Network Working Group
Internet-Draft
Updates: 3550 (if approved)
Intended status: Standards Track
Expires: November 6, 2010

C. Perkins
University of Glasgow
T. Schierl
Fraunhofer HHI

May 5, 2010

Rapid Synchronisation of RTP Flows
draft-ietf-avt-rapid-rtp-sync-10.txt

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 Internet-Draft is submitted in full conformance with the provisions of BCP 78 and BCP 79.

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF). Note that other groups may also distribute working documents as Internet-Drafts. The list of current Internet-Drafts is at http://datatracker.ietf.org/drafts/current/.

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."

This Internet-Draft will expire on November 6, 2010.

Copyright Notice

Copyright (c) 2010 IETF Trust and the persons identified as the document authors. All rights reserved.

This document is subject to BCP 78 and the IETF Trust's Legal Provisions Relating to IETF Documents (http://trustee.ietf.org/license-info) in effect on the date of publication of this document. Please review these documents carefully, as they describe your rights and restrictions with respect to this document. Code Components extracted from this document must include Simplified BSD License text as described in Section 4.e of the Trust Legal Provisions and are provided without warranty as described in the Simplified BSD License.

Table of Contents

1. Introduction 3
2. Synchronisation of RTP Flows 4
2.1. Initial Synchronisation Delay 5
2.1.1. Unicast Sessions 5
2.1.2. Source Specific Multicast (SSM) Sessions 6
2.1.3. Any Source Multicast (ASM) Sessions 7
2.1.4. Discussion 8
2.2. Synchronisation for Late Joiners 8
3. Reducing RTP Synchronisation Delays 9
3.1. Reduced Initial RTCP Interval for SSM Senders 9
3.2. Rapid Resynchronisation Request 10
3.3. In-band Delivery of Synchronisation Metadata 11
4. Application to Decoding Order Recovery in Layered Codecs . . . 13
4.1. In-band Synchronisation for Decoding Order Recovery . . . 14
4.2. Timestamp based decoding order recovery 15
4.3. Example 16
5. Security Considerations 17
6. IANA Considerations 17
7. Acknowledgements 18
8. References 18
8.1. Normative References 18
8.2. Informative References 19
Authors' Addresses 20

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 [1]. 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:

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

- An enhancement to the Extended RTP Profile for RTCP-based Feedback (RTP/AVPF) [2] 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.

- 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 session in NI-T mode [9]). 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 [3].

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 [10]. For layered media, different layers can be sent in different RTP sessions, or using different 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 [1].

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 SDP [11]). 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" [1]. Also, for unicast sessions, "the delay before sending the initial compound RTCP packet MAY be zero" [1]. 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." [1]

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. [12]) and/or security keying (e.g. [13],[14]) 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.

Note that NAT pinholes, firewall holes, quality-of-service, and media security keys should 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 [1] 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 Bandwidth	Number of receivers: 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 [15]) 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 Bandwidth	Number of receivers: 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 Bandwidth	Number of receivers: 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 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 SR 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 [16] 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. RTP receivers in an SSM session, sending unicast RTCP feedback, MUST NOT send RTCP packets with zero initial delay; the timing rules defined in [4] 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 [2] for details).

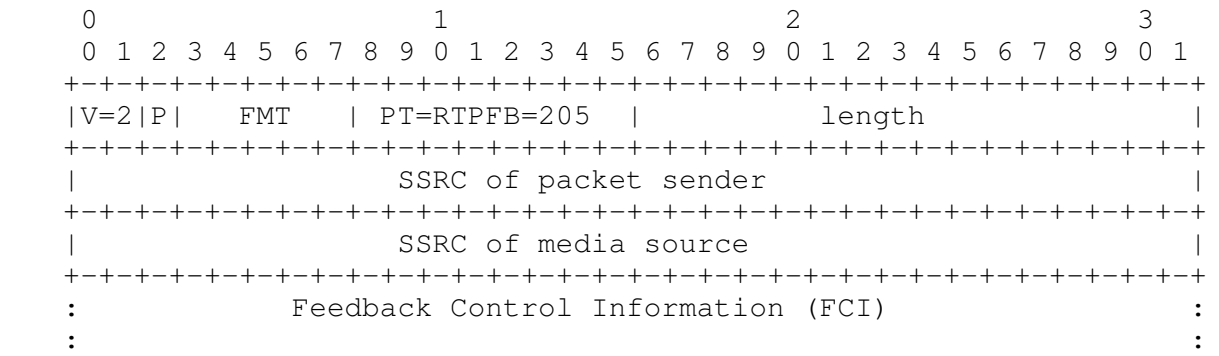


Figure 4: RTP/AVP Transport Layer Feedback Message

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 packet sender indicates the member that is unable to synchronise media streams, while the SSRC of media source indicates the sender of the media it is unable to synchronise. The length MUST equal 2.

If the RTP/AVPF profile [2] 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 [5] 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, 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.

3.3. In-band Delivery of Synchronisation Metadata

The RTP header extension mechanism defined in [6] can be adopted 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 [10], 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 was 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 timestamp format timestamp as defined in [7]. The second variant carries the lower 24 bit part of the Seconds of a NTP timestamp format timestamp and the 32 bit of the Fraction of a NTP timestamp format timestamp. The formats of the two variants are shown in Figure 5 and Figure 6.

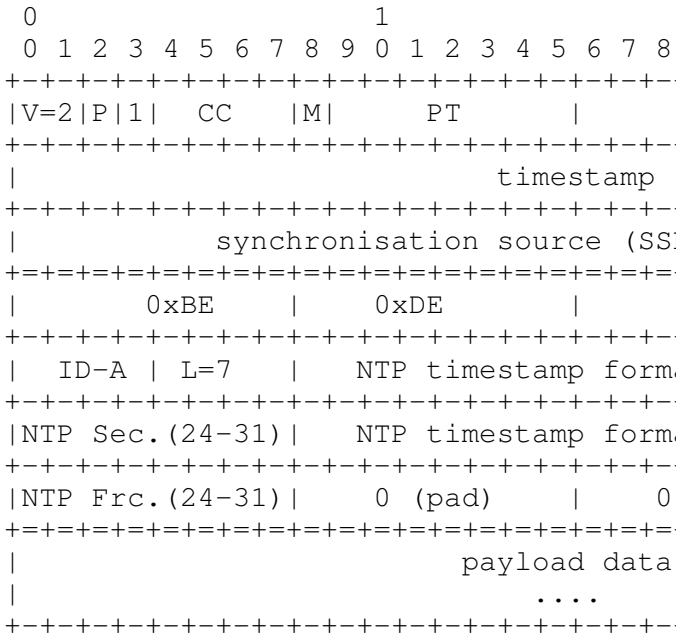


Figure 5: Variant A/64-bit NTP RTP header extension

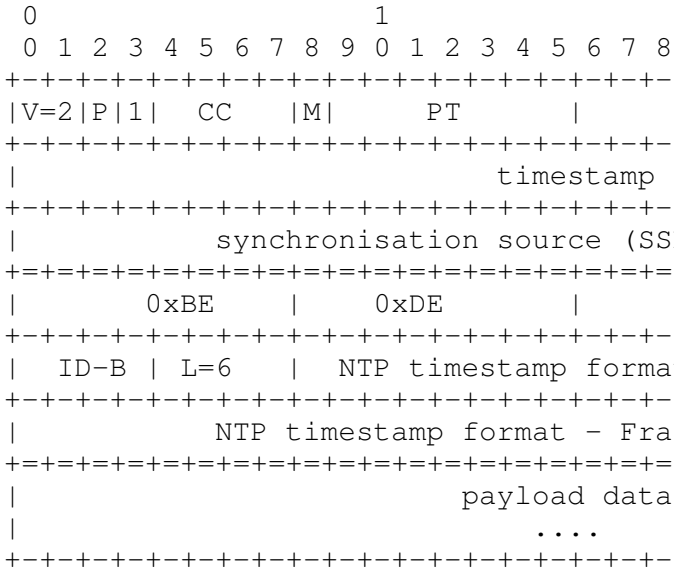


Figure 6: Variant B/56-bit NTP RTP header extension

An NTP timestamp format timestamp MAY be included on 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 on the RTP packets corresponding to a video random access point, and on 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, middle boxes which 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 timestamp 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 [1], 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 bit of the Seconds for the NTP timestamp format timestamp.

When the SDP is used, the use of the RTP header extensions defined above MUST be indicated as specified in [6]. Therefore the following URIs MUST be used:

- 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".

- 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".

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 [9] and MPEG surround multi-channel audio [17]. As described in Section 2, such synchronisation is possible in RTP, but can be difficult to perform rapidly. In the following, 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:

- 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.

- Synchronous insertion requires the sender insert these RTP header extensions into packets corresponding to the exact 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).

- 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-flows 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 [8] 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 NAL unit packet [9] 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:

- 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.

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

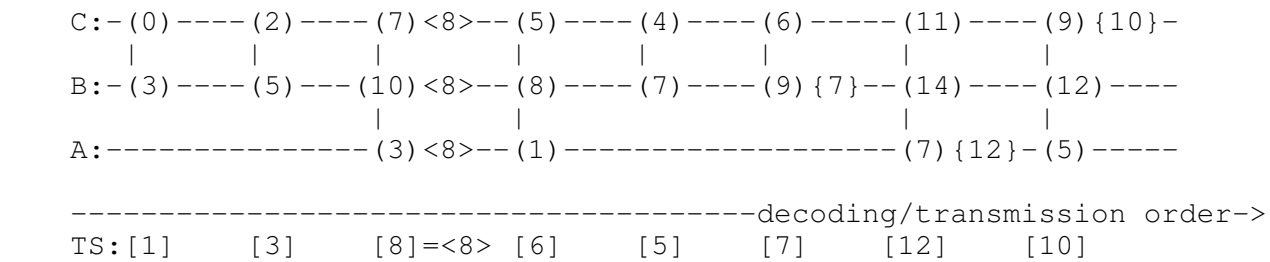
- Processing groups in order of ascending NTP-format timestamp, decode the RTP packets in each group according to the signalled RTP flow decoding hierarchy. That is, pass the RTP packet data from the flow on which all other flows depend to the decoder first, then that from the next dependent flow, and so on. The decoding order of the RTP flow hierarchy may be indicated by mechanisms defined in [8] 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 3 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 [8] indicates that flow A is the lowest RTP flow, B is the first higher RTP flow and depends on A, and C is the second higher RTP flow corresponding to flow A and depends on A and B. A media coding structure is used that results in samples present in higher flows but not present in all lower flows. Flow A has the lowest frame rate and Flow B and C have the same but higher frame rate. The figure shows the full video samples with their corresponding RTP timestamps "(x)". The video samples are already re-ordered according to their RTP sequence number order. The figure indicates for the received sample in decoding order within each RTP flow, as well as the associated NTP media timestamps ("TS[..]"). These timestamps may be derived using the NTP format timestamp provided in the RTCP sender reports or as shown in the figure directly from the NTP timestamp contained in the RTP header extensions as indicate 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 process first proceeds to the sample parts associated with the first available synchronous insertion of NTP timestamp into RTP header extensions at NTP media timestamp TS=[8] and starts in the highest RTP flow C and removes/ignores all preceding sample parts (in decoding order) to sample 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 sample parts starting from RTP flow A, and flow B and C are placed in order of the RTP flow dependency as indicated by mechanisms defined in [8] (in the example for TS[8]: first flow B and then flow C into the video sample VS(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 samples with no sample parts present, e.g., in the lowest RTP flow A (see, e.g., TS=[5]). The decoding order recovery process could be also started after receiving all RTP sender reports RTP timestamp to NTP-format timestamp mapping (indicated as timestamps "(x){y}") assuming that there is no clock skew in the source used for the NTP-format timestamp generation.



Key:
A, B, C - RTP flows
Integer values in "()"- video sample with its RTP timestamp as indicated in its RTP packet.
"|" - indicates corresponding samples / parts of sample of the same video sample VS(TS[..]) in the RTP flows.
Integer values in "[]"- NTP media timestamp TS, sampling time as derived from the NTP timestamp associated with the video sample AU(TS[..]), consisting of sample 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.

5. Security Considerations

The security considerations of the RTP specification [1], the Extended RTP profile for RTCP-Based Feedback [2], and the General Mechanism for RTP Header Extensions [6] 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 framework [18], 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.

6. IANA Considerations

NOTE TO RFC EDITOR: Please replace "RFC XXXX" in the following with the RFC number assigned to this memo, and delete this note.

The IANA is requested to register one new value in the table of FMT

Values for RTPFB Payload Types [2] as follows:

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

The IANA is also requested to register two new RTP Compact Header Extensions [6], according to the following:

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

   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 XXXX
```

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 ISMA.

8. References

8.1. Normative References

[1] Schulzrinne, H., Casner, S., Frederick, R., and V. Jacobson, "RTP: A Transport Protocol for Real-Time Applications", STD 64, RFC 3550, July 2003.

[2] 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.

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

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

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

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

[7] Mills, D., "Network Time Protocol (Version 3) Specification, Implementation", RFC 1305, March 1992.

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

8.2. Informative References

[9] Wenger, S., Wang, Y., Schierl, T., and A. Eleftheriadis, "RTP Payload Format for SVC Video", draft-ietf-avt-rtp-svc-21 (work in progress), April 2010.

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

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

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

[13] McGrew, D. and E. Rescorla, "Datagram Transport Layer Security (DTLS) Extension to Establish Keys for Secure Real-time Transport Protocol (SRTP)", draft-ietf-avt-dtls-srtp-07 (work in progress), February 2009.

[14] Zimmermann, P., Johnston, A., and J. Callas, "ZRTP: Media Path Key Agreement for Unicast Secure RTP", draft-zimmermann-avt-zrtp-18 (work in progress), April 2010.

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

[16] Steeg, B., Begen, A., Caenegem, T., and Z. Vax, "Unicast-Based Rapid Acquisition of Multicast RTP Sessions", draft-ietf-avt-rapid-acquisition-for-rtp-09 (work in progress), April 2010.

[17] 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.

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

Authors' Addresses

Colin Perkins
University of Glasgow
Department 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