows, the RTP timestamps for
   each flow are derived from separate, and media specific, clocks, but
   the NTP-format timestamps in the RTCP SR packets of all flows to be
   synchronised MUST be sampled from the same clock). To achieve faster
   and more accurate synchronisation, it is further RECOMMENDED that
   senders and receivers use a synchronised common NTP-format reference
   clock with common properties, especially timebase, where possible
   (recognising that this is often not possible when RTP is used outside
   of controlled environments); the means by which that common reference
   clock and its properties are signalled and distributed is outside the
   scope of this memo.

   For multimedia sessions, each type of media (e.g., audio or video) is
   sent in a separate RTP session, and the receiver associates RTP flows
   to be synchronised by means of the canonical end-point identifier
   (CNAME) item included in the RTCP Source Description (SDES) packets
   generated by the sender or signalled out of band [RFC5576]. For
   layered media, different layers can be sent in different RTP
   sessions, or using different synchronisation source (SSRC) values
   within a single RTP session; in both cases, the CNAME is used to
   identify flows to be synchronised. To ensure synchronisation, an RTP
   sender MUST therefore send periodic compound RTCP packets following
   Section 6 of RFC 3550 [RFC3550].

   The timing of these periodic compound RTCP packets will depend on the
   number of members in each RTP session, the fraction of those that are
   sending data, the session bandwidth, the configured RTCP bandwidth
   fraction, and whether the session is multicast or unicast (see
   RFC 3550, Section 6.2 for details). In summary, RTCP control traffic
   is allocated a small fraction, generally 5%, of the session
   bandwidth, and of that fraction, one quarter is allocated to active
   RTP senders, while receivers use the remaining three quarters (these
   fractions can be configured via the Session Description Protocol
   (SDP) [RFC3556]). Each member of an RTP session derives an RTCP
   reporting interval based on these fractions, whether the session is
   multicast or unicast, the number of members it has observed, and
   whether it is actively sending data or not. It then sends a compound

RTCP packet on average once per reporting interval (the actual packet transmission time is randomised in the range [0.5 ... 1.5] times the reporting interval to avoid synchronisation of reports).

A minimum reporting interval of 5 seconds is RECOMMENDED, except that the delay before sending the initial report "MAY be set to half the minimum interval to allow quicker notification that the new participant is present" [RFC3550]. Also, for unicast sessions, "the delay before sending the initial compound RTCP packet MAY be zero" [RFC3550]. In addition, for unicast sessions, and for active senders in a multicast session, the fixed minimum reporting interval MAY be scaled to "360 divided by the session bandwidth in kilobits/second. This minimum is smaller than 5 seconds for bandwidths greater than 72 kb/s" [RFC3550].

2.1. Initial Synchronisation Delay

A multimedia session comprises a set of concurrent RTP sessions among a common group of participants, using one RTP session for each media type. For example, a videoconference (which is a multimedia session) might contain an audio RTP session and a video RTP session. To allow a receiver to synchronise the components of a multimedia session, a compound RTCP packet containing an RTCP SR packet and an RTCP SDES packet with a CNAME item MUST be sent to each of the RTP sessions in the multimedia session by each sender. A receiver cannot synchronise playout across the multimedia session until such RTCP packets have been received on all of the component RTP sessions. If there is no packet loss, this gives an expected initial synchronisation delay equal to the average time taken to receive the first RTCP packet in the RTP session with the longest RTCP reporting interval. This will vary between unicast and multicast RTP sessions.

The initial synchronisation delay for layered sessions is similar to that for multimedia sessions. The layers cannot be synchronised until the RTCP SR and CNAME information has been received for each layer in the session.

2.1.1. Unicast Sessions

For unicast multimedia or layered sessions, senders SHOULD transmit an initial compound RTCP packet (containing an RTCP SR packet and an RTCP SDES packet with a CNAME item) immediately on joining each RTP session in the multimedia session. The individual RTP sessions are considered to be joined once any in-band signalling for NAT traversal

(e.g., [RFC5245]) and/or security keying (e.g., [RFC5764], [ZRTP])
has concluded, and the media path is open. This implies that the
initial RTCP packet is sent in parallel with the first data packet
following the guidance in RFC 3550 that "the delay before sending the
initial compound RTCP packet MAY be zero" and, in the absence of any
packet loss, flows can be synchronised immediately.

It is expected that NAT pinholes, firewall holes, quality-of-service,
and media security keys will have been negotiated as part of the
signalling, whether in-band or out-of-band, before the first RTCP
packet is sent. This should ensure that any middleboxes are ready to
accept traffic, and reduce the likelihood that the initial RTCP
packet will be lost.

2.1.2. Source-Specific Multicast (SSM) Sessions

For multicast sessions, the delay before sending the initial RTCP
packet, and hence the synchronisation delay, varies with the session
bandwidth and the number of members in the session. For a multicast
multimedia or layered session, the average synchronisation delay will
depend on the slowest of the component RTP sessions; this will
generally be the session with the lowest bandwidth (assuming all the
RTP sessions have the same number of members).

When sending to a multicast group, the reduced minimum RTCP reporting
interval of 360 seconds divided by the session bandwidth in kilobits
per second [RFC3550] should be used when synchronisation latency is
likely to be an issue. Also, as usual, the reporting interval is
halved for the first RTCP packet. Depending on the session bandwidth
and the number of members, this gives the average synchronisation
delays shown in Figure 1.

```
     Session| Number of receivers:
   Bandwidth| 2      3      4      5      10     100    1000   10000
       --+-----------------------------------------------------
     8 kbps| 2.73   4.10   5.47   5.47   5.47   5.47   5.47   5.47
    16 kbps| 2.50   2.50   2.73   2.73   2.73   2.73   2.73   2.73
    32 kbps| 2.50   2.50   2.50   2.50   2.50   2.50   2.50   2.50
    64 kbps| 2.50   2.50   2.50   2.50   2.50   2.50   2.50   2.50
   128 kbps| 1.41   1.41   1.41   1.41   1.41   1.41   1.41   1.41
   256 kbps| 0.70   0.70   0.70   0.70   0.70   0.70   0.70   0.70
   512 kbps| 0.35   0.35   0.35   0.35   0.35   0.35   0.35   0.35
     1 Mbps| 0.18   0.18   0.18   0.18   0.18   0.18   0.18   0.18
     2 Mbps| 0.09   0.09   0.09   0.09   0.09   0.09   0.09   0.09
     4 Mbps| 0.04   0.04   0.04   0.04   0.04   0.04   0.04   0.04
```

     Figure 1: Average Initial Synchronisation Delay in Seconds
                 for an RTP Session with 1 Sender

These numbers assume a source-specific multicast channel with a
single active sender, assuming an average RTCP packet size of
70 octets. These intervals are sufficient for lip-synchronisation
without excessive delay, but might be viewed as having too much
latency for synchronising parts of a layered video stream.

The RTCP interval is randomised in the usual manner, so the minimum
synchronisation delay will be half these intervals, and the maximum
delay will be 1.5 times these intervals. Note also that these RTCP
intervals are calculated assuming perfect knowledge of the number of
members in the session.

2.1.3. Any-Source Multicast (ASM) Sessions

For ASM sessions, the fraction of members that are senders plays an
important role, and causes more variation in average RTCP reporting
interval. This is illustrated in Figure 2 and Figure 3, which show
the RTCP reporting interval for the same session bandwidths and
receiver populations as the SSM session described in Figure 1, but
for sessions with 2 and 10 senders, respectively. It can be seen
that the initial synchronisation delay scales with the number of
senders (this is to ensure that the total RTCP traffic from all group
members does not grow without bound) and can be significantly larger
than for source-specific groups. Despite this, the initial
synchronisation time remains acceptable for lip-synchronisation in
typical small-to-medium sized group video conferencing scenarios.

Note that multi-sender groups implemented using multi-unicast with a
central RTP translator (Topo-Translator in the terminology of
[RFC5117]) or mixer (Topo-Mixer), or some forms of video switching
MCU (Topo-Video-switch-MCU) distribute RTCP packets to all members of
the group, and so scale in the same way as an ASM group with regards
to initial synchronisation latency.

```
   Session| Number of receivers:
 Bandwidth|  2     3     4     5    10    100   1000  10000
       --+----------------------------------------------
   8 kbps| 2.73  4.10  5.47  6.84 10.94 10.94 10.94 10.94
  16 kbps| 2.50  2.50  2.73  3.42  5.47  5.47  5.47  5.47
  32 kbps| 2.50  2.50  2.50  2.50  2.73  2.73  2.73  2.73
  64 kbps| 2.50  2.50  2.50  2.50  2.50  2.50  2.50  2.50
 128 kbps| 1.41  1.41  1.41  1.41  1.41  1.41  1.41  1.41
 256 kbps| 0.70  0.70  0.70  0.70  0.70  0.70  0.70  0.70
 512 kbps| 0.35  0.35  0.35  0.35  0.35  0.35  0.35  0.35
   1 Mbps| 0.18  0.18  0.18  0.18  0.18  0.18  0.18  0.18
   2 Mbps| 0.09  0.09  0.09  0.09  0.09  0.09  0.09  0.09
   4 Mbps| 0.04  0.04  0.04  0.04  0.04  0.04  0.04  0.04
```

        Figure 2: Average Initial Synchronisation Delay in Seconds
                   for an RTP Session with 2 Senders

```
   Session| Number of receivers:
 Bandwidth|  2     3     4     5    10    100   1000  10000
       --+----------------------------------------------
   8 kbps| 2.73  4.10  5.47  6.84 13.67 54.69 54.69 54.69
  16 kbps| 2.50  2.50  2.73  3.42  6.84 27.34 27.34 27.34
  32 kbps| 2.50  2.50  2.50  2.50  3.42 13.67 13.67 13.67
  64 kbps| 2.50  2.50  2.50  2.50  2.50  6.84  6.84  6.84
 128 kbps| 1.41  1.41  1.41  1.41  1.41  3.42  3.42  3.42
 256 kbps| 0.70  0.70  0.70  0.70  0.70  1.71  1.71  1.71
 512 kbps| 0.35  0.35  0.35  0.35  0.35  0.85  0.85  0.85
   1 Mbps| 0.18  0.18  0.18  0.18  0.18  0.43  0.43  0.43
   2 Mbps| 0.09  0.09  0.09  0.09  0.09  0.21  0.21  0.21
   4 Mbps| 0.04  0.04  0.04  0.04  0.04  0.11  0.11  0.11
```

        Figure 3: Average Initial Synchronisation Delay in Seconds
                  for an RTP Session with 10 Senders

2.1.4. Discussion

   For unicast sessions, the existing RTCP SR-based mechanism allows for
   immediate synchronisation, provided the initial RTCP packet is not
   lost.

   For SSM sessions, the initial synchronisation delay is sufficient for
   lip-synchronisation, but may be larger than desired for some layered
   codecs. The rationale for not sending immediate RTCP packets for
   multicast groups is to avoid implosion of requests when large numbers
   of members simultaneously join the group ("flash crowd"). This is
   not an issue for SSM senders, since there can be at most one sender,
   so it is desirable to allow SSM senders to send an immediate RTCP SR

on joining a session (as is currently allowed for unicast sessions,
which also don't suffer from the implosion problem). SSM receivers
using unicast feedback would not be allowed to send immediate RTCP.
For ASM sessions, implosion of responses is a concern, so no change
is proposed to the RTCP timing rules.

In all cases, it is possible that the initial RTCP SR packet is lost.
In this case, the receiver will not be able to synchronise the media
until the reporting interval has passed, and the next RTCP SR packet
is sent. This is undesirable. Section 3.2 defines a new RTP/AVPF
transport layer feedback message to request that an RTCP SR be
generated, allowing rapid resynchronisation in the case of packet
loss.

2.2. Synchronisation for Late Joiners

Synchronisation between RTP sessions is potentially slower for late
joiners than for participants present at the start of the session.
The reasons for this are three-fold:

1. Many of the optimisations that allow rapid transmission of RTCP SR
   packets apply only at the start of a session. This implies that a
   new participant may have to wait a complete RTCP reporting
   interval for each session before receiving the necessary data to
   synchronise media streams. This might potentially take several
   seconds, depending on the configured session bandwidth and the
   number of participants.

2. Additional synchronisation delay comes from the nature of the RTCP
   timing rules. Packets are generated on average once per reporting
   interval, but with the exact transmission times being randomised
   +/- 50% to avoid synchronisation of reports. This is important to
   avoid network congestion in multicast sessions, but does mean that
   the timing of RTCP sender reports for different RTP sessions isn't
   synchronised. Accordingly, a receiver must estimate the skew on
   the NTP-format clock in order to align RTP timestamps across
   sessions. This estimation is an essential part of an RTP
   synchronisation implementation, and can be done with high accuracy
   given sufficient reports. Collecting sufficient RTCP SR data to
   perform this estimation, however, may require reception of several
   RTCP reports, further increasing the synchronisation delay.

3. Many media codecs have the notion of periodic access points, such
   that a newly joined receiver often cannot start decoding a media
   stream until the packets corresponding to the access point have
   been received. These access points may be sent less often than
   RTCP SR packets, and so may be the limiting factor in starting
   synchronised media playout for late joiners. The RTP extension

for unicast-based rapid acquisition of multicast RTP sessions
[AVT-ACQUISITION-RTP] may be used to reduce the time taken to
receive the access points in some scenarios.

These delays are likely an issue for tuning in to an ongoing
multicast RTP session, or for video switching MCUs.

3. Reducing RTP Synchronisation Delays

Three backwards compatible RTP extensions are defined to reduce the
possible synchronisation delay: a reduced initial RTCP interval for
SSM senders, a rapid resynchronisation request message, and RTP
header extensions that can convey synchronisation metadata in-band.

3.1. Reduced Initial RTCP Interval for SSM Senders

In SSM sessions where the initial synchronisation delay is important,
the RTP sender MAY set the delay before sending the initial compound
RTCP packet to zero, and send its first RTCP packet immediately upon
joining the SSM session. This is purely a local change to the sender
that can be implemented as a configurable option. RTP receivers in
an SSM session, sending unicast RTCP feedback, MUST NOT send RTCP
packets with zero initial delay; the timing rules defined in
[RFC5760] apply unchanged to receivers.

3.2. Rapid Resynchronisation Request

The general format of an RTP/AVPF transport layer feedback message is
shown in Figure 4 (see [RFC4585] for details).

```
 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|   FMT   | PT=RTPFB=205  |           length              |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                  SSRC of packet sender                        |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                  SSRC of media source                        |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
:            Feedback Control Information (FCI)                 :
:                                                              :
```
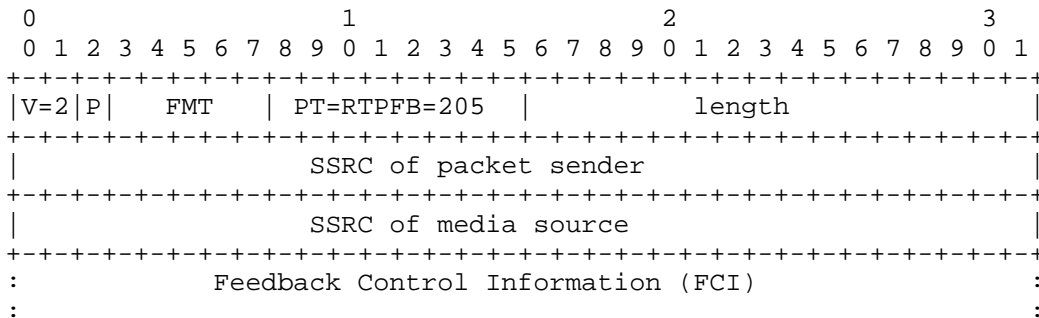
Figure 4: RTP/AVPF Transport Layer Feedback Message

   One new feedback message type, RTCP-SR-REQ, is defined with FMT = 5.
   The Feedback Control Information (FCI) part of the feedback message
   MUST be empty. The SSRC of the packet sender indicates the member
   that is unable to synchronise media streams, while the SSRC of the
   media source indicates the sender of the media it is unable to
   synchronise. The length MUST equal 2.

   If the RTP/AVPF profile [RFC4585] is in use, this feedback message
   MAY be sent by a receiver to indicate that it's unable to synchronise
   some media streams, and desires that the media source transmit an
   RTCP SR packet as soon as possible (within the constraints of the
   RTCP timing rules for early feedback). When it receives such an
   indication, a media source that understands the RTCP-SR-REQ packet
   SHOULD generate an RTCP SR packet as soon as possible while complying
   with the RTCP early feedback rules. If the use of non-compound RTCP
   [RFC5506] was previously negotiated, both the feedback request and
   the RTCP SR response may be sent as non-compound RTCP packets. The
   RTCP-SR-REQ packet MAY be repeated once per RTCP reporting interval
   if no RTCP SR packet is forthcoming. The media source may ignore
   RTCP-SR-REQ packets if its regular schedule for transmission of
   synchronisation metadata can be expected to allow the receiver to
   synchronise the media streams within a reasonable time frame.

   When using SSM sessions with unicast feedback, it is possible that
   the feedback target and media source are not co-located. If a
   feedback target receives an RTCP-SR-REQ feedback message in such a
   case, the request should be forwarded to the media source. The
   mechanism to be used for forwarding such requests is not defined
   here.

   If the feedback target provides a network management interface, it
   might be useful to provide a log of which receivers send RTCP-SR-REQ
   feedback packets and which do not, since those that do not will see
   slower stream synchronisation.

3.3. In-Band Delivery of Synchronisation Metadata

   The RTP header extension mechanism defined in [RFC5285] can be
   adapted to carry an OPTIONAL NTP-format timestamp in RTP data
   packets. If such a timestamp is included, it MUST correspond to the
   same time instant as the RTP timestamp in the packet's header, and
   MUST be derived from the same clock used to generate the NTP-format
   timestamps included in RTCP SR packets. Provided it has knowledge of
   the SSRC to CNAME mapping, either from prior receipt of an RTCP CNAME
   packet or via out-of-band signalling [RFC5576], the receiver can use
   the information provided as input to the synchronisation algorithm,
   in exactly the same way as if an additional RTCP SR packet had been
   received for the flow.

Two variants are defined for this header extension. The first
variant extends the RTP header with a 64-bit NTP-format timestamp as
defined in [RFC5905]. The second variant carries the lower 24-bit
part of the Seconds of a NTP-format timestamp and the 32 bits of the
Fraction of a NTP-format timestamp. The formats of the two variants
are shown in Figure 5 and Figure 6.

```
 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|1|  CC   |M|     PT      |       sequence number         |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+R
|                           timestamp                          |T
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+P
|           synchronisation source (SSRC) identifier           |
+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+
|      0xBE     |    0xDE       |           length=3            |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+E
|  ID-A | L=7   |   NTP timestamp format - Seconds (bit 0-23)  |x
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+t
|NTP Sec.(24-31)|   NTP timestamp format - Fraction (bit 0-23) |n
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|NTP Frc.(24-31)|   0 (pad)     |    0 (pad)     |    0 (pad)    |
+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+
|                          payload data                         |
|                             ....                              |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```
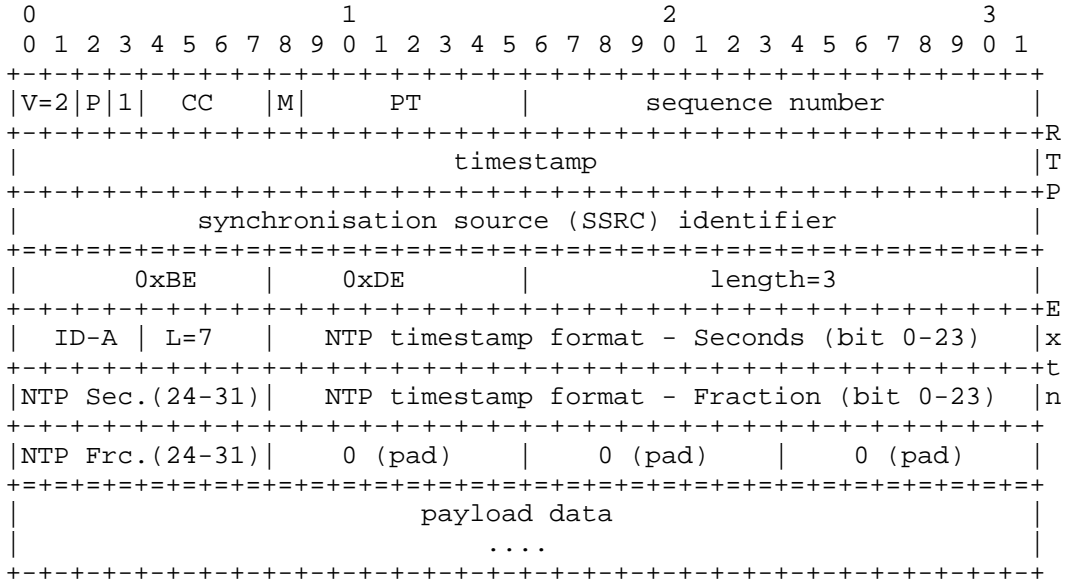
Figure 5: Variant A/64-Bit NTP RTP Header Extension

```
      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|1|   CC  |M|     PT      |       sequence number         |
     +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+R
     |                           timestamp                           |T
     +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+P
     |           synchronisation source (SSRC) identifier            |
     +=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+
     |       0xBE    |    0xDE       |           length=2             |
     +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+E
     |  ID-B |  L=6  |  NTP timestamp format - Seconds (bit 8-31)    |x
     +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+t
     |       NTP timestamp format - Fraction (bit 0-31)             |n
     +=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+
     |                          payload data                         |
     |                             ....                              |
     +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```
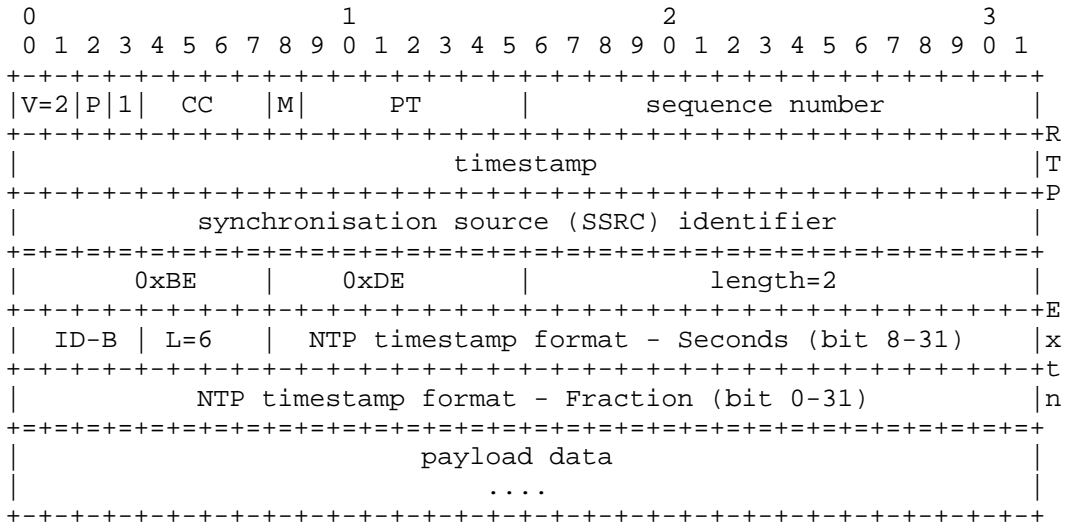
             Figure 6: Variant B/56-Bit NTP RTP Header Extension

An NTP-format timestamp MAY be included in any RTP packets the sender
chooses, but it is RECOMMENDED when performing timestamp-based
decoding order recovery for layered codecs transported in multiple
RTP flows, as further specified in Section 4.1. This header
extension SHOULD be also sent in the RTP packets corresponding to a
video random access point, and in the associated audio packets, to
allow rapid synchronisation for late joiners in multimedia sessions,
and in video switching scenarios.

   Note: The inclusion of an RTP header extension will reduce the
   efficiency of RTP header compression, if it is used. Furthermore,
   middleboxes that do not understand the header extensions may
   remove them or may not update the content according to this memo.

In all cases, irrespective of whether in-band NTP-format timestamps
are included or not, regular RTCP SR packets MUST be sent to provide
backwards compatibility with receivers that synchronise RTP flows
according to [RFC3550], and robustness in the face of middleboxes
(RTP translators) that might strip RTP header extensions. If the
Variant B/56-bit NTP RTP header extension is used, RTCP sender
reports MUST be used to derive the upper 8 bits of the Seconds for
the NTP-format timestamp.

When SDP is used, the use of the RTP header extensions defined above
MUST be indicated as specified in [RFC5285]. Therefore, the
following URIs MUST be used:

   o The URI used for signalling the use of Variant A/64-bit NTP RTP
      header extension in SDP is "urn:ietf:params:rtp-hdrext:ntp-64".

   o The URI used for signalling the use of Variant B/56-bit NTP RTP
      header extension in SDP is "urn:ietf:params:rtp-hdrext:ntp-56".

   The use of these RTP header extensions can greatly improve the user
   experience in IPTV channel surfing and in some interactive video
   conferencing scenarios. Network management tools that attempt to
   monitor the user experience may wish to log which sessions signal and
   use these extensions.

4. Application to Decoding Order Recovery in Layered Codecs

   Packets in RTP flows are often predictively coded, with a receiver
   having to arrange the packets into a particular order before it can
   decode the media data. Depending on the payload format, the decoding
   order might be explicitly specified as a field in the RTP payload
   header, or the receiver might decode the packets in order of their
   RTP timestamps. If a layered encoding is used, where the media data
   is split across several RTP flows, then it is often necessary to
   exactly synchronise the RTP flows comprising the different layers
   before layers other than the base layer can be decoded. Examples of
   such layered encodings are H.264 SVC in NI-T mode [AVT-RTP-SVC] and
   MPEG surround multi-channel audio [RFC5691]. As described in
   Section 2, such synchronisation is possible in RTP, but can be
   difficult to perform rapidly. Below, we describe how the extensions
   defined in Section 3.3 can be used to synchronise layered flows, and
   provide a common timestamp-based decoding order.

4.1. In-Band Synchronisation for Decoding Order Recovery

   When a layered, multi-description, or multi-view codec is used, with
   the different components of the media being transferred on separate
   RTP flows, the RTP sender SHOULD use periodic synchronous in-band
   delivery of synchronisation metadata to allow receivers to rapidly
   and accurately synchronise the separate components of the layered
   media flow. There are three parts to this:

   o The sender must negotiate the use of the RTP header extensions
      described in Section 3.3, and must periodically and synchronously
      insert such header extensions into all the RTP flows forming the
      separate components of the layered, multi-description, or multi-
      view flow.

   o Synchronous insertion requires that the sender insert these RTP
      header extensions into packets corresponding to exactly the same
      sampling instant in all the flows. Since the header extensions

for each flow are inserted at exactly the same sampling instant,
they will have identical NTP-format timestamps, hence allowing
receivers to exactly align the RTP timestamps for the component
flows. This may require the insertion of extra data packets into
some of the component RTP flows, if some component flows contain
packets for sampling instants that do not exist in other flows
(for example, a layered video codec, where the layers have
differing frame rates).

o The frequency with which the sender inserts the header extensions
   will directly correspond to the synchronisation latency, with more
   frequent insertion leading to higher per-flow overheads, but lower
   synchronisation latency. It is RECOMMENDED that the sender insert
   the header extensions synchronously into all component RTP flows
   at least once per random access point of the media, but they MAY
   be inserted more often.

The sender MUST continue to send periodic RTCP reports including SR
packets, and MUST ensure the RTP timestamp to NTP-format timestamp
mapping in the RTCP SR packets is consistent with that used in the
RTP header extensions. Receivers should use both the information
contained in RTCP SR packets and the in-band mapping of RTP and NTP-
format timestamps as input to the synchronisation process, but it is
RECOMMENDED that receivers sanity check the mappings received and
discard outliers, to provide robustness against invalid data (one
might think it more likely that the RTCP SR mappings are invalid,
since they are sent at irregular times and subject to skew, but the
presence of broken RTP translators could also corrupt the timestamps
in the RTP header extension; receivers need to cope with both types
of failure).

4.2. Timestamp-Based Decoding Order Recovery

Once a receiver has synchronised the components of a layered, multi-
description, or multi-view flow using the RTP header extensions as
described in Section 4.1, it may then derive a decoding order based
on the synchronised timestamps as follows (or it may use information
in the RTP payload header to derive the decoding order, if present
and desired).

There may be explicit dependencies between the component flows of a
layered, multi-description, or multi-view flow. For example, it is
common for layered flows to be arranged in a hierarchy, where flows
from "higher" layers cannot be decoded until the corresponding data
in "lower" layer flows has been received and decoded. If such a
decoding hierarchy exists, it MUST be signalled out of band, for
example using [RFC5583] when SDP signalling is used.

Each component RTP flow MUST contain packets corresponding to all the
sampling instants of the RTP flows on which it depends. If such
packets are not naturally present in the RTP flow, the sender MUST
generate additional packets as necessary in order to satisfy this
rule. The format of these packets depends on the payload format
used. For H.264 SVC, the Empty Network Abstraction Layer (NAL) unit
packet [AVT-RTP-SVC] should be used. Flows may also include packets
corresponding to additional sampling instants that are not present in
the flows on which they depend.

The receiver should decode the packets in all the component RTP flows
as follows:

o For each RTP packet in each flow, use the mapping contained in the
   RTP header extensions and RTCP SR packets to derive the NTP-format
   timestamp corresponding to its RTP timestamp.

o Group together RTP data packets from all component flows that have
   identical calculated NTP-format timestamps.

o Processing groups in order of ascending NTP-format timestamps,
   decode the RTP packets in each group according to the signalled
   RTP flow decoding hierarchy. That is, pass the RTP packet data
   from the flow on which all other flows depend to the decoder
   first, then that from the next dependent flow, and so on. The
   decoding order of the RTP flow hierarchy may be indicated by
   mechanisms defined in [RFC5583] or by some other means.

Note that the decoding order will not necessarily match the packet
transmission order. The receiver will need to buffer packets for a
codec-dependent amount of time in order for all necessary packets to
arrive to allow decoding.

4.3. Example

The example shown in Figure 7 refers to three RTP flows A, B, and C,
containing a layered, a multi-view, or a multi-description media
stream. In the example, the dependency signalling as defined in
[RFC5583] indicates that flow A is the lowest RTP flow. Flow B is
the next higher RTP flow and depends on A. Flow C is the highest of
the three RTP flows and depends on both A and B. A media coding
structure is used that results in video access units (i.e., coded
video frames) present in higher flows but not present in all lower
flows. Flow A has the lowest frame rate. Flows B and C have the
same frame rate, which is higher than that of Flow A. The figure
shows the full video access units with their corresponding RTP
timestamps "(x)". The video access units are already re-ordered
according to their RTP sequence number order. The figure indicates

the received video access unit part in decoding order within each RTP
flow, as well as the associated NTP media timestamps ("TS[..]"). As
shown in the figure, these timestamps may be derived using the
NTP-format timestamp provided in the RTCP sender reports as indicated
by the timestamp in "{x}", or derived directly from the NTP timestamp
contained in the RTP header extensions as indicated by the timestamp
in "<x>". Note that the timestamps are not in increasing order
since, in this example, the decoding order is different from the
output/presentation order.

The decoding order recovery process first advances to the video
access unit parts associated with the first available synchronous
insertion of the NTP timestamp into RTP header extensions at NTP
media timestamp TS=[8]. The receiver starts in the highest RTP
flow C and removes/ignores all preceding video access unit parts (in
decoding order) to video access unit parts with TS=[8] in each of the
de-jittering buffers of RTP flows A, B, and C. Then, starting from
flow C, the first media timestamp available in decoding order
(TS=[8]) is selected, and video access unit parts starting from RTP
flow A, and flows B and C are placed in order of the RTP flow
dependency as indicated by mechanisms defined in [RFC5583] (in the
example for TS[8]: first flow B and then flow C into the video access
unit AU(TS[8]) associated with NTP media timestamp TS=[8]). Then the
next media timestamp TS=[6] (RTP timestamp=(4)) in order of
appearance in the highest RTP flow C is processed, and the process
described above is repeated. Note that there may be video access
units with no video access unit parts present, e.g., in the lowest
RTP flow A (see, e.g., TS=[5]). The decoding order recovery process
could also be started after an RTP sender report containing the
mapping between the RTP timestamp and the NTP-format timestamp
(indicated as timestamps "(x){y}") has been received, assuming that
there is no clock skew in the source used for the NTP-format
timestamp generation.

```
   C:-(0)----(2)----(7)<8>--(5)----(4)----(6)-----(11)----(9){10}-
      |       |       |       |       |       |       |       |
   B:-(3)----(5)---(10)<8>--(8)----(7)----(9){7}--(14)----(12)----
              |       |       |       |               |       |
   A:--------------(3)<8>--(1)------------------(7){12}-(5)-----

   ------------------------------------------decoding/transmission order->
   TS:[1]    [3]     [8]=<8> [6]     [5]     [7]     [12]    [10]
```

   Key:

   A, B, C   - RTP flows

   Integer values in "()" - video access unit with its RTP timestamp as
                            indicated in its RTP packet.

   "|"   - indicates the corresponding parts of the
                            same video access unit AU(TS[..]) in the
                            RTP flows.

   Integer values in "[]" - NTP media timestamp TS, sampling time
                            as derived from the NTP timestamp
                            associated with the video access unit
                            AU(TS[..]), consisting of video access unit
                            parts in the flows above.

   Integer values in "<>" - NTP media timestamp TS as directly
                            taken from the NTP RTP header extensions.

   Integer values in "{}" - NTP media timestamp TS as provided in the
                            RTCP sender reports.

                   Figure 7: Example of a Layered RTP Stream

5. Security Considerations

   The security considerations of the RTP specification [RFC3550], the
   extended RTP profile for RTCP-based feedback [RFC4585], and the
   general mechanism for RTP header extensions [RFC5285] apply.

   The RTP header extensions defined in Section 3.3 include an NTP-
   format timestamp. When an RTP session using this header extension is
   protected by the Secure RTP (SRTP) framework [RFC3711], that header
   extension is not part of the encrypted portion of the RTP data
   packets or RTCP control packets; however, these NTP-format timestamps
   are encrypted when using SRTP without this header extension. This is
   a minor information leak, but one that is not believed to be

significant. The inclusion of this header extension will also reduce
the efficiency of RTP header compression, if it is used.
Furthermore, middleboxes that do not understand the header extensions
may remove them or may not update the content according to this memo.

6. IANA Considerations

The IANA has registered one new value in the table of FMT Values for
RTPFB Payload Types [RFC4585] as follows:

    Name: RTCP-SR-REQ
    Long name: RTCP Rapid Resynchronisation Request
    Value: 5
    Reference: RFC 6051

The IANA has also registered two new RTP Compact Header Extensions
[RFC5285], according to the following:

    Extension URI: urn:ietf:params:rtp-hdrext:ntp-64
    Description: Synchronisation metadata: 64-bit timestamp format
    Contact: Thomas Schierl <ts@thomas-schierl.de>
                    IETF Audio/Video Transport Working Group
    Reference: RFC 6051

    Extension URI: urn:ietf:params:rtp-hdrext:ntp-56
    Description: Synchronisation metadata: 56-bit timestamp format
    Contact: Thomas Schierl <ts@thomas-schierl.de>
                    IETF Audio/Video Transport Working Group
    Reference: RFC 6051

7. Acknowledgements

This memo has benefited from discussions with numerous members of the
IETF AVT working group, including Jonathan Lennox, Magnus Westerlund,
Randell Jesup, Gerard Babonneau, Ingemar Johansson, Ali C. Begen,
Ye-Kui Wang, Roni Even, Michael Dolan, Art Allison, and Stefan
Doehla. The RTP header extension format of Variant A in Section 3.3
was suggested by Dave Singer, matching a similar mechanism specified
by the Internet Streaming Media Alliance (ISMA).

8. References

8.1. Normative References

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

   [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.

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

   [RFC5506]   Johansson, I. and M. Westerlund, "Support for
                Reduced-Size Real-Time Transport Control Protocol (RTCP):
                Opportunities and Consequences", RFC 5506, April 2009.

   [RFC5583]   Schierl, T. and S. Wenger, "Signaling Media Decoding
                Dependency in the Session Description Protocol (SDP)",
                RFC 5583, July 2009.

   [RFC5760]   Ott, J., Chesterfield, J., and E. Schooler, "RTP Control
                Protocol (RTCP) Extensions for Single-Source Multicast
                Sessions with Unicast Feedback", RFC 5760, February 2010.

   [RFC5905]   Mills, D., Martin, J., Burbank, J., and W. Kasch,
                "Network Time Protocol Version 4: Protocol and Algorithms
                Specification", RFC 5905, June 2010.

8.2. Informative References

   [AVT-ACQUISITION-RTP]
                VerSteeg, B., Begen, A., VanCaenegem, T., and Z. Vax,
                "Unicast-Based Rapid Acquisition of Multicast RTP
                Sessions", Work in Progress, October 2010.

   [AVT-RTP-SVC]
                Wenger, S., Wang, Y., Schierl, T., and A. Eleftheriadis,
                "RTP Payload Format for SVC Video Coding", Work
                in Progress, October 2010.

   [RFC3556]   Casner, S., "Session Description Protocol (SDP) Bandwidth
               Modifiers for RTP Control Protocol (RTCP) Bandwidth",
               RFC 3556, July 2003.

   [RFC3711]   Baugher, M., McGrew, D., Naslund, M., Carrara, E., and K.
               Norrman, "The Secure Real-time Transport Protocol
               (SRTP)", RFC 3711, March 2004.

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

   [RFC5245]   Rosenberg, J., "Interactive Connectivity Establishment
               (ICE): A Protocol for Network Address Translator (NAT)
               Traversal for Offer/Answer Protocols", RFC 5245,
               April 2010.

   [RFC5576]   Lennox, J., Ott, J., and T. Schierl, "Source-Specific
               Media Attributes in the Session Description Protocol
               (SDP)", RFC 5576, June 2009.

   [RFC5691]   de Bont, F., Doehla, S., Schmidt, M., and R.
               Sperschneider, "RTP Payload Format for Elementary Streams
               with MPEG Surround Multi-Channel Audio", RFC 5691,
               October 2009.

   [RFC5764]   McGrew, D. and E. Rescorla, "Datagram Transport Layer
               Security (DTLS) Extension to Establish Keys for the
               Secure Real-time Transport Protocol (SRTP)", RFC 5764,
               May 2010.

   [ZRTP]   Zimmermann, P., Johnston, A., Ed., and J. Callas, "ZRTP:
               Media Path Key Agreement for Unicast Secure RTP", Work
               in Progress, June 2010.

Authors' Addresses

    Colin Perkins
    University of Glasgow
    School of Computing Science
    Glasgow G12 8QQ
    UK

    EMail: csp@csperkins.org

    Thomas Schierl
    Fraunhofer HHI
    Einsteinufer 37
    D-10587 Berlin
    Germany

    Phone: +49-30-31002-227
    EMail: ts@thomas-schierl.de