AVTCORE
Internet-Draft
Updates: 3550, 4585 (if approved)
Intended status: Standards Track
Expires: August 18, 2014

J. Lennox
Vidyo
M. Westerlund
Ericsson
Q. Wu
Huawei
C. Perkins
University of Glasgow

February 14, 2014

Sending Multiple Media Streams in a Single RTP Session
draft-ietf-avtcore-rtp-multi-stream-03

Abstract

This document expands and clarifies the behavior of the Real-Time Transport Protocol (RTP) endpoints when they are using multiple synchronization sources (SSRCs), e.g. for sending multiple media streams, in a single RTP session. In particular, issues involving RTCP Control Protocol (RTCP) messages are described.

This document updates RFC 3550 in regards to handling of multiple SSRCs per endpoint in RTP sessions. It also updates RFC 4585 to clarify the calculation of the timeout of SSRCs and the inclusion of feeback messages.

Table of Contents

1. Introduction

At the time The Real-Time Transport Protocol (RTP) [RFC3550] was originally written, and for quite some time after, endpoints in RTP sessions typically only transmitted a single media stream, and thus used a single synchronization source (SSRC) per RTP session, where separate RTP sessions were typically used for each distinct media type.

Recently, however, a number of scenarios have emerged (discussed further in Section 3) in which endpoints wish to send multiple RTP media streams, distinguished by distinct RTP synchronization source (SSRC) identifiers, in a single RTP session. Although RTP's initial design did consider such scenarios, the specification was not consistently written with such use cases in mind. The specifications are thus somewhat unclear.

The purpose of this document is to expand and clarify [RFC3550]'s language for these use cases. The authors believe this does not result in any major normative changes to the RTP specification, however this document defines how the RTP specification is to be interpreted. In these cases, this document updates RFC3550. The document also updates RFC 4585 in regards to the timeout of inactive SSRCs as specificed in Section 6.1 as well as clarifying the inclusion of feedback messages.

The document starts with terminology and some use cases where multiple sources will occur. This is followed by RTP and RTCP recommendations to resolve issues. Next are security considerations and remaining open issues.

2. Terminology

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "NOT RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in RFC 2119 [RFC2119] and indicate requirement levels for compliant implementations.

3. Use Cases For Multi-Stream Endpoints

This section discusses several use cases that have motivated the development of endpoints that sends RTP data using multiple SSRCs in a single RTP session.

3.1. Multiple-Capturer Endpoints

The most straightforward motivation for an endpoint to send multiple RTP streams in a session is the scenario where an endpoint has multiple capture devices, and thus media sources, of the same media type and characteristics. For example, telepresence endpoints, of the type described by the CLUE Telepresence Framework [I-D.ietf-clue-framework], often have multiple cameras or microphones covering various areas of a room.

3.2. Multi-Media Sessions

Recent work has been done in RTP [I-D.ietf-avtcore-multi-media-rtp-session] and SDP [I-D.ietf-mmusic-sdp-bundle-negotiation] to update RTP's historical assumption that media sources of different media types would always be sent on different RTP sessions. In this work, a single endpoint's audio and video RTP media streams (for example) are instead sent in a single RTP session.

3.3. Multi-Stream Mixers

There are several RTP topologies which can involve a central device that itself generates multiple RTP media streams in a session.

One example is a mixer providing centralized compositing for a multi-capture scenario like that described in Section 3.1. In this case, the centralized node is behaving much like a multi-capturer endpoint, generating several similar and related sources.

More complicated is the Selective Forwarding Middlebox, see Section 3.7 of [I-D.ietf-avtcore-rtp-topologies-update]. This is a middlebox that receives media streams from several endpoints, and then selectively forwards modified versions of some of the streams toward the other endpoints it is connected to. Toward one destination, a separate media source appears in the session for every other source connected to the middlebox, "projected" from the original streams, but at any given time many of them can appear to be inactive (and thus are receivers, not senders, in RTP). This sort of device is closer to being an RTP mixer than an RTP translator, in that it terminates RTCP reporting about the mixed streams, and it can re-write SSRCs, timestamps, and sequence numbers, as well as the

contents of the RTP payloads, and can turn sources on and off at will without appearing to be generating packet loss. Each projected stream will typically preserve its original RTCP source description (SDES) information.

3.4. Multiple SSRCs for a Single Media Source

There are also several cases where a single media source results in the usage of multiple SSRCs within the same RTP session. Transport robustification tools like RTP Retransmission [RFC4588] result in multiple SSRCs, one with source data, and another with the repair data. Scalable encoders and their RTP payload foramts, like H.264's extension for Scalable Video Coding(SVC) [RFC6190] can be transmitted in a configuration where the scalable layers are distributed over multiple SSRCs within the same session, to enable RTP packet stream level (SSRC) selection and routing in conferencing middleboxes.

4. Multi-Stream Endpoint RTP Media Recommendations

While an endpoint MUST (of course) stay within its share of the available session bandwidth, as determined by signalling and congestion control, this need not be applied independently or uniformly to each media stream and its SSRCs. In particular, session bandwidth MAY be reallocated among an endpoint's SSRCs, for example by varying the bandwidth use of a variable-rate codec, or changing the codec used by the media stream, up to the constraints of the session's negotiated (or declared) codecs. This includes enabling or disabling media streams and their redundancy streams as more or less bandwidth becomes available.

5. Multi-Stream Endpoint RTCP Recommendations

This section contains a number of different RTCP clarifications or recommendations that enables more efficient and simpler behavior without loss of functionality.

The RTP Control Protocol (RTCP) is defined in Section 6 of [RFC3550], but it is largely documented in terms of "participants". In many cases, the specification's recommendations for "participants" are to be interpreted as applying to individual SSRCs, rather than to endpoints. This section describes several concrete cases where this applies.

5.1. RTCP Reporting Requirement

For each of an endpoint's SSRCs, whether or not they are currently sending media, SR/RR and SDES packets MUST be sent at least once per RTCP report interval. (For discussion of the content of SR or RR

packets' reception statistic reports, see [I-D.ietf-avtcore-rtp-multi-stream-optimisation].)

5.2. Initial Reporting Interval

When a new SSRC is added to a unicast session, the sentence in [RFC3550]'s Section 6.2 applies: "For unicast sessions ... the delay before sending the initial compound RTCP packet MAY be zero." This applies to individual SSRCs as well. Thus, endpoints MAY send an initial RTCP packet for an SSRC immediately upon adding it to a unicast session.

This allowance also applies, as written, when initially joining a unicast session. However, in this case some caution needs to be exercised if the end-point or mixer has a large number of sources (SSRCs) as this can create a significant burst. How big an issue this is depends on the number of sources for which the initial SR or RR packets and Session Description CNAME items are to be sent, in relation to the RTCP bandwidth.

(tbd: Maybe some recommendation here? The aim in restricting this to unicast sessions was to avoid this burst of traffic, which the usual RTCP timing and reconsideration rules will prevent.)

5.3. Compound RTCP Packets

Section 6.1 in [RFC3550] gives the following advice to RTP translators and mixers:

> "It is RECOMMENDED that translators and mixers combine individual RTCP packets from the multiple sources they are forwarding into one compound packet whenever feasible in order to amortize the packet overhead (see Section 7). An example RTCP compound packet as might be produced by a mixer is shown in Fig. 1. If the overall length of a compound packet would exceed the MTU of the network path, it SHOULD be segmented into multiple shorter compound packets to be transmitted in separate packets of the underlying protocol. This does not impair the RTCP bandwidth estimation because each compound packet represents at least one distinct participant. Note that each of the compound packets MUST begin with an SR or RR packet."

> Note: To avoid confusion, an RTCP packet is an individual item, such as a Sender Report (SR), Receiver Report (RR), Source Description (SDES), Goodbye (BYE), Application Defined (APP), Feedback [RFC4585] or Extended Report (XR) [RFC3611] packet. A compound packet is the combination of two or more such RTCP

packets where the first packet has to be an SR or an RR packet, and which contains a SDES packet containing an CNAME item.

The above results in compound RTCP packets that contain multiple SR or RR packets from different sources (SSRCs) as well as any of the other packet types. There are no restrictions on the order in which the packets can occur within the compound packet, except the regular compound rule, i.e., starting with an SR or RR.

This advice applies to multi-media-stream endpoints as well, with the same restrictions and considerations. (Note, however, that the last sentence does not apply to AVPF [RFC4585] or SAVPF [RFC5124] feedback packets if Reduced-Size RTCP [RFC5506] is in use.)

5.3.1. Maintaining AVG_RTCP_SIZE

When multiple local SSRCs are sending their RTCP packets in the same compound packet, this obviously results in larger RTCP compound packets. This will have an affect on the value of the average RTCP packet size metering (avg_rtcp_size) that is done for the purpose of RTCP transmission scheduling calculation. This section discusses the impact of this and provide recommendations with how to deal with it.

This section will use the concept of an 'RTCP Compound Packet' to represent not just proper RTCP compound packets, i.e. ones that start with an SR or RR RTCP packet and include at least one SDES CNAME item. For the purpose of the below calculation, other valid lower layer datagram units an RTCP implementation can send or receive, independently if they are an aggregate or not of RTCP packets are also considered. This especially includes Reduced-Size RTCP packets [RFC5506].

The RTCP packet scheduling algorithm that is defined in RTP [RFC3550] deals with individual SSRCs. These SSRCs transmit their set of RTCP packets at each scheduled interval. Thus, to maintain this per-SSRC property of the scheduling, the avg_rtcp_size needs to be updated with per-SSRC average RTCP compound packet sizes. The avg_rtcp_size value SHALL be updated for each received or sent RTCP compound packet with the total size (including packet overhead such as IP/UDP) divided by the number of reporting SSRCs. The number of reporting SSRCs SHALL be determined by counting the number of different SSRCs that are the source of Sender Report (SR) or Receiver Report (RR) RTCP packets within the compound. A non-compound RTCP packet, i.e. it contains no SR or RR RTCP packets at all -- as can happen with Reduced-Size RTCP packets [RFC5506] -- the SSRC count SHALL be considered to be 1.

Note: The above makes it possible to amortize the packet overhead between the number of SSRCs sharing a RTCP compound packet.

For an RTCP end-point that doesn't follow the above rule, and instead uses the full RTCP compound packet size as input, the average RTCP reporting interval will be scaled up (i.e. become longer) with a factor that is proportional to the number of SSRCs sourcing RTCP packets in an RTCP compound packet as well as the set of SSRCs being aggregated in proportion to the total number of participants. This factor can quite easily become larger than 5, e.g. with an 1500 byte MTU and an average per-SSRC sum of RTCP packets of 240 bytes, the MTU will fit 6 packets. If the receiver end-point has a single SSRC and all other endpoints fill their MTU fully, the factor will be close to 6. If the RTCP configuration is such that the transmission interval is bandwidth limited, rather than any type of minimal interval limitation (Tmin or T_RR_INT), then the other end-points will likely time out this SSRC due to it using an regular RTCP interval is more than 5 times the rest of the endpoints.

5.3.2. Scheduling RTCP with Multiple Reporting SSRCs

When implementing RTCP packet scheduling for cases where multiple reporting SSRCs are aggregating their RTCP packets in the same compound packet there are a number of challenges. First of all, we have the goal of not changing the general properties of the RTCP packet transmissions, which include the general inter-packet distribution, and the behavior for dealing with flash joins as well as other dynamic events.

The below specified mechanism deals with:

- That one can't have a-priori knowledge about which RTCP packets are to be sent, or their size, prior to generating the packets. In which case, the time from generation to transmission ought to be as short as possible to minimize the information that becomes stale.

- That one has an MTU limit, that one ought to avoid exceeding, as that requires lower-layer fragmentation (e.g., IP fragmentation) which impacts the packets' probability of reaching the receiver(s).

Schedule all the endpoint's local SSRCs individually for transmission using the regular calculation of Tn for the profile being used. Each time a SSRC's Tn timer expires, do the regular reconsideration. If the reconsideration indictes that an RTCP packet is to be sent:

1. Consider if an additional SSRC can be added. That consideration is done by picking the SSRC which has the Tn value closest in time to now (Tc).

2. Calculate how much space for RTCP packets would be needed to add that SSRC.

3. If the considered SSRC's RTCP Packets fit within the lower layer datagram's Maximum Transmission Unit, taking the necessary protocol headers into account and the consumed space by prior SSRCs, then add that SSRC's RTCP packets to the compound packet and go again to Step 1.

4. If the considered SSRC's RTCP Packets will not fit within the compound packet, then transmit the generated compound packet.

5. Update the RTCP Parameters for each SSRC that has been included in the sent RTCP packet. The Tp value for each SSRC MUST be updated as follows:

   For the first SSRC: As this SSRC was the one that was reconsidered the tp value is set to the tc as defined in RTP [RFC3550].

   For any additional SSRC: The tp value SHALL be set to the transmission time this SSRC would have had it not been aggregated and given the current existing session context. This value is derived by taking this SSRC's Tn value and performing reconisderation and updating tn until tp + T <= tn. Then set tp to this tn value.

6. For the sent SSRCs calculate new tn values based on the updated parameters and reschedule the timers.

Reverse reconsideration needs to be performed as specified in RTP [RFC3550]. It is important to note that under the above algorithm when performing reconsideration, the value of tp can actually be larger than tc. However, that still has the desired effect of proportionally pulling the tp value towards tc (as well as tn) as the group size shrinks in direct proportion the reduced group size.

The above algorithm has been shown in simulations to maintain the inter-RTCP-packet transmission distribution for the SSRCs and consume the same amount of bandwidth as non-aggregated packets in RTP sessions with static sets of participants. With this algorithm the actual transmission interval for any SSRC triggering an RTCP compound packet transmission is following the regular transmission rules. It also handles the cases where the number of SSRCs that can be included

in an aggregated packet varies. An SSRC that previously was aggregated and fails to fit in a packet still has its own transmission scheduled according to normal rules. Thus, it will trigger a transmission in due time, or the SSRC will be included in another aggregate.

The algorithm's behavior under SSRC group size changes is under investigation. However, it is expected to be well behaved based on the following analyses.

RTP sessions where the number of SSRC are growing: When the group size is growing, the Td values grow in proportion to the number of new SSRCs in the group. The reconsideration when the timer for the tn expires, that SSRC will reconsider the transmission and with a certain probability reschedule the tn timer. This part of the reconsideration algorithm is only impacted by the above algorithm by having tp values that are in the future instead of set to the time of the actual last transmission at the time of updating tp. Thus the scheduling causes in worst case a plateau effect for that SSRC. That effect depends on how far into the future tp can advance.

RTP sessions where the number of SSRC are shrinking: When the group shrinks, reverse reconsideration moves the tp and tn values towards tc proportionally to the number of SSRCs that leave the session compared to the total number of participants when they left. Thus the also group size reductions need to be handled.

In general the potential issue that might exist depends on how far into the future the tp value can drift compared to the actual packet transmissions that occur. That drift can only occur for an SSRC that never is the trigger for RTCP packet transmission and always gets aggregated and where the calculcated packet transmission interval randomly occurs so that tn - tp for this SSRC is on average larger than the ones that gets transmitted.

5.4. RTP/AVPF Feedback Packets

This section discusses the transmission of RTP/AVPF feedback packets when the transmitting endpoint has multiple SSRCs.

5.4.1. The SSRC Used

When an RTP endpoint has multiple SSRCs, it can make certain choices on which SSRC to use as the source of an RTCP Feedback Packet. This sub-section discusses some considerations of this.

- o The media type of the media the SSRC transmits is actually not a relevant factor when considering if an SSRC can transmit a particular Feedback message.

- o Feedback messages which are Notification or Indications regarding the endpoint's own RTP packet stream need to be sent using the SSRC transmitting the media it relates to. This also includes notifications that are related to a received request or command.

- o The SSRC used to send feedback messages has a role as either a media sender or a receiver. The bandwidth pools can be different for SSRCs that are senders and receivers. Thus feedback messages that expect to be more frequent can be sent from an SSRC that has the better possibility of sending frequent RTCP compound packets or reduced size packets. This also affects the consideration if the SSRC can be used in immediate mode or not.

- o Some Feedback Types requires consistency in the sender. For example TMMBR, if one sets a limitation, the same SSRC needs to be the one that increases it. Others can simply benefit from having this property.

Note that the source of the feedback RTCP packet does not need to be any of the sources (SSRC) including SR/RR packets in a compound packet. For Reduced-Size RTCP [RFC5506] the aggregation of feedback messages from multiple sources are not limited, beyond the consideration in Section 4.2.2 of [RFC5506].

5.4.2. Scheduling a Feedback Packet

When an SSRC has a need to transmit a feedback packet in early mode it follows the scheduling rules defined in Section 3.5 in RTP/AVPF [RFC4585]. When following these rules the following clarifications need to be taken into account:

- o That a session is considered to be point-to-point or multiparty not based on the number of SSRCs, but the number of endpoints directly seen in the RTP session by the endpoint. tbd: Clarify what is considered to "see" an endpoint?

- o Note that when checking if there is already a scheduled compound RTCP packet containing feedback messages (Step 2 in Section 3.5.2), that check is done considering all local SSRCs.

TBD: The above does not allow an SSRC that is unable to send either an early or regular RTCP packet with the feedback message within the T_max_fb_delay to trigger another SSRC to send an early packet to which it could piggyback. Nor does it allow feedback to piggyback on

even regular RTCP packet transmissions that occur within T_max_fb_delay. A question is if either of these behaviours ought to be allowed.

The latter appears simple and straight forward. Instead of discarding a FB message in step 4a: alternative 2, one could place such messages in a cache with a discard time equal to T_max_fb_delay, and in case any of the SSRCs schedule an RTCP packet for transmission within that time, it includes this message.

The former case can have more widespread impact on the application, and possibly also on the RTCP bandwidth consumption as it allows for more massive bursts of RTCP packets. Still, on a time scale of a regular reporting interval, it ough to have no effect on the RTCP bandwidth as the extra feedback messages increase the avg_rtcp_size.

6. RTCP Considerations for Streams with Disparate Rates

It is possible for a single RTP session to carry streams of greatly differing bandwidth. There are two scenarios where this can occur. The first is when a single RTP session carries multiple flows of the same media type, but with very different quality; for example a video switching multi-point conference unit might send a full rate high-definition video stream of the active speaker but only thumbnails for the other participants, all sent in a single RTP session. The second scenarios occurs when audio and video flows are sent in a single RTP session, as discussed in [I-D.ietf-avtcore-multi-media-rtp-session].

An RTP session has a single set of parameters that configure the session bandwidth, the RTCP sender and receiver fractions (e.g., via the SDP "b=RR:" and "b=RS:" lines), and the parameters of the RTP/AVPF profile [RFC4585] (e.g., trr-int) if that profile (or its secure extension, RTP/SAVPF [RFC5124]) is used. As a consequence, the RTCP reporting interval will be the same for every SSRC in an RTP session. This uniform RTCP reporting interval can result in RTCP reports being sent more often than is considered desirable for a particular media type. For example, if an audio flow is multiplexed with a high quality video flow where the session bandwidth is configured to match the video bandwidth, this can result in the RTCP packets having a greater bandwidth allocation than the audio data rate. If the reduced minimum RTCP interval described in Section 6.2 of [RFC3550] is used in the session, which might be appropriate for video where rapid feedback is wanted, the audio sources could be expected to send RTCP packets more often than they send audio data packets. This is most likely undesirable, and while the mismatch can be reduced through careful tuning of the RTCP parameters, particularly trr_int in RTP/AVPF sessions, it is inherent in the design of the RTCP timing

rules, and affects all RTP sessions containing flows with mismatched bandwidth.

Having multiple media types in one RTP session also results in more SSRCs being present in this RTP session. This increasing the amount of cross reporting between the SSRCs. From an RTCP perspective, two RTP sessions with half the number of SSRCs in each will be slightly more efficient. If someone needs either the higher efficiency due to the lesser number of SSRCs or the fact that one can't tailor RTCP usage per media type, they need to use independent RTP sessions.

When it comes to configuring RTCP the need for regular periodic reporting needs to be weighted against any feedback or control messages being sent. Applications using RTP/AVPF or RTP/SAVPF are RECOMMENDED to consider setting the trr-int parameter to a value suitable for the application's needs, thus potentially reducing the need for regular reporting and thus releasing more bandwidth for use for feedback or control.

Another aspect of an RTP session with multiple media types is that the RTCP packets, RTCP Feedback Messages, or RTCP XR metrics used might not be applicable to all media types. Instead, all RTP/RTCP endpoints need to correlate the media type of the SSRC being referenced in a message or packet and only use those that apply to that particular SSRC and its media type. Signalling solutions might have shortcomings when it comes to indicating that a particular set of RTCP reports or feedback messages only apply to a particular media type within an RTP session.

6.1. Timing out SSRCs

All SSRCs used in an RTP session MUST use the same timeout behaviour to avoid premature timeouts. This will depend on the RTP profile and its configuration. The RTP specification provides several options that can influence the values used when calculating the time interval. To avoid interoperability issues when using this specification, this document makes several clarifications to the calculations.

For RTP/AVP, RTP/SAVP, RTP/AVPF, and RTP/SAVPF with T_rr_interval = 0, the timeout interval SHALL be calculated using a multiplier of 5, i.e. the timeout interval becomes 5*Td. The Td calculation SHALL be done using a Tmin value of 5 seconds, not the reduced minimal interval even if used to calculate RTCP packet transmission intervals. If using either the RTP/AVPF or RTP/SAVPF profiles with T_rr_interval != 0 then the calculation as specified in Section 3.5.4 of RFC 4585 SHALL be used with a multiplier of 5, i.e. Tmin in the Td calculation is the T_rr_interval.

If endpoints implementing the RTP/AVP and RTP/AVPF profiles (or their secure variants) are combined in a single RTP session, and the RTP/AVPF endpoints use a non-zero T_rr_interval that is significantly lower than 5 seconds, then there is a risk that the RTP/AVPF endpoints will prematurely timeout the RTP/AVP SSRCs due to their different RTCP timeout intervals. Conversely, if the RTP/AVPF endpoints use a T_rr_interval that is significant larger than 5 seconds, there is a risk that the RTP/AVP endpoints will timeout the RTP/AVPF SSRCs. If such mixed RTP profiles are used, (though this is NOT RECOMMENDED), the RTP/AVPF session SHOULD use a non-zero T_rr_interval that is 4 seconds.

Note: It might appear strange to use a T_rr_interval of 4 seconds. It might be intuitive that this value ought to be 5 seconds, as then both the RTP/AVP and RTP/AVPF would use the same timeout period. However, considering regular RTCP transmission and their packet intervals for RTP/AVPF its mean value will (with non-zero T_rr_interval) be larger than T_rr_interval due to the scheduling algorithm. Thus, to enable an equal amount of regular RTCP transmissions in each directions between RTP/AVP and RTP/AVPF endpoints, taking the altered timeout intervals into account, the optimal value is around four (4), where almost four transmissions will on average occur in each direction between the different profile types given an otherwise good configuration of parameters in regards to T_rr_interval. If the RTCP bandwidth paramters are selected so that Td based on bandwidth is close to 4, i.e. close to T_rr_interval the risk increases that RTP/AVPF SSRCs will be timed out by RTP/AVP endpoints, as the RTP/AVPF SSRC might only manage two transmissions in the timeout period.

6.2. Tuning RTCP transmissions

This sub-section discusses what tuning can be done to reduce the downsides of the shared RTCP packet intervals. First, it is considered what possibilites exist for the RTP/AVP [RFC3551] profile, then what additional tools are provided by RTP/AVPF [RFC4585].

6.2.1. RTP/AVP and RTP/SAVP

When using the RTP/AVP or RTP/SAVP profiles the tuning one can do is very limited. The controls one has are limited to the RTCP bandwidth values and whether the minimum RTCP interval is scaled according to the bandwidth. As the scheduling algorithm includes both random factors and reconsideration, one can't simply calculate the expected average transmission interval using the formula for Td. But it does indicate the important factors affecting the transmission interval, namely the RTCP bandwidth available for the role (Active Sender or Participant), the average RTCP packet size, and the number of SSRCs

classified in the relevant role. Note that if the ratio of senders to total number of session participants is larger than the ratio of RTCP bandwidth for senders in relation to the total RTCP bandwidth, then senders and receivers are treated together.

Let's start with some basic observations:

a. Unless the scaled minimum RTCP interval is used, then Td prior to randomization and reconsideration can never be less than 5 seconds (assuming default Tmin of 5 seconds).

b. If the scaled minimum RTCP interval is used, Td can become as low as 360 divided by RTP Session bandwidth in kilobits. In SDP the RTP session bandwidth is signalled using b=AS. An RTP Session bandwidth of 72 kbps results in Tmin being 5 seconds. An RTP session bandwidth of 360 kbps of course gives a Tmin of 1 second, and to achieve a Tmin equal to once every frame for a 25 Hz video stream requires an RTP session bandwidth of 9 Mbps! (The use of the RTP/AVPF or RTP/SAVPF profile allows a smaller Tmin, and hence more frequent RTCP reports, as discussed below).

c. Let's calculate the number (n) of SSRCs in the RTP session that 5% of the session bandwidth can support to yield a Td value equal to Tmin with minimal scaling. For this calculation we have to make two assumptions. The first is that we will consider most or all SSRC being senders, resulting in everyone sharing the available bandwidth. Secondly we will select an average RTCP packet size. This packet will consist of an SR, containing (n-1) report blocks up to 31 report blocks, and an SDES item with at least a CNAME (17 bytes in size) in it. Such a basic packet will be 800 bytes for n>=32. With these parameters, and as the bandwidth goes up the time interval is proportionally decreased (due to minimal scaling), thus all the example bandwidths 72 kbps, 360 kbps and 9 Mbps all support 9 SSRCs.

d. The actual transmission interval for a Td value is [0.5*Td/1.21828,1.5*Td/1.21828], which means that for Td = 5 seconds, the interval is actually [2.052,6.156] and the distribution is not uniform, but rather exponentially-increasing. The probability for sending at time X, given it is within the interval, is probability of picking X in the interval times the probability to randomly picking a number that is <=X within the interval with an uniform probability distribution. This results in that the majority of the probability mass is above the Td value.

To conclude, with RTP/AVP and RTP/SAVP the key limitation for small unicast sessions is going to be the Tmin value. Thus the RTP session bandwidth configured in RTCP has to be sufficiently high to reach the

reporting goals the application has following the rules for the scaled minimal RTCP interval.

6.2.2. RT/AVPF and RTP/SAVPF

When using RTP/AVPF or RTP/SAVPF we get a quite powerful additional tool, the setting of the T_rr_interval which has several effects on the RTCP reporting. First of all as Tmin is set to 0 after the initial transmission, the regular reporting interval is instead determined by the regular bandwidth based calculation and the T_rr_interval. This has the effect that we are no longer restricted by the minimal interval or even the scaling rule for the minimal rule. Instead the RTCP bandwidth and the T_rr_interval are the governing factors.

Now it also becomes important to separate between the application's need for regular reports and RTCP feedback packet types. In both regular RTCP mode, as in Early RTCP Mode, the usage of the T_rr_interval prevents regular RTCP packets, i.e. packets without any Feedback packets, to be sent more often than T_rr_interval. This value is applied to prevent any regular RTCP packet to be sent less than T_rr_interval times a uniformly distributed random value from the interval [0.5,1.5] after the previous regular packet packet. The random value recalculated after each regular RTCP packet transmission.

So applications that have a use for feedback packets for some media streams, for example video streams, but don't want frequent regular reporting for audio, could configure the T_rr_interval to a value so that the regular reporting for both audio and video is at a level that is considered acceptable for the audio. They could then use feedback packets, which will include RTCP SR/RR packets, unless reduced-size RTCP feedback packets [RFC5506] are used, and can include other report information in addition to the feedback packet that needs to be sent. That way the available RTCP bandwidth can be focused for the use which provides the most utility for the application.

Using T_rr_interval still requires one to determine suitable values for the RTCP bandwidth value, in fact it might make it even more important, as this is more likely to affect the RTCP behaviour and performance than when using RTP/AVP, as there are fewer limitations affecting the RTCP transmission.

When using T_rr_interval, i.e. having it be non zero, there are configurations that have to be avoided. If the resulting Td value is smaller but close to T_rr_interval then the interval in which the actual regular RTCP packet transmission falls into becomes very

large, from 0.5 times T_rr_interval up to 2.73 times the T_rr_interval. Therefore for configuration where one intends to have Td smaller than T_rr_interval, then Td is RECOMMENDED to be targeted at values less than 1/4th of T_rr_interval which results in that the range becomes [0.5*T_rr_interval, 1.81*T_rr_interval].

With RTP/AVPF, using a T_rr_interval of 0 or with another low value significantly lower than Td still has utility, and different behaviour compared to RTP/AVP. This avoids the Tmin limitations of RTP/AVP, thus allowing more frequent regular RTCP reporting. In fact this will result that the RTCP traffic becomes as high as the configured values.

(tbd: a future version of this memo will include examples of how to choose RTCP parameters for common scenarios)

There exists no method within the specification for using different regular RTCP reporting intervals depending on the media type or individual media stream.

7. Security Considerations

In the secure RTP protocol (SRTP) [RFC3711], the cryptographic context of a compound SRTCP packet is the SSRC of the sender of the first RTCP (sub-)packet. This could matter in some cases, especially for keying mechanisms such as Mikey [RFC3830] which allow use of per-SSRC keying.

Other than that, the standard security considerations of RTP apply; sending multiple media streams from a single endpoint does not appear to have different security consequences than sending the same number of streams.

8. Open Issues

At this stage this document contains a number of open issues. The below list tries to summarize the issues:

1. Do we need to provide a recommendation for unicast session joiners with many sources to not use 0 initial minimal interval from bit-rate burst perspective?

2. RTCP parameters for common scenarios in Section 6.2?

3. Is scheduling algorithm working well with dynamic changes?

4. Are the scheduling algorithm changes impacting previous implementations in such a way that the report aggregation has to be agreed on, and thus needs to be considered as an optimization?

5. An open question is if any improvements or clarifications ought to be allowed regarding FB message scheduling in multi-SSRC endpoints.

9. IANA Considerations

No IANA actions needed.

10. References

10.1. 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.

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

[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.

[RFC5124] Ott, J. and E. Carrara, "Extended Secure RTP Profile for Real-time Transport Control Protocol (RTCP)-Based Feedback (RTP/SAVPF)", RFC 5124, February 2008.

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

10.2. Informative References

[I-D.ietf-avtcore-multi-media-rtp-session] Westerlund, M., Perkins, C., and J. Lennox, "Sending Multiple Types of Media in a Single RTP Session", draft-ietf-avtcore-multi-media-rtp-session-04 (work in progress), January 2014.

[I-D.ietf-avtcore-rtp-multi-stream-optimisation]
Lennox, J., Westerlund, M., Wu, W., and C. Perkins, "Sending Multiple Media Streams in a Single RTP Session: Grouping RTCP Reception Statistics and Other Feedback", draft-ietf-avtcore-rtp-multi-stream-optimisation-01 (work in progress), January 2014.

[I-D.ietf-avtcore-rtp-topologies-update]
Westerlund, M. and S. Wenger, "RTP Topologies", draft-ietf-avtcore-rtp-topologies-update-01 (work in progress), October 2013.

[I-D.ietf-clue-framework]
Duckworth, M., Pepperell, A., and S. Wenger, "Framework for Telepresence Multi-Streams", draft-ietf-clue-framework-14 (work in progress), February 2014.

[I-D.ietf-mmusic-sdp-bundle-negotiation]
Holmberg, C., Alvestrand, H., and C. Jennings, "Multiplexing Negotiation Using Session Description Protocol (SDP) Port Numbers", draft-ietf-mmusic-sdp-bundle-negotiation-05 (work in progress), October 2013.

[RFC3551] Schulzrinne, H. and S. Casner, "RTP Profile for Audio and Video Conferences with Minimal Control", STD 65, RFC 3551, July 2003.

[RFC3611] Friedman, T., Caceres, R., and A. Clark, "RTP Control Protocol Extended Reports (RTCP XR)", RFC 3611, November 2003.

[RFC3830] Arkko, J., Carrara, E., Lindholm, F., Naslund, M., and K. Norrman, "MIKEY: Multimedia Internet KEYing", RFC 3830, August 2004.

[RFC4588] Rey, J., Leon, D., Miyazaki, A., Varsa, V., and R. Hakenberg, "RTP Retransmission Payload Format", RFC 4588, July 2006.

[RFC6190] Wenger, S., Wang, Y., Schierl, T., and A. Eleftheriadis, "RTP Payload Format for Scalable Video Coding", RFC 6190, May 2011.

Appendix A. Changes From Earlier Versions

Note to the RFC-Editor: please remove this section prior to publication as an RFC.

A.1. Changes From WG Draft -02

- Changed usage of Media Stream

- Added Updates RFC 4585

- Added rules for how to deal with RTCP when aggregating multiple SSRCs report in same compound packet:

  * avg_rtcp_size calcualtion

  * Scheduling rules to maintain timing

- Started a section clarifying and discsussing RTP/AVPF Feedback Packets and their scheduling.

A.2. Changes From WG Draft -01

- None, a keep-alive version

A.3. Changes From WG Draft -00

- Split the Reporting Group Extension from this draft into draft-ietf-avtcore-rtp-multi-stream-optimization-00.

- Added RTCP tuning considerations from draft-ietf-avtcore-multi-media-rtp-session-02.

A.4. Changes From Individual Draft -02

- Resubmitted as working group draft.

- Updated references.

A.5. Changes From Individual Draft -01

- Merged with draft-wu-avtcore-multisrc-endpoint-adver.

- Changed how Reporting Groups are indicated in RTCP, to make it clear which source(s) is the group's reporting sources.

- Clarified the rules for when sources can be placed in the same reporting group.

- Clarified that mixers and translators need to pass reporting group SDES information if they are forwarding RR and SR traffic from members of a reporting group.

A.6. Changes From Individual Draft -00

- Added the Reporting Group semantic to explicitly indicate which sources come from a single endpoint, rather than leaving it implicit.

- Specified that Reporting Group semantics (as they now are) apply to AVPF and XR, as well as to RR/SR report blocks.

- Added a description of the cascaded source-projecting mixer, along with a calculation of its RTCP overhead if reporting groups are not in use.

- Gave some guidance on how the flexibility of RTCP randomization allows some freedom in RTCP multiplexing.

- Clarified the language of several of the recommendations.

- Added an open issue discussing how avg_rtcp_size ought to be calculated for multiplexed RTCP.

- Added an open issue discussing how RTCP bandwidths are to be chosen for sessions where source bandwidths greatly differ.

Authors' Addresses

Jonathan Lennox
Vidyo, Inc.
433 Hackensack Avenue
Seventh Floor
Hackensack, NJ 07601
US

Email: jonathan@vidyo.com

Magnus Westerlund
Ericsson
Farogatan 6
SE-164 80 Kista
Sweden

Phone: +46 10 714 82 87
Email: magnus.westerlund@ericsson.com

Qin Wu
Huawei
101 Software Avenue, Yuhua District
Nanjing, Jiangsu 210012
China

Email: sunseawq@huawei.com

Colin Perkins
University of Glasgow
School of Computing Science
Glasgow G12 8QQ
United Kingdom

Email: csp@csperkins.org