

AVTCORE Working Group  
Internet-Draft  
Updates: 3550 (if approved)  
Intended status: Standards Track  
Expires: January 16, 2014

C. S. Perkins  
University of Glasgow  
V. Singh  
Aalto University  
July 15, 2013

Multimedia Congestion Control: Circuit Breakers for Unicast RTP Sessions  
draft-ietf-avtcore-rtp-circuit-breakers-03

Abstract

The Real-time Transport Protocol (RTP) is widely used in telephony, video conferencing, and telepresence applications. Such applications are often run on best-effort UDP/IP networks. If congestion control is not implemented in the applications, then network congestion will deteriorate the user's multimedia experience. This document does not propose a congestion control algorithm; instead, it defines a minimal set of RTP "circuit-breakers". Circuit-breakers are conditions under which an RTP sender needs to stop transmitting media data in order to protect the network from excessive congestion. It is expected that, in the absence of severe congestion, all RTP applications running on best-effort IP networks will be able to run without triggering these circuit breakers. Any future RTP congestion control specification will be expected to operate within the constraints defined by these circuit breakers.

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 January 16, 2014.

Copyright Notice

Copyright (c) 2013 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 . . . . .	2
2. Terminology . . . . .	3
3. Background . . . . .	3
4. RTP Circuit Breakers for Systems Using the RTP/AVP Profile .	6
4.1. RTP/AVP Circuit Breaker #1: Media Timeout . . . . .	7
4.2. RTP/AVP Circuit Breaker #2: RTCP Timeout . . . . .	8
4.3. RTP/AVP Circuit Breaker #3: Congestion . . . . .	9
4.4. Ceasing Transmission . . . . .	12
5. RTP Circuit Breakers for Systems Using the RTP/AVPF Profile .	12
6. Impact of RTCP XR . . . . .	13
7. Impact of RTCP Reporting Groups . . . . .	14
8. Impact of Explicit Congestion Notification (ECN) . . . . .	14
9. Security Considerations . . . . .	14
10. IANA Considerations . . . . .	15
11. Acknowledgements . . . . .	15
12. References . . . . .	15
12.1. Normative References . . . . .	15
12.2. Informative References . . . . .	15
Authors' Addresses . . . . .	17

## 1. Introduction

The Real-time Transport Protocol (RTP) [RFC3550] is widely used in voice-over-IP, video teleconferencing, and telepresence systems. Many of these systems run over best-effort UDP/IP networks, and can suffer from packet loss and increased latency if network congestion occurs. Designing effective RTP congestion control algorithms, to adapt the transmission of RTP-based media to match the available network capacity, while also maintaining the user experience, is a difficult but important problem. Many such congestion control and media adaptation algorithms have been proposed, but to date there is no consensus on the correct approach, or even that a single standard algorithm is desirable.

This memo does not attempt to propose a new RTP congestion control algorithm. Rather, it proposes a minimal set of "circuit breakers";

conditions under which there is general agreement that an RTP flow is causing serious congestion, and ought to cease transmission. It is expected that future standards-track congestion control algorithms for RTP will operate within the envelope defined by this memo.

## 2. Terminology

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]. This interpretation of these key words applies only when written in ALL CAPS. Mixed- or lower-case uses of these key words are not to be interpreted as carrying special significance in this memo.

## 3. Background

We consider congestion control for unicast RTP traffic flows. This is the problem of adapting the transmission of an audio/visual data flow, encapsulated within an RTP transport session, from one sender to one receiver, so that it matches the available network bandwidth. Such adaptation needs to be done in a way that limits the disruption to the user experience caused by both packet loss and excessive rate changes. Congestion control for multicast flows is outside the scope of this memo. Multicast traffic needs different solutions, since the available bandwidth estimator for a group of receivers will differ from that for a single receiver, and because multicast congestion control has to consider issues of fairness across groups of receivers that do not apply to unicast flows.

Congestion control for unicast RTP traffic can be implemented in one of two places in the protocol stack. One approach is to run the RTP traffic over a congestion controlled transport protocol, for example over TCP, and to adapt the media encoding to match the dictates of the transport-layer congestion control algorithm. This is safe for the network, but can be suboptimal for the media quality unless the transport protocol is designed to support real-time media flows. We do not consider this class of applications further in this memo, as their network safety is guaranteed by the underlying transport.

Alternatively, RTP flows can be run over a non-congestion controlled transport protocol, for example UDP, performing rate adaptation at the application layer based on RTP Control Protocol (RTCP) feedback. With a well-designed, network-aware, application, this allows highly effective media quality adaptation, but there is potential to disrupt the network's operation if the application does not adapt its sending rate in a timely and effective manner. We consider this class of applications in this memo.

Congestion control relies on monitoring the delivery of a media flow, and responding to adapt the transmission of that flow when there are signs that the network path is congested. Network congestion can be detected in one of three ways: 1) a receiver can infer the onset of congestion by observing an increase in one-way delay caused by queue build-up within the network; 2) if Explicit Congestion Notification (ECN) [RFC3168] is supported, the network can signal the presence of congestion by marking packets using ECN Congestion Experienced (CE) marks; or 3) in the extreme case, congestion will cause packet loss that can be detected by observing a gap in the received RTP sequence numbers. Once the onset of congestion is observed, the receiver has to send feedback to the sender to indicate that the transmission rate needs to be reduced. How the sender reduces the transmission rate is highly dependent on the media codec being used, and is outside the scope of this memo.

There are several ways in which a receiver can send feedback to a media sender within the RTP framework:

- o The base RTP specification [RFC3550] defines RTCP Reception Report (RR) packets to convey reception quality feedback information, and Sender Report (SR) packets to convey information about the media transmission. RTCP SR packets contain data that can be used to reconstruct media timing at a receiver, along with a count of the total number of octets and packets sent. RTCP RR packets report on the fraction of packets lost in the last reporting interval, the cumulative number of packets lost, the highest sequence number received, and the inter-arrival jitter. The RTCP RR packets also contain timing information that allows the sender to estimate the network round trip time (RTT) to the receivers. RTCP reports are sent periodically, with the reporting interval being determined by the number of SSRCs used in the session and a configured session bandwidth estimate (the number of SSRCs used is usually two in a unicast session, one for each participant, but can be greater if the participants send multiple media streams). The interval between reports sent from each receiver tends to be on the order of a few seconds on average, and it is randomised to avoid synchronisation of reports from multiple receivers. RTCP RR packets allow a receiver to report ongoing network congestion to the sender. However, if a receiver detects the onset of congestion partway through a reporting interval, the base RTP specification contains no provision for sending the RTCP RR packet early, and the receiver has to wait until the next scheduled reporting interval.
- o The RTCP Extended Reports (XR) [RFC3611] allow reporting of more complex and sophisticated reception quality metrics, but do not change the RTCP timing rules. RTCP extended reports of potential

interest for congestion control purposes are the extended packet loss, discard, and burst metrics [RFC3611], [I-D.ietf-xrblock-rtcp-xr-discard], [I-D.ietf-xrblock-rtcp-xr-discard-rle-metrics], [I-D.ietf-xrblock-rtcp-xr-burst-gap-discard], [I-D.ietf-xrblock-rtcp-xr-burst-gap-loss]; and the extended delay metrics [RFC6843], [RFC6798]. Other RTCP Extended Reports that could be helpful for congestion control purposes might be developed in future.

- o Rapid feedback about the occurrence of congestion events can be achieved using the Extended RTP Profile for RTCP-Based Feedback (RTP/AVPF) [RFC4585] in place of the more common RTP/AVP profile [RFC3551]. This modifies the RTCP timing rules to allow RTCP reports to be sent early, in some cases immediately, provided the average RTCP reporting interval remains unchanged. It also defines new transport-layer feedback messages, including negative acknowledgements (NACKs), that can be used to report on specific congestion events. The use of the RTP/AVPF profile is dependent on signalling, but is otherwise generally backwards compatible with the RTP/AVP profile, as it keeps the same average RTCP reporting interval as the base RTP specification. The RTP Codec Control Messages [RFC5104] extend the RTP/AVPF profile with additional feedback messages that can be used to influence that way in which rate adaptation occurs. The dynamics of how rapidly feedback can be sent are unchanged.
- o Finally, Explicit Congestion Notification (ECN) for RTP over UDP [RFC6679] can be used to provide feedback on the number of packets that received an ECN Congestion Experienced (CE) mark. This RTCP extension builds on the RTP/AVPF profile to allow rapid congestion feedback when ECN is supported.

In addition to these mechanisms for providing feedback, the sender can include an RTP header extension in each packet to record packet transmission times. There are two methods: [RFC5450] represents the transmission time in terms of a time-offset from the RTP timestamp of the packet, while [RFC6051] includes an explicit NTP-format sending timestamp (potentially more accurate, but a higher header overhead). Accurate sending timestamps can be helpful for estimating queuing delays, to get an early indication of the onset of congestion.

Taken together, these various mechanisms allow receivers to provide feedback on the senders when congestion events occur, with varying degrees of timeliness and accuracy. The key distinction is between systems that use only the basic RTCP mechanisms, without RTP/AVPF rapid feedback, and those that use the RTP/AVPF extensions to respond to congestion more rapidly.

#### 4. RTP Circuit Breakers for Systems Using the RTP/AVP Profile

The feedback mechanisms defined in [RFC3550] and available under the RTP/AVP profile [RFC3551] are the minimum that can be assumed for a baseline circuit breaker mechanism that is suitable for all unicast applications of RTP. Accordingly, for an RTP circuit breaker to be useful, it needs to be able to detect that an RTP flow is causing excessive congestion using only basic RTCP features, without needing RTCP XR feedback or the RTP/AVPF profile for rapid RTCP reports.

RTCP is a fundamental part of the RTP protocol, and the mechanisms described here rely on the implementation of RTCP. Implementations which claim to support RTP, but that do not implement RTCP, cannot use the circuit breaker mechanisms described in this memo. Such implementations SHOULD NOT be used on networks that might be subject to congestion unless equivalent mechanisms are defined using some non-RTCP feedback channel to report congestion and signal circuit breaker conditions.

Three potential congestion signals are available from the basic RTCP SR/RR packets and are reported for each synchronisation source (SSRC) in the RTP session:

1. The sender can estimate the network round-trip time once per RTCP reporting interval, based on the contents and timing of RTCP SR and RR packets.
2. Receivers report a jitter estimate (the statistical variance of the RTP data packet inter-arrival time) calculated over the RTCP reporting interval. Due to the nature of the jitter calculation ([RFC3550], section 6.4.4), the jitter is only meaningful for RTP flows that send a single data packet for each RTP timestamp value (i.e., audio flows, or video flows where each packet comprises one video frame).
3. Receivers report the fraction of RTP data packets lost during the RTCP reporting interval, and the cumulative number of RTP packets lost over the entire RTP session.

These congestion signals limit the possible circuit breakers, since they give only limited visibility into the behaviour of the network.

RTT estimates are widely used in congestion control algorithms, as a proxy for queuing delay measures in delay-based congestion control or to determine connection timeouts. RTT estimates derived from RTCP SR and RR packets sent according to the RTP/AVP timing rules are far too infrequent to be useful though, and don't give enough information to distinguish a delay change due to routing updates from queuing delay

caused by congestion. Accordingly, we cannot use the RTT estimate alone as an RTP circuit breaker.

Increased jitter can be a signal of transient network congestion, but in the highly aggregated form reported in RTCP RR packets, it offers insufficient information to estimate the extent or persistence of congestion. Jitter reports are a useful early warning of potential network congestion, but provide an insufficiently strong signal to be used as a circuit breaker.

The remaining congestion signals are the packet loss fraction and the cumulative number of packets lost. If considered carefully, these can be effective indicators that congestion is occurring in networks where packet loss is primarily due to queue overflows, although loss caused by non-congestive packet corruption can distort the result in some networks. TCP congestion control intentionally tries to fill the router queues, and uses the resulting packet loss as congestion feedback. An RTP flow competing with TCP traffic will therefore expect to see a non-zero packet loss fraction that has to be related to TCP dynamics to estimate available capacity. This behaviour of TCP is reflected in the congestion circuit breaker below, and will affect the design of any RTP congestion control protocol.

Two packet loss regimes can be observed: 1) RTCP RR packets show a non-zero packet loss fraction, while the extended highest sequence number received continues to increment; and 2) RR packets show a loss fraction of zero, but the extended highest sequence number received does not increment even though the sender has been transmitting RTP data packets. The former corresponds to the TCP congestion avoidance state, and indicates a congested path that is still delivering data; the latter corresponds to a TCP timeout, and is most likely due to a path failure. A third condition is that data is being sent but no RTCP feedback is received at all, corresponding to a failure of the reverse path. We derive circuit breaker conditions for these loss regimes in the following.

#### 4.1. RTP/AVP Circuit Breaker #1: Media Timeout

If RTP data packets are being sent, but the RTCP SR or RR packets reporting on that SSRC indicate a non-increasing extended highest sequence number received, this is an indication that those RTP data packets are not reaching the receiver. This could be a short-term issue affecting only a few packets, perhaps caused by a slow-to-open firewall or a transient connectivity problem, but if the issue persists, it is a sign of a more ongoing and significant problem. Accordingly, if a sender of RTP data packets receives two or more consecutive RTCP SR or RR packets from the same receiver, and those packets correspond to its transmission and have a non-increasing

extended highest sequence number received field (i.e., the sender receivers at least three RTCP SR or RR packets that report the same value in the extended highest sequence number received field for an SSRC, but the sender has sent RTP data packets for that SSRC that would have caused an increase in the reported value of the extended highest sequence number received if they had reached the receiver), then that sender SHOULD cease transmission (see Section 4.4).

The reason for waiting for two or more consecutive RTCP packets with a non-increasing extended highest sequence number is to give enough time for transient reception problems to resolve themselves, but to stop problem flows quickly enough to avoid causing serious ongoing network congestion. A single RTCP report showing no reception could be caused by a transient fault, and so will not cease transmission. Waiting for more than two consecutive RTCP reports before stopping a flow might avoid some false positives, but could lead to problematic flows running for a long time period (potentially tens of seconds, depending on the RTCP reporting interval) before being cut off.

#### 4.2. RTP/AVP Circuit Breaker #2: RTCP Timeout

In addition to media timeouts, as were discussed in Section 4.1, an RTP session has the possibility of an RTCP timeout. This can occur when RTP data packets are being sent, but there are no RTCP reports returned from the receiver. This is either due to a failure of the receiver to send RTCP reports, or a failure of the return path that is preventing those RTCP reporting from being delivered. In either case, it is not safe to continue transmission, since the sender has no way of knowing if it is causing congestion. Accordingly, an RTP sender that has not received any RTCP SR or RTCP RR packets reporting on the SSRC it is using for three or more RTCP reporting intervals SHOULD cease transmission (see Section 4.4). When calculating the timeout, the fixed minimum RTCP reporting interval SHOULD be used (based on the rationale in Section 6.2 of RFC 3550 [RFC3550]).

The choice of three RTCP reporting intervals as the timeout is made following Section 6.3.5 of RFC 3550 [RFC3550]. This specifies that participants in an RTP session will timeout and remove an RTP sender from the list of active RTP senders if no RTP data packets have been received from that RTP sender within the last two RTCP reporting intervals. Using a timeout of three RTCP reporting intervals is therefore large enough that the other participants will have timed out the sender if a network problem stops the data packets it is sending from reaching the receivers, even allowing for loss of some RTCP packets.

If a sender is transmitting a large number of RTP media streams, such that the corresponding RTCP SR or RR packets are too large to fit



into the network MTU, this will force the receiver to generate RTCP SR or RR packets in a round-robin manner. In this case, the sender MAY treat receipt of an RTCP SR or RR packet corresponding to an SSRC it sent using the same 5-tuple of source and destination IP address, port, and protocol, as an indication that the receiver and return path are working to prevent the RTCP timeout circuit breaker from triggering.

#### 4.3. RTP/AVP Circuit Breaker #3: Congestion

If RTP data packets are being sent, and the corresponding RTCP SR or RR packets show non-zero packet loss fraction and increasing extended highest sequence number received, then those RTP data packets are arriving at the receiver, but some degree of congestion is occurring. The RTP/AVP profile [RFC3551] states that:

If best-effort service is being used, RTP receivers SHOULD monitor packet loss to ensure that the packet loss rate is within acceptable parameters. Packet loss is considered acceptable if a TCP flow across the same network path and experiencing the same network conditions would achieve an average throughput, measured on a reasonable time scale, that is not less than the RTP flow is achieving. This condition can be satisfied by implementing congestion control mechanisms to adapt the transmission rate (or the number of layers subscribed for a layered multicast session), or by arranging for a receiver to leave the session if the loss rate is unacceptably high.

The comparison to TCP cannot be specified exactly, but is intended as an "order-of-magnitude" comparison in time scale and throughput. The time scale on which TCP throughput is measured is the round-trip time of the connection. In essence, this requirement states that it is not acceptable to deploy an application (using RTP or any other transport protocol) on the best-effort Internet which consumes bandwidth arbitrarily and does not compete fairly with TCP within an order of magnitude.

The phrase "order of magnitude" in the above means within a factor of ten, approximately. In order to implement this, it is necessary to estimate the throughput a TCP connection would achieve over the path. For a long-lived TCP Reno connection, Padhye et al. [Padhye] showed that the throughput can be estimated using the following equation:

$$X = \frac{s}{R \sqrt{2 * b * p / 3} + (t_{RTO} * (3 * \sqrt{3 * b * p / 8} * p * (1 + 32 * p^2)))}$$

where:

X is the transmit rate in bytes/second.

s is the packet size in bytes. If data packets vary in size, then the average size is to be used.

R is the round trip time in seconds.

p is the loss event rate, between 0 and 1.0, of the number of loss events as a fraction of the number of packets transmitted.

t\_RTO is the TCP retransmission timeout value in seconds, approximated by setting  $t\_RTO = 4 * R$ .

b is the number of packets acknowledged by a single TCP acknowledgement ([RFC3448] recommends the use of  $b=1$  since many TCP implementations do not use delayed acknowledgements).

This is the same approach to estimated TCP throughput that is used in [RFC3448]. Under conditions of low packet loss, this formula can be approximated as follows with reasonable accuracy:

$$X = \frac{s}{R * \sqrt{p*2/3}}$$

It is RECOMMENDED that this simplified throughput equation be used, since the reduction in accuracy is small, and it is much simpler to calculate than the full equation.

Given this TCP equation, two parameters need to be estimated and reported to the sender in order to calculate the throughput: the round trip time, R, and the loss event rate, p (the packet size, s, is known to the sender). The round trip time can be estimated from RTCP SR and RR packets. This is done too infrequently for accurate statistics, but is the best that can be done with the standard RTCP mechanisms.

Report blocks in RTCP SR or RR packets contain the packet loss fraction, rather than the loss event rate, so p cannot be reported (TCP typically treats the loss of multiple packets within a single RTT as one loss event, but RTCP RR packets report the overall fraction of packets lost, not caring about when the losses occurred). Using the loss fraction in place of the loss event rate can overestimate the loss. We believe that this overestimate will not be significant, given that we are only interested in order of magnitude

comparison ([Floyd] section 3.2.1 shows that the difference is small for steady-state conditions and random loss, but using the loss fraction is more conservative in the case of bursty loss).

The congestion circuit breaker is therefore: when a sender receives an RTCP SR or RR packet that contains a report block for an SSRC it is using, that sender has to check the fraction lost field in that report block to determine if there is a non-zero packet loss rate. If the fraction lost field is zero, then continue sending as normal. If the fraction lost is greater than zero, then estimate the TCP throughput using the simplified equation above, and the measured  $R$ ,  $p$  (approximated by the fraction lost), and  $s$ . Compare this with the actual sending rate. If the actual sending rate is more than ten times the estimated sending rate derived from the TCP throughput equation for two consecutive RTCP reporting intervals, the sender SHOULD cease transmission (see Section 4.4). Systems that usually send at a high data rate, but that can reduce their data rate significantly (i.e., by at least a factor of ten), MAY first reduce their sending rate to this lower value to see if this resolves the congestion, but MUST then cease transmission if the problem does not resolve itself within a further two RTCP reporting intervals (see Section 4.4). An example of this might be a video conferencing system that backs off to sending audio only, before completely dropping the call. If such a reduction in sending rate resolves the congestion problem, the sender MAY gradually increase the rate at which it sends data after a reasonable amount of time has passed, provided it takes care not to cause the problem to recur ("reasonable" is intentionally not defined here).

If the incoming RTCP SR or RR packets are using a reduced minimum RTCP reporting interval (as specified in Section 6.2 of RFC 3550 [RFC3550] or the RTP/AVPF profile [RFC4585]), then that reduced RTCP reporting interval is used when determining if the circuit breaker is triggered. The RTCP reporting interval of the media sender does not affect how quickly congestion circuit breaker can trigger. The timing is based on the RTCP reporting interval of the receiver that matters (note that RTCP requires all participants in a session to have similar reporting intervals, else the participant timeout rules in [RFC3550] will not work).

As in Section 4.1, we use two reporting intervals to avoid triggering the circuit breaker on transient failures. This circuit breaker is a worst-case condition, and congestion control needs to be performed to keep well within this bound. It is expected that the circuit breaker will only be triggered if the usual congestion control fails for some reason.

If there are more media streams that can be reported in a single RTCP SR or RR packet, or if the size of a complete RTCP SR or RR packet exceeds the network MTU, then the receiver will report on a subset of sources in each reporting interval, with the subsets selected round-robin across multiple intervals so that all sources are eventually reported [RFC3550]. When generating such round-robin RTCP reports, priority SHOULD be given to reports on sources that have high packet loss rates, to ensure that senders are aware of network congestion they are causing (this is an update to [RFC3550]).

#### 4.4. Ceasing Transmission

What it means to cease transmission depends on the application, but the intention is that the application will stop sending RTP data packets to a particular destination 3-tuple (transport protocol, destination port, IP address), until the user makes an explicit attempt to restart the call. It is important that a human user is involved in the decision to try to restart the call, since that user will eventually give up if the calls repeatedly trigger the circuit breaker. This will help avoid problems with automatic redial systems from congesting the network. Accordingly, RTP flows halted by the circuit breaker SHOULD NOT be restarted automatically unless the sender has received information that the congestion has dissipated.

It is recognised that the RTP implementation in some systems might not be able to determine if a call set-up request was initiated by a human user, or automatically by some scripted higher-level component of the system. These implementations SHOULD rate limit attempts to restart a call to the same destination 3-tuple as used by a previous call that was recently halted by the circuit breaker. The chosen rate limit ought to not exceed the rate at which an annoyed human caller might redial a misbehaving phone.

#### 5. RTP Circuit Breakers for Systems Using the RTP/AVPF Profile

Use of the Extended RTP Profile for RTCP-based Feedback (RTP/AVPF) [RFC4585] allows receivers to send early RTCP reports in some cases, to inform the sender about particular events in the media stream. There are several use cases for such early RTCP reports, including providing rapid feedback to a sender about the onset of congestion.

Receiving rapid feedback about congestion events potentially allows congestion control algorithms to be more responsive, and to better adapt the media transmission to the limitations of the network. It is expected that many RTP congestion control algorithms will adopt the RTP/AVPF profile for this reason, defining new transport layer feedback reports that suit their requirements. Since these reports are not yet defined, and likely very specific to the details of the

congestion control algorithm chosen, they cannot be used as part of the generic RTP circuit breaker.

If the extension for Reduced-Size RTCP [RFC5506] is not used, early RTCP feedback packets sent according to the RTP/AVPF profile will be compound RTCP packets that include an RTCP SR/RR packet. That RTCP SR/RR packet MUST be processed as if it were sent as a regular RTCP report and counted towards the circuit breaker conditions specified in Section 4 of this memo. This will potentially make the RTP circuit breaker fire earlier than it would if the RTP/AVPF profile was not used.

Reduced-size RTCP reports sent under the RTP/AVPF early feedback rules that do not contain an RTCP SR or RR packet MUST be ignored by the RTP circuit breaker (they do not contain the information used by the circuit breaker algorithm). Reduced-size RTCP reports sent under the RTP/AVPF early feedback rules that contain RTCP SR or RR packets MUST be processed as if they were sent as regular RTCP reports, and counted towards the circuit breaker conditions specified in Section 4 of this memo. This will potentially make the RTP circuit breaker fire earlier than it would if the RTP/AVPF profile was not used.

When using ECN with RTP (see Section 8), early RTCP feedback packets can contain ECN feedback reports. The count of ECN-CE marked packets contained in those ECN feedback reports is counted towards the number of lost packets reported if the ECN Feedback Report report is sent in an compound RTCP packet along with an RTCP SR/RR report packet. Reports of ECN-CE packets sent as reduced-size RTCP ECN feedback packets without an RTCP SR/RR packet MUST be ignored.

These rules are intended to allow the use of low-overhead early RTP/AVPF feedback for generic NACK messages without triggering the RTP circuit breaker. This is expected to make such feedback suitable for RTP congestion control algorithms that need to quickly report loss events in between regular RTCP reports. The reaction to reduced-size RTCP SR/RR packets is to allow such algorithms to send feedback that can trigger the circuit breaker, when desired.

## 6. Impact of RTCP XR

RTCP Extended Report (XR) blocks provide additional reception quality metrics, but do not change the RTCP timing rules. Some of the RTCP XR blocks provide information that might be useful for congestion control purposes, others provided non-congestion-related metrics. With the exception of RTCP XR ECN Summary Reports (see Section 8), the presence of RTCP XR blocks in a compound RTCP packet does not affect the RTP circuit breaker algorithm. For consistency and ease of implementation, only the reception report blocks contained in RTCP

SR packets, RTCP RR packets, or RTCP XR ECN Summary Report packets, are used by the RTP circuit breaker algorithm.

## 7. Impact of RTCP Reporting Groups

An optimisation for grouping RTCP reception statistics and other feedback in RTP sessions with large numbers of participants is given in [I-D.ietf-avtcore-rtp-multi-stream-optimisation]. This allows one SSRC to act as a representative that sends reports on behalf of other SSRCs that are co-located in the same endpoint and see identical reception quality. When running the circuit breaker algorithms, an endpoint MUST treat a reception report from the representative of the reporting group as if a reception report was received from all members of that group.

## 8. Impact of Explicit Congestion Notification (ECN)

The use of ECN for RTP flows does not affect the media timeout RTP circuit breaker (Section 4.1) or the RTCP timeout circuit breaker (Section 4.2), since these are both connectivity checks that simply determinate if any packets are being received.

ECN-CE marked packets SHOULD be treated as if it were lost for the purposes of congestion control, when determining the optimal media sending rate for an RTP flow. If an RTP sender has negotiated ECN support for an RTP session, and has successfully initiated ECN use on the path to the receiver [RFC6679], then ECN-CE marked packets SHOULD be treated as if they were lost when calculating if the congestion-based RTP circuit breaker (Section 4.3) has been met. The count of ECN-CE marked RTP packets is returned in RTCP XR ECN summary report packets if support for ECN has been initiated for an RTP session.

## 9. Security Considerations

The security considerations of [RFC3550] apply.

If the RTP/AVPF profile is used to provide rapid RTCP feedback, the security considerations of [RFC4585] apply. If ECN feedback for RTP over UDP/IP is used, the security considerations of [RFC6679] apply.

If non-authenticated RTCP reports are used, an on-path attacker can trivially generate fake RTCP packets that indicate high packet loss rates, causing the circuit breaker to trigger and disrupting an RTP session. This is somewhat more difficult for an off-path attacker, due to the need to guess the randomly chosen RTP SSRC value and the RTP sequence number. This attack can be avoided if RTCP packets are authenticated, for example using the Secure RTP profile [RFC3711].

## 10. IANA Considerations

There are no actions for IANA.

## 11. Acknowledgements

The authors would like to thank Bernard Aboba, Harald Alvestrand, Kevin Gross, Cullen Jennings, Randell Jesup, Jonathan Lennox, Matt Mathis, Stephen McQuistin, Eric Rescorla, and Abheek Saha for their valuable feedback.

## 12. References

### 12.1. Normative References

- [RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, March 1997.
- [RFC3448] Handley, M., Floyd, S., Padhye, J., and J. Widmer, "TCP Friendly Rate Control (TFRC): Protocol Specification", RFC 3448, January 2003.
- [RFC3550] Schulzrinne, H., Casner, S., Frederick, R., and V. Jacobson, "RTP: A Transport Protocol for Real-Time Applications", STD 64, RFC 3550, July 2003.
- [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.
- [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.

### 12.2. Informative References

- [Floyd] Floyd, S., Handley, M., Padhye, J., and J. Widmer, "Equation-Based Congestion Control for Unicast Applications", Proc. ACM SIGCOMM 2000, DOI 10.1145/347059.347397, August 2000.

[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-00 (work  
in progress), July 2013.

[I-D.ietf-xrblock-rtcp-xr-burst-gap-discard]

Clark, A., Huang, R., and W. Wu, "RTP Control  
Protocol (RTCP) Extended Report (XR) Block for Burst/Gap  
Discard metric Reporting", draft-ietf-xrblock-rtcp-xr-  
burst-gap-discard-14 (work in progress), April 2013.

[I-D.ietf-xrblock-rtcp-xr-burst-gap-loss]

Clark, A., Zhang, S., Zhao, J., and W. Wu, "RTP Control  
Protocol (RTCP) Extended Report (XR) Block for Burst/Gap  
Loss metric Reporting", draft-ietf-xrblock-rtcp-xr-burst-  
gap-loss-12 (work in progress), April 2013.

[I-D.ietf-xrblock-rtcp-xr-discard-rle-metrics]

Ott, J., Singh, V., and I. Curcio, "RTP Control Protocol  
(RTCP) Extended Reports (XR) for Run Length Encoding (RLE)  
of Discarded Packets", draft-ietf-xrblock-rtcp-xr-discard-  
rle-metrics-06 (work in progress), July 2013.

[I-D.ietf-xrblock-rtcp-xr-discard]

Clark, A., Zorn, G., and W. Wu, "RTP Control Protocol  
(RTCP) Extended Report (XR) Block for Discard Count metric  
Reporting", draft-ietf-xrblock-rtcp-xr-discard-15 (work in  
progress), June 2013.

[Padhye]

Padhye, J., Firoiu, V., Towsley, D., and J. Kurose,  
"Modeling TCP Throughput: A Simple Model and its Empirical  
Validation", Proc. ACM SIGCOMM 1998, DOI 10.1145/  
285237.285291, August 1998.

[RFC3168]

Ramakrishnan, K., Floyd, S., and D. Black, "The Addition  
of Explicit Congestion Notification (ECN) to IP", RFC  
3168, September 2001.

[RFC3711]

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

[RFC5104]

Wenger, S., Chandra, U., Westerlund, M., and B. Burman,  
"Codec Control Messages in the RTP Audio-Visual Profile  
with Feedback (AVPF)", RFC 5104, February 2008.



- [RFC5450] Singer, D. and H. Desineni, "Transmission Time Offsets in RTP Streams", RFC 5450, March 2009.
- [RFC5506] Johansson, I. and M. Westerlund, "Support for Reduced-Size Real-Time Transport Control Protocol (RTCP): Opportunities and Consequences", RFC 5506, April 2009.
- [RFC6051] Perkins, C. and T. Schierl, "Rapid Synchronisation of RTP Flows", RFC 6051, November 2010.
- [RFC6679] Westerlund, M., Johansson, I., Perkins, C., O'Hanlon, P., and K. Carlberg, "Explicit Congestion Notification (ECN) for RTP over UDP", RFC 6679, August 2012.
- [RFC6798] Clark, A. and Q. Wu, "RTP Control Protocol (RTCP) Extended Report (XR) Block for Packet Delay Variation Metric Reporting", RFC 6798, November 2012.
- [RFC6843] Clark, A., Gross, K., and Q. Wu, "RTP Control Protocol (RTCP) Extended Report (XR) Block for Delay Metric Reporting", RFC 6843, January 2013.

#### Authors' Addresses

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

Email: [csp@csp Perkins.org](mailto:csp@csp Perkins.org)

Varun Singh  
Aalto University  
School of Electrical Engineering  
Otakaari 5 A  
Espoo, FIN 02150  
Finland

Email: [varun@comnet.tkk.fi](mailto:varun@comnet.tkk.fi)  
URI: <http://www.netlab.tkk.fi/~varun/>

AVTCORE  
Internet-Draft  
Updates: 3550 (if approved)  
Intended status: Standards Track  
Expires: January 12, 2014

J. Lennox  
Vidyo  
M. Westerlund  
Ericsson  
Q. Wu  
Huawei  
C. Perkins  
University of Glasgow  
July 11, 2013

Sending Multiple Media Streams in a Single RTP Session  
draft-ietf-avtccore-rtp-multi-stream-01

Abstract

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

This document updates RFC 3550 in regards to handling of multiple SSRCs per endpoint in RTP sessions.

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 January 12, 2014.

Copyright Notice

Copyright (c) 2013 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 . . . . .	2
2. Terminology . . . . .	3
3. Use Cases For Multi-Stream Endpoints . . . . .	3
3.1. Multiple-Capturer Endpoints . . . . .	3
3.2. Multi-Media Sessions . . . . .	3
3.3. Multi-Stream Mixers . . . . .	4
4. Multi-Stream Endpoint RTP Media Recommendations . . . . .	4
5. Multi-Stream Endpoint RTCP Recommendations . . . . .	4
5.1. RTCP Reporting Requirement . . . . .	5
5.2. Initial Reporting Interval . . . . .	5
5.3. Compound RTCP Packets . . . . .	5
6. RTCP Considerations for Streams with Disparate Rates . . . . .	7
6.1. Timing out SSRCS . . . . .	8
6.2. Tuning RTCP transmissions . . . . .	9
7. Security Considerations . . . . .	11
8. Open Issues . . . . .	12
9. IANA Considerations . . . . .	12
10. References . . . . .	12
10.1. Normative References . . . . .	12
10.2. Informative References . . . . .	13
Appendix A. Changes From Earlier Versions . . . . .	14
A.1. Changes From WG Draft -00 . . . . .	14
A.2. Changes From Individual Draft -02 . . . . .	14
A.3. Changes From Individual Draft -01 . . . . .	14
A.4. Changes From Individual Draft -00 . . . . .	14
Authors' Addresses . . . . .	15

## 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 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 starts with terminology and some use cases where multiple sources will occur. This is followed by some case studies to try to identify issues that exist and need considerations. 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 send multiple streams in a single RTP session.

### 3.1. Multiple-Capturer Endpoints

The most straightforward motivation for an endpoint to send multiple media streams in a session is the scenario where an endpoint has multiple capture devices 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] is designed, 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 streams of different media types would always be sent on different RTP sessions. In this work, a single endpoint's audio and video 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 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 Source Projecting Mixer, see Section 3.6 of [I-D.ietf-avtcore-rtp-topologies-update]. This is a central box 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 mixer, "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.

## 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. In particular, session bandwidth MAY be reallocated among an endpoint's media streams, 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 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 media streams, rather than to endpoints. This section describes several concrete cases where this applies.

(tbd: rather than think in terms of media streams, it might be clearer to refer to SSRC values, where a participant with multiple active SSRC values counts as multiple participants, once per SSRC)

#### 5.1. RTCP Reporting Requirement

For each of an endpoint's media streams, whether or not it is 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 media stream 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 media sources 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 depends on the number of source to send initial SR or RR and Session Description CNAME items for 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 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 (SDS), 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 SDS packet containing an CNAME item. Thus the above results in compound RTCP packets that contain multiple SR or RR packets from different sources 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.)

Due to RTCP's randomization of reporting times, there is a fair bit of tolerance in precisely when an endpoint schedules RTCP to be sent. Thus, one potential way of implementing this recommendation would be to randomize all of an endpoint's sources together, with a single randomization schedule, so an MTU's worth of RTCP all comes out simultaneously.

(tbd: Multiplexing RTCP packets from multiple different sources might require some adjustment to the calculation of RTCP's avg\_rtcp\_size, as the RTCP group interval is proportional to avg\_rtcp\_size times the group 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 scenario 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  $5 \cdot T_d$ . The  $T_d$  calculation SHALL be done using a  $T_{min}$  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.  $T_{min}$  in the  $T_d$  calculation is the `T_rr_interval`.

Note: 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/AVP endpoints will prematurely timeout the RTP/AVPF endpoints due to their different RTCP timeout intervals. Since an RTP session can only use a single RTP profile, this issue ought never occur. If such mixed RTP profiles are used, however, the RTP/AVPF session MUST NOT use a non-zero `T_rr_interval` that is smaller than 5 seconds.

(tbd: it has been suggested that a minimum non-zero  $T_{rr\_interval}$  of 4 seconds is more appropriate, due to the nature of the timing rules).

## 6.2. Tuning RTCP transmissions

This sub-section discusses what tuning can be done to reduce the downsides of the shared RTCP packet intervals.

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  $T_d$ . 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  $T_d$  prior to randomization and reconsideration can never be less than 5 seconds (assuming default  $T_{min}$  of 5 seconds).
- b. If the scaled minimum RTCP interval is used,  $T_d$  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  $T_{min}$  being 5 seconds. An RTP session bandwidth of 360 kbps of course gives a  $T_{min}$  of 1 second, and to achieve a  $T_{min}$  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  $T_{min}$ , 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  $T_d$  value equal to  $T_{min}$  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  $n \geq 32$ . 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  $T_d$  value is  $[0.5 \cdot T_d / 1.21828, 1.5 \cdot T_d / 1.21828]$ , which means that for  $T_d = 5$  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  $\leq X$  within the interval with an uniform probability distribution. This results in that the majority of the probability mass is above the  $T_d$  value.

To conclude, with RTP/AVP and RTP/SAVP the key limitation for small unicast sessions is going to be the  $T_{min}$  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.

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  $T_{min}$  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 as hard as no regular RTCP packet can be sent less than  $T_{rr\_interval}$  after the previous regular packet.

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 SRTP 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 use 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. Further clarifications on how to handle the RTCP scheduler when sending multiple sources in one compound packet.
2. How is the RTCP avg\_rtcp\_size be calculated when RTCP packets are routinely multiplexed among multiple RTCP senders?
3. 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?

## 9. IANA Considerations

No IANA actions needed.

## 10. References

### 10.1. Normative References

- [I-D.ietf-avtcore-6222bis]  
Begen, A., Perkins, C., Wing, D., and E. Rescorla,  
"Guidelines for Choosing RTP Control Protocol (RTCP)  
Canonical Names (CNAMEs)", draft-ietf-avtcore-6222bis-04  
(work in progress), June 2013.
- [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, "Multiple Media Types in an RTP Session", draft-ietf-avtcore-multi-media-rtp-session-02 (work in progress), February 2013.
- [I-D.ietf-avtcore-rtp-multi-stream-optimisation]  
Lennox, J., Westerlund, M., Wu, Q., 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-00 (work in progress), July 2013.
- [I-D.ietf-avtcore-rtp-topologies-update]  
Westerlund, M. and S. Wenger, "RTP Topologies", draft-ietf-avtcore-rtp-topologies-update-00 (work in progress), April 2013.
- [I-D.ietf-clue-framework]  
Duckworth, M., Pepperell, A., and S. Wenger, "Framework for Telepresence Multi-Streams", draft-ietf-clue-framework-10 (work in progress), May 2013.
- [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-04 (work in progress), June 2013.

- [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.

#### 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 -00

- o Split the Reporting Group Extension from this draft into draft-ietf-avtcore-rtp-multi-stream-optimization-00.
- o Added RTCP tuning considerations from draft-ietf-avtcore-multi-media-rtp-session-02.

##### A.2. Changes From Individual Draft -02

- o Resubmitted as working group draft.
- o Updated references.

##### A.3. Changes From Individual Draft -01

- o Merged with draft-wu-avtcore-multisrc-endpoint-adver.
- o Changed how Reporting Groups are indicated in RTCP, to make it clear which source(s) is the group's reporting sources.
- o Clarified the rules for when sources can be placed in the same reporting group.
- o 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.4. Changes From Individual Draft -00

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

- o Specified that Reporting Group semantics (as they now are) apply to AVPF and XR, as well as to RR/SR report blocks.
- o Added a description of the cascaded source-projecting mixer, along with a calculation of its RTCP overhead if reporting groups are not in use.
- o Gave some guidance on how the flexibility of RTCP randomization allows some freedom in RTCP multiplexing.
- o Clarified the language of several of the recommendations.
- o Added an open issue discussing how avg\_rtcp\_size ought to be calculated for multiplexed RTCP.
- o 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](mailto:jonathan@vidyo.com)

Magnus Westerlund  
Ericsson  
Farogatan 6  
SE-164 80 Kista  
Sweden

Phone: +46 10 714 82 87  
Email: [magnus.westerlund@ericsson.com](mailto:magnus.westerlund@ericsson.com)

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

Email: [sunseawq@huawei.com](mailto:sunseawq@huawei.com)



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

Email: [csp@csperrkins.org](mailto:csp@csperrkins.org)

AVTCORE WG  
Internet-Draft  
Updates: 3550 (if approved)  
Intended status: Standards Track  
Expires: January 12, 2014

J. Lennox  
Vidyo  
M. Westerlund  
Ericsson  
Q. Wu  
Huawei  
C. Perkins  
University of Glasgow  
July 11, 2013

Sending Multiple Media Streams in a Single RTP Session: Grouping RTCP  
Reception Statistics and Other Feedback  
draft-ietf-avtccore-rtp-multi-stream-optimisation-00

Abstract

RTP allows multiple media streams to be sent in a single session, but requires each Synchronisation Source (SSRC) to send RTCP reception quality reports for every other SSRC visible in the session. This causes the number of RTCP reception reports to grow with the number of SSRCs, rather than the number of endpoints. In many cases most of these RTCP reception reports are unnecessary, since all SSRCs of an endpoint are co-located and see the same reception quality. This memo defines a Reporting Group extension to RTCP to reduce the reporting overhead in such scenarios.

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 January 12, 2014.

Copyright Notice

Copyright (c) 2013 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 . . . . .	2
2. Terminology . . . . .	3
3. Grouping of RTCP Reception Statistics and Other Feedback . .	3
3.1. Semantics and Behavior of Reporting Groups . . . . .	3
3.2. Determine the Report Group . . . . .	4
3.3. RTCP Packet Reporting Group's Reporting Sources . . . . .	5
3.4. RTCP Source Description (SDES) item for Reporting Groups	6
3.5. Middlebox Considerations . . . . .	6
3.6. SDP signaling for Reporting Groups . . . . .	6
3.7. Bandwidth Benefits of RTCP Reporting Groups . . . . .	6
3.8. Consequences of RTCP Reporting Groups . . . . .	7
4. Security Considerations . . . . .	8
5. IANA Considerations . . . . .	8
6. References . . . . .	8
6.1. Normative References . . . . .	8
6.2. Informative References . . . . .	9
Authors' Addresses . . . . .	9

## 1. Introduction

The Real-time Transport Protocol (RTP) [RFC3550] is a protocol for group communication, supporting multiparty multimedia sessions. A single RTP session can support multiple participants sending at once, and can also support participants sending multiple simultaneous media streams. Examples of the latter might include a participant with multiple cameras who chooses to send multiple views of a scene, or a participant that sends audio and video flows multiplexed in a single RTP session. Rules for handling RTP sessions containing multiple media streams are described in [RFC3550] with some clarifications in [I-D.ietf-avtcore-rtp-multi-stream].

An RTP endpoint will have one or more synchronisation sources (SSRCs) that send and receive media streams (it will have one SSRC for each media stream it sends). Each SSRC has to send RTCP sender reports corresponding to the RTP packets it sends, and receiver reports for traffic it receives. That is, every SSRC will send RTCP packets to

report on every other SSRC. This rule is simple, but can be quite inefficient for endpoints that send large numbers of media streams in a single RTP session. Consider a session comprising ten participants, each sending three media streams with their own SSRC. There will be 30 SSRCs in such an RTP session, and 30 RTCP reception reports will be sent per reporting interval as each SSRC reports on all the others. However, the three SSRCs comprising each participant will almost certainly see identical reception quality, since they are co-located. If there was a way to indicate that several SSRCs are co-located, and see the same reception quality, then two-thirds of those RTCP reports could be suppressed.

This memo defines such an RTCP extension, Reporting Groups. This extension is used to indicate the SSRCs that originate from the same endpoint, and therefore have identical reception quality, allowing the endpoint to suppress unnecessary RTCP reception reports.

## 2. Terminology

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in [RFC2119].

## 3. Grouping of RTCP Reception Statistics and Other Feedback

### 3.1. Semantics and Behavior of Reporting Groups

An RTCP Reporting Group indicates that a set of sources (SSRCs) that originate from a single entity (endpoint or middlebox) in an RTP session, and therefore all the sources in the group have an identical view of the network. If reporting groups are in use, two sources SHOULD be put into the same reporting group if their view of the network is identical; i.e., if they report on traffic received at the same interface of an RTP endpoint. Sources with different views of the network MUST NOT be put into the same reporting group.

If reporting groups are in use, an endpoint MUST NOT send reception reports from one source in a reporting group about another one in the same group ("self-reports"). Similarly, an endpoint MUST NOT send reception reports about a remote media source from more than one source in a reporting group ("cross-reports"). Instead, it MUST pick one of its local media sources as the "reporting" source for each remote media source, and use it to send reception reports about that remote source; all the other media sources in the reporting group MUST NOT send any reception reports about that remote media source.

This reporting source MUST also be the source for any RTP/AVPF Feedback [RFC4585] or Extended Report (XR) [RFC3611] packets about

the corresponding remote sources as well. If a reporting source leaves the session (i.e., if it sends a BYE, or leaves the group without sending BYE under the rules of [RFC3550] section 6.3.7), another reporting source MUST be chosen if any members of the group still exist.

An endpoint or middlebox MAY use multiple sources as reporting sources; however, each reporting source MUST have non-overlapping sets of remote SSRCs it reports on. This is primarily to be done when the reporting source's number of reception report blocks is so large that it would be forced to round-robin around the sources. Thus, by splitting the reports among several reporting SSRCs, more consistent reporting can be achieved.

If RTP/AVPF feedback is in use, a reporting source MAY send immediate or early feedback at any point when any member of the reporting group could validly do so.

An endpoint SHOULD NOT create single-source reporting groups, unless it is anticipated that the group might have additional sources added to it in the future.

### 3.2. Determine the Report Group

A remote RTP entity, such as an endpoint or a middlebox needs to be able to determine the report group used by another RTP entity. To achieve this goal two RTCP extensions have been defined. For the SSRCs that are reporting on behalf of the reporting group, an SDES item RGRP has been defined for providing the report group with an identifier. For SSRCs that aren't reporting on any peer SSRC a new RTCP packet type is defined. This RTCP packet type "Reporting Sources" lists the SSRC that are reporting on this SSRC's behalf.

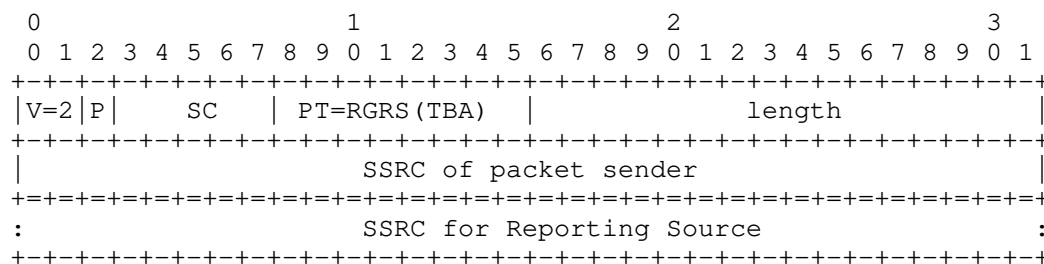
This divided approach has been selected for the following reasons:

1. To enable an explicit indication of who reports on this SSRC's behalf. Being explicit prevents the remote entity from detecting that is missing the reports if there issues with the reporting SSRC's RTCP packets.
2. To enable explicit identification of the SSRCs that are actively reporting as one entity.

### 3.3. RTCP Packet Reporting Group's Reporting Sources

This section defines a new RTCP packet type called "Reporting Group's Reporting Sources" (RGRS). It identifies the SSRC(s) that report on behalf of the SSRC that is the sender of the RGRS packet.

This packet consists of the fixed RTCP packet header which indicates the packet type, the number of reporting sources included and the SSRC which behalf the reporting SSRCs report on. This is followed by the list of reporting SSRCs.



The RTCP Packets field has the following definition.

version (V): This field identifies the RTP version. The current version is 2.

padding (P): 1 bit If set, the padding bit indicates that the packet contains additional padding octets at the end that are not part of the control information but are included in the length field. See [RFC3550].

Source Count (SC): 5 bits Indicating the number of reporting SSRCs (1-31) that are included in this RTCP packet type.

Payload type (PT): 8 bits This is the RTCP packet type that identifies the packet as being an RTCP FB message. The RGRS RTCP packet has the value [TBA].

Length: 16 bits The length of this packet in 32-bit words minus one, including the header and any padding. This is in line with the definition of the length field used in RTCP sender and receiver reports [RFC3550].

SSRC of packet sender: 32 bits. The SSRC of the sender of this packet which indicates which SSRCs that reports on its behalf, instead of reporting itself.

SSRC for Reporting Source: A variable number (as indicated by Source Count) of 32-bit SSRC values. Each SSRC is an reporting SSRC belonging to the same Report Group.

Each RGRS packet MUST contain at least one reporting SSRC. In case the reporting SSRC field is insufficient to list all the SSRCs that are reporting in this report group, the SSRC SHALL round robin around the reporting sources.

Any RTP mixer or translator which forwards SR or RR packets from members of a reporting group MUST forward the corresponding RGRS RTCP packet as well.

#### 3.4. RTCP Source Description (SDES) item for Reporting Groups

A new RTCP Source Description (SDES) item is defined for the purpose of identifying reporting groups.

The Source Description (SDES) item "RGRP" is sent by a reporting group's reporting SSRC. Syntactically, its format is the same as the RTCP SDES CNAME item [RFC6222], and MUST be chosen with the same global-uniqueness and privacy considerations as CNAME. This name MUST be stable across the lifetime of the reporting group, even if the SSRC of a reporting source changes.

Every source which belongs to a reporting group MUST either include an RGRP SDSE item in an SDSE packet (if it is a reporting source), or an RGRS packet (if it is not), in every compound RTCP packet in which it sends an RR or SR packet (i.e., in every RTCP packet it sends, unless Reduced-Size RTCP [RFC5506] is in use).

Any RTP mixer or translator which forwards SR or RR packets from members of a reporting group MUST forward the corresponding RGRP SDSE items as well, even if it otherwise strips SDSE items other than CNAME.

#### 3.5. Middlebox Considerations

This section discusses middlebox considerations for Reporting groups.

To be expanded.

#### 3.6. SDP signaling for Reporting Groups

TBD

#### 3.7. Bandwidth Benefits of RTCP Reporting Groups

To understand the benefits of RTCP reporting groups, consider a scenario in which the two endpoints in a session each have a hundred sources, of which eight each are sending within any given reporting interval.

For ease of analysis, we can make the simplifying approximation that the duration of the RTCP reporting interval is equal to the total size of the RTCP packets sent during an RTCP interval, divided by the RTCP bandwidth. (This will be approximately true in scenarios where the bandwidth is not so high that the minimum RTCP interval is reached.) For further simplification, we can assume RTCP senders are following the recommendations regarding Compound RTCP Packets in [I-D.ietf-avtcore-rtp-multi-stream]; thus, the per-packet transport-layer overhead will be small relative to the RTCP data. Thus, only the actual RTCP data itself need be considered.

In a report interval in this scenario, there will, as a baseline, be 200 SDES packets, 184 RR packets, and 16 SR packets. This amounts to approximately 6.5 kB of RTCP per report interval, assuming 16-byte CNAMEs and no other SDES information.

Using the original [RFC3550] everyone-reports-on-every-sender feedback rules, each of the 184 receivers will send 16 report blocks, and each of the 16 senders will send 15. This amounts to approximately 76 kB of report block traffic per interval; 92% of RTCP traffic consists of report blocks.

If reporting groups are used, however, there is only 0.4 kB of reports per interval, with no loss of useful information. Additionally, there will be (assuming 16-byte RGRPs, and a single reporting source per reporting group) an additional 2.4 kB per cycle of RGRP SDES items and RGRS packets. Put another way, the unmodified [RFC3550] reporting interval is approximately 8.9 times longer than if reporting groups are in use.

### 3.8. Consequences of RTCP Reporting Groups

The RTCP traffic generated by receivers using RTCP Reporting Groups might appear, to observers unaware of these semantics, to be generated by receivers who are experiencing a network disconnection, as the non-reporting sources appear not to be receiving a given sender at all.

This could be a potentially critical problem for such a sender using RTCP for congestion control, as such a sender might think that it is sending so much traffic that it is causing complete congestion collapse.



However, such an interpretation of the session statistics would require a fairly sophisticated RTCP analysis. Any receiver of RTCP statistics which is just interested in information about itself needs to be prepared that any given reception report might not contain information about a specific media source, because reception reports in large conferences can be round-robin.

Thus, it is unclear to what extent such backward compatibility issues would actually cause trouble in practice.

#### 4. Security Considerations

The security considerations of [RFC3550] and [I-D.ietf-avtcore-rtp-multi-stream] apply.

(tbd: any security considerations due to these extensions?)

#### 5. IANA Considerations

(Note to the RFC-Editor: please replace "TBA" with the IANA-assigned value, and "XXXX" with the number of this document, prior to publication as an RFC.)

The IANA is requested to register one new RTCP SDES items in the "RTCP SDES Item Types" registry, as follows:

Value	Abbrev	Name	Reference
TBA	RGRP	Reporting Group	[RFCXXXX]

Figure 1: Item for the IANA Source Attribute Registry

The IANA is also requested to register one new RTCP packet type as follows:

Value	Abbrev	Name	Reference
TBA	RGRR	Reporting Group Reporting Sources	[RFCXXXX]

Figure 2: Item for the IANA RTCP Control Packet Types (PT) Registry

#### 6. References

##### 6.1. 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.
- [RFC6222] Begen, A., Perkins, C., and D. Wing, "Guidelines for Choosing RTP Control Protocol (RTCP) Canonical Names (CNAMEs)", RFC 6222, April 2011.

## 6.2. Informative References

- [I-D.ietf-avtcore-rtp-multi-stream]  
Lennox, J., Westerlund, M., Wu, W., and C. Perkins, "RTP Considerations for Endpoints Sending Multiple Media Streams", draft-ietf-avtcore-rtp-multi-stream-00 (work in progress), April 2013.
- [RFC3611] Friedman, T., Caceres, R., and A. Clark, "RTP Control Protocol Extended Reports (RTCP XR)", RFC 3611, November 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.
- [RFC5506] Johansson, I. and M. Westerlund, "Support for Reduced-Size Real-Time Transport Control Protocol (RTCP): Opportunities and Consequences", RFC 5506, April 2009.

## Authors' Addresses

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

Email: [jonathan@vidyo.com](mailto: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

Network Working Group  
Internet-Draft  
Intended status: Standards Track  
Expires: February 03, 2014

J. Spittka  
  
K. Vos  
Skype Technologies S.A.  
JM. Valin  
Mozilla  
August 02, 2013

RTP Payload Format for Opus Speech and Audio Codec  
draft-ietf-payload-rtp-opus-01

Abstract

This document defines the Real-time Transport Protocol (RTP) payload format for packetization of Opus encoded speech and audio data that is essential to integrate the codec in the most compatible way. Further, media type registrations are described for the RTP payload format.

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 February 03, 2014.

Copyright Notice

Copyright (c) 2013 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 . . . . .	2
2. Conventions, Definitions and Acronyms used in this document .	3
2.1. Audio Bandwidth . . . . .	3
3. Opus Codec . . . . .	3
3.1. Network Bandwidth . . . . .	4
3.1.1. Recommended Bitrate . . . . .	4
3.1.2. Variable versus Constant Bit Rate . . . . .	4
3.1.3. Discontinuous Transmission (DTX) . . . . .	4
3.2. Complexity . . . . .	5
3.3. Forward Error Correction (FEC) . . . . .	5
3.4. Stereo Operation . . . . .	6
4. Opus RTP Payload Format . . . . .	6
4.1. RTP Header Usage . . . . .	6
4.2. Payload Structure . . . . .	7
5. Congestion Control . . . . .	8
6. IANA Considerations . . . . .	9
6.1. Opus Media Type Registration . . . . .	9
6.2. Mapping to SDP Parameters . . . . .	13
6.2.1. Offer-Answer Model Considerations for Opus . . . . .	14
6.2.2. Declarative SDP Considerations for Opus . . . . .	16
7. Security Considerations . . . . .	16
8. Acknowledgements . . . . .	16
9. Normative References . . . . .	17
Authors' Addresses . . . . .	18

## 1. Introduction

The Opus codec is a speech and audio codec developed within the IETF Internet Wideband Audio Codec working group (codec). The codec has a very low algorithmic delay and it is highly scalable in terms of audio bandwidth, bitrate, and complexity. Further, it provides different modes to efficiently encode speech signals as well as music signals, thus, making it the codec of choice for various applications using the Internet or similar networks.

This document defines the Real-time Transport Protocol (RTP) [RFC3550] payload format for packetization of Opus encoded speech and audio data that is essential to integrate the Opus codec in the most compatible way. Further, media type registrations are described for the RTP payload format. More information on the Opus codec can be obtained from [RFC6716].

## 2. Conventions, Definitions and Acronyms used in this document

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in [RFC2119].

CBR: Constant bitrate

CPU: Central Processing Unit

DTX: Discontinuous transmission

FEC: Forward error correction

IP: Internet Protocol

samples: Speech or audio samples (usually per channel)

SDP: Session Description Protocol

VBR: Variable bitrate

### 2.1. Audio Bandwidth

Throughout this document, we refer to the following definitions:

Abbreviation	Name	Bandwidth	Sampling
nb	Narrowband	0 - 4000	8000
mb	Mediumband	0 - 6000	12000
wb	Wideband	0 - 8000	16000
swb	Super-wideband	0 - 12000	24000
fb	Fullband	0 - 20000	48000

Audio bandwidth naming

Table 1

## 3. Opus Codec

The Opus [RFC6716] speech and audio codec has been developed to encode speech signals as well as audio signals. Two different modes, a voice mode or an audio mode, may be chosen to allow the most efficient coding dependent on the type of input signal, the sampling frequency of the input signal, and the specific application.

The voice mode allows efficient encoding of voice signals at lower bit rates while the audio mode is optimized for audio signals at medium and higher bitrates.

The Opus speech and audio codec is highly scalable in terms of audio bandwidth, bitrate, and complexity. Further, Opus allows transmitting stereo signals.

### 3.1. Network Bandwidth

Opus supports all bitrates from 6 kb/s to 510 kb/s. The bitrate can be changed dynamically within that range. All other parameters being equal, higher bitrate results in higher quality.

#### 3.1.1. Recommended Bitrate

For a frame size of 20 ms, these are the bitrate "sweet spots" for Opus in various configurations:

- o 8-12 kb/s for NB speech,
- o 16-20 kb/s for WB speech,
- o 28-40 kb/s for FB speech,
- o 48-64 kb/s for FB mono music, and
- o 64-128 kb/s for FB stereo music.

#### 3.1.2. Variable versus Constant Bit Rate

For the same average bitrate, variable bitrate (VBR) can achieve higher quality than constant bitrate (CBR). For the majority of voice transmission application, VBR is the best choice. One potential reason for choosing CBR is the potential information leak that may occur when encrypting the compressed stream. See [RFC6562] for guidelines on when VBR is appropriate for encrypted audio communications. In the case where an existing VBR stream needs to be converted to CBR for security reasons, then the Opus padding mechanism described in [RFC6716] is the RECOMMENDED way to achieve padding because the RTP padding bit is unencrypted.

The bitrate can be adjusted at any point in time. To avoid congestion, the average bitrate SHOULD be adjusted to the available network capacity. If no target bitrate is specified, the bitrates specified in Section 3.1.1 are RECOMMENDED.

#### 3.1.3. Discontinuous Transmission (DTX)

The Opus codec may, as described in Section 3.1.2, be operated with an adaptive bitrate. In that case, the bitrate will automatically be reduced for certain input signals like periods of silence. During continuous transmission the bitrate will be reduced, when the input signal allows to do so, but the transmission to the receiver itself will never be interrupted. Therefore, the received signal will maintain the same high level of quality over the full duration of a transmission while minimizing the average bit rate over time.

In cases where the bitrate of Opus needs to be reduced even further or in cases where only constant bitrate is available, the Opus encoder may be set to use discontinuous transmission (DTX), where parts of the encoded signal that correspond to periods of silence in the input speech or audio signal are not transmitted to the receiver.

On the receiving side, the non-transmitted parts will be handled by a frame loss concealment unit in the Opus decoder which generates a comfort noise signal to replace the non transmitted parts of the speech or audio signal.

The DTX mode of Opus will have a slightly lower speech or audio quality than the continuous mode. Therefore, it is RECOMMENDED to use Opus in the continuous mode unless restraints on network capacity are severe. The DTX mode can be engaged for operation in both adaptive or constant bitrate.

### 3.2. Complexity

Complexity can be scaled to optimize for CPU resources in real-time, mostly as a trade-off between audio quality and bitrate. Also, different modes of Opus have different complexity.

### 3.3. Forward Error Correction (FEC)

The voice mode of Opus allows for "in-band" forward error correction (FEC) data to be embedded into the bit stream of Opus. This FEC scheme adds redundant information about the previous packet (n-1) to the current output packet n. For each frame, the encoder decides whether to use FEC based on (1) an externally-provided estimate of the channel's packet loss rate; (2) an externally-provided estimate of the channel's capacity; (3) the sensitivity of the audio or speech signal to packet loss; (4) whether the receiving decoder has indicated it can take advantage of "in-band" FEC information. The decision to send "in-band" FEC information is entirely controlled by the encoder and therefore no special precautions for the payload have to be taken.



On the receiving side, the decoder can take advantage of this additional information when, in case of a packet loss, the next packet is available. In order to use the FEC data, the jitter buffer needs to provide access to payloads with the FEC data. The decoder API function has a flag to indicate that a FEC frame rather than a regular frame should be decoded. If no FEC data is available for the current frame, the decoder will consider the frame lost and invokes the frame loss concealment.

If the FEC scheme is not implemented on the receiving side, FEC SHOULD NOT be used, as it leads to an inefficient usage of network resources. Decoder support for FEC SHOULD be indicated at the time a session is set up.

### 3.4. Stereo Operation

Opus allows for transmission of stereo audio signals. This operation is signaled in-band in the Opus payload and no special arrangement is required in the payload format. Any implementation of the Opus decoder MUST be capable of receiving stereo signals, although it MAY decode those signals as mono.

If a decoder can not take advantage of the benefits of a stereo signal this SHOULD be indicated at the time a session is set up. In that case the sending side SHOULD NOT send stereo signals as it leads to an inefficient usage of the network.

## 4. Opus RTP Payload Format

The payload format for Opus consists of the RTP header and Opus payload data.

### 4.1. RTP Header Usage

The format of the RTP header is specified in [RFC3550]. The Opus payload format uses the fields of the RTP header consistent with this specification.

The payload length of Opus is a multiple number of octets and therefore no padding is required. The payload MAY be padded by an integer number of octets according to [RFC3550].

The marker bit (M) of the RTP header is used in accordance with Section 4.1 of [RFC3551].

The RTP payload type for Opus has not been assigned statically and is expected to be assigned dynamically.

The receiving side **MUST** be prepared to receive duplicates of RTP packets. Only one of those payloads **MUST** be provided to the Opus decoder for decoding and others **MUST** be discarded.

Opus supports 5 different audio bandwidths which may be adjusted during the duration of a call. The RTP timestamp clock frequency is defined as the highest supported sampling frequency of Opus, i.e. 48000 Hz, for all modes and sampling rates of Opus. The unit for the timestamp is samples per single (mono) channel. The RTP timestamp corresponds to the sample time of the first encoded sample in the encoded frame. For sampling rates lower than 48000 Hz the number of samples has to be multiplied with a multiplier according to Table 2 to determine the RTP timestamp.

fs (Hz)	Multiplier
8000	6
12000	4
16000	3
24000	2
48000	1

Table 2: Timestamp multiplier

#### 4.2. Payload Structure

The Opus encoder can be set to output encoded frames representing 2.5, 5, 10, 20, 40, or 60 ms of speech or audio data. Further, an arbitrary number of frames can be combined into a packet. The maximum packet length is limited to the amount of encoded data representing 120 ms of speech or audio data. The packetization of encoded data is purely done by the Opus encoder and therefore only one packet output from the Opus encoder **MUST** be used as a payload.

Figure 1 shows the structure combined with the RTP header.

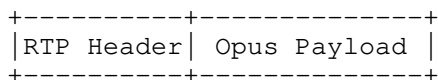


Figure 1: Payload Structure with RTP header

Table 3 shows supported frame sizes in milliseconds of encoded speech or audio data for speech and audio mode (Mode) and sampling rates (fs) of Opus and how the timestamp needs to be incremented for packetization (ts incr). If the Opus encoder outputs multiple encoded frames into a single packet the timestamps have to be added up according to the combined frames.

Mode	fs	2.5	5	10	20	40	60
ts incr	all	120	240	480	960	1920	2880
voice	nb/mb/wb/swb/fb			x	x	x	x
audio	nb/wb/swb/fb	x	x	x	x		

Table 3: Supported Opus frame sizes and timestamp increments

## 5. Congestion Control

The adaptive nature of the Opus codec allows for an efficient congestion control.

The target bitrate of Opus can be adjusted at any point in time and thus allowing for an efficient congestion control. Furthermore, the amount of encoded speech or audio data encoded in a single packet can be used for congestion control since the transmission rate is inversely proportional to these frame sizes. A lower packet transmission rate reduces the amount of header overhead but at the same time increases latency and error sensitivity and should be done with care.

It is RECOMMENDED that congestion control is applied during the transmission of Opus encoded data.

## 6. IANA Considerations

One media subtype (audio/opus) has been defined and registered as described in the following section.

### 6.1. Opus Media Type Registration

Media type registration is done according to [RFC4288] and [RFC4855].

Type name: audio

Subtype name: opus

Required parameters:

rate: RTP timestamp clock rate is incremented with 48000 Hz clock rate for all modes of Opus and all sampling frequencies. For audio sampling rates other than 48000 Hz the rate has to be adjusted to 48000 Hz according to Table 2.

Optional parameters:

maxplaybackrate: a hint about the maximum output sampling rate that the receiver is capable of rendering in Hz. The decoder MUST be capable of decoding any audio bandwidth but due to hardware limitations only signals up to the specified sampling rate can be played back. Sending signals with higher audio bandwidth results in higher than necessary network usage and encoding complexity, so an encoder SHOULD NOT encode frequencies above the audio bandwidth specified by maxplaybackrate. This parameter can take any value between 8000 and 48000, although commonly the value will match one of the Opus bandwidths (Table 1). By default, the receiver is assumed to have no limitations, i.e. 48000.

**sprop-maxcapture**rate: a hint about the maximum input sampling rate that the sender is likely to produce. This is not a guarantee that the sender will never send any higher bandwidth (e.g. it could send a pre-recorded prompt that uses a higher bandwidth), but it indicates to the receiver that frequencies above this maximum can safely be discarded. This parameter is useful to avoid wasting receiver resources by operating the audio processing pipeline (e.g. echo cancellation) at a higher rate than necessary. This parameter can take any value between 8000 and 48000, although commonly the value will match one of the Opus bandwidths (Table 1). By default, the sender is assumed to have no limitations, i.e. 48000.

**maxptime**: the decoder's maximum length of time in milliseconds rounded up to the next full integer value represented by the media in a packet that can be encapsulated in a received packet according to Section 6 of [RFC4566]. Possible values are 3, 5, 10, 20, 40, and 60 or an arbitrary multiple of Opus frame sizes rounded up to the next full integer value up to a maximum value of 120 as defined in Section 4. If no value is specified, 120 is assumed as default. This value is a recommendation by the decoding side to ensure the best performance for the decoder. The decoder **MUST** be capable of accepting any allowed packet sizes to ensure maximum compatibility.

**ptime**: the decoder's recommended length of time in milliseconds rounded up to the next full integer value represented by the media in a packet according to Section 6 of [RFC4566]. Possible values are 3, 5, 10, 20, 40, or 60 or an arbitrary multiple of Opus frame sizes rounded up to the next full integer value up to a maximum value of 120 as defined in Section 4. If no value is specified, 20 is assumed as default. If ptime is greater than maxptime, ptime **MUST** be ignored. This parameter **MAY** be changed during a session. This value is a recommendation by the decoding side to ensure the best performance for the decoder. The decoder **MUST** be capable of accepting any allowed packet sizes to ensure maximum compatibility.

**minptime:** the decoder's minimum length of time in milliseconds rounded up to the next full integer value represented by the media in a packet that SHOULD be encapsulated in a received packet according to Section 6 of [RFC4566]. Possible values are 3, 5, 10, 20, 40, and 60 or an arbitrary multiple of Opus frame sizes rounded up to the next full integer value up to a maximum value of 120 as defined in Section 4. If no value is specified, 3 is assumed as default. This value is a recommendation by the decoding side to ensure the best performance for the decoder. The decoder MUST be capable to accept any allowed packet sizes to ensure maximum compatibility.

**maxaveragebitrate:** specifies the maximum average receive bitrate of a session in bits per second (b/s). The actual value of the bitrate may vary as it is dependent on the characteristics of the media in a packet. Note that the maximum average bitrate MAY be modified dynamically during a session. Any positive integer is allowed but values outside the range between 6000 and 510000 SHOULD be ignored. If no value is specified, the maximum value specified in Section 3.1.1 for the corresponding mode of Opus and corresponding maxplaybackrate: will be the default.

**stereo:** specifies whether the decoder prefers receiving stereo or mono signals. Possible values are 1 and 0 where 1 specifies that stereo signals are preferred and 0 specifies that only mono signals are preferred. Independent of the stereo parameter every receiver MUST be able to receive and decode stereo signals but sending stereo signals to a receiver that signaled a preference for mono signals may result in higher than necessary network utilisation and encoding complexity. If no value is specified, mono is assumed (stereo=0).

**sprop-stereo:** specifies whether the sender is likely to produce stereo audio. Possible values are 1 and 0 where 1 specifies that stereo signals are likely to be sent, and 0 specifies that the sender will likely only send mono. This is not a guarantee that the sender will never send stereo audio (e.g. it could send a pre-recorded prompt that uses stereo), but it indicates to the receiver that the received signal can be safely downmixed to mono. This parameter is useful to avoid wasting receiver resources by operating the audio processing pipeline (e.g. echo cancellation) in stereo when not necessary. If no value is specified, mono is assumed (sprop-stereo=0).

**cbr:** specifies if the decoder prefers the use of a constant bitrate versus variable bitrate. Possible values are 1 and 0 where 1 specifies constant bitrate and 0 specifies variable bitrate. If no value is specified, cbr is assumed to be 0. Note that the

maximum average bitrate may still be changed, e.g. to adapt to changing network conditions.

**useinbandfec:** specifies that the decoder has the capability to take advantage of the Opus in-band FEC. Possible values are 1 and 0. It is RECOMMENDED to provide 0 in case FEC cannot be utilized on the receiving side. If no value is specified, **useinbandfec** is assumed to be 0. This parameter is only a preference and the receiver MUST be able to process packets that include FEC information, even if it means the FEC part is discarded.

**usedtx:** specifies if the decoder prefers the use of DTX. Possible values are 1 and 0. If no value is specified, **usedtx** is assumed to be 0.

Encoding considerations:

Opus media type is framed and consists of binary data according to Section 4.8 in [RFC4288].

Security considerations:

See Section 7 of this document.

Interoperability considerations: none

Published specification: none

Applications that use this media type:

Any application that requires the transport of speech or audio data may use this media type. Some examples are, but not limited to, audio and video conferencing, Voice over IP, media streaming.

Person & email address to contact for further information:

SILK Support [silksupport@skype.net](mailto:silksupport@skype.net)  
Jean-Marc Valin [jmvalin@jmvalin.ca](mailto:jmvalin@jmvalin.ca)

Intended usage: COMMON

Restrictions on usage:

For transfer over RTP, the RTP payload format (Section 4 of this document) SHALL be used.

Author:

Julian Spittka jspittka@gmail.com

Koen Vos koenvos74@gmail.com

Jean-Marc Valin jmvalin@jmvalin.ca

Change controller: TBD

## 6.2. Mapping to SDP Parameters

The information described in the media type specification has a specific mapping to fields in the Session Description Protocol (SDP) [RFC4566], which is commonly used to describe RTP sessions. When SDP is used to specify sessions employing the Opus codec, the mapping is as follows:

- o The media type ("audio") goes in SDP "m=" as the media name.
- o The media subtype ("opus") goes in SDP "a=rtpmap" as the encoding name. The RTP clock rate in "a=rtpmap" MUST be 48000 and the number of channels MUST be 2.
- o The OPTIONAL media type parameters "ptime" and "maxptime" are mapped to "a=ptime" and "a=maxptime" attributes, respectively, in the SDP.
- o The OPTIONAL media type parameters "maxaveragebitrate", "maxplaybackrate", "minptime", "stereo", "cbr", "useinbandfec", and "usedtx", when present, MUST be included in the "a=fmtp" attribute in the SDP, expressed as a media type string in the form of a semicolon-separated list of parameter=value pairs (e.g., maxaveragebitrate=20000). They MUST NOT be specified in an SSRC-specific "fmtp" source-level attribute (as defined in Section 6.3 of [RFC5576]).



- o The OPTIONAL media type parameters "sprop-maxcapture", and "sprop-stereo" MAY be mapped to the "a=fmtp" SDP attribute by copying them directly from the media type parameter string as part of the semicolon-separated list of parameter=value pairs (e.g., sprop-stereo=1). These same OPTIONAL media type parameters MAY also be specified using an SSRC-specific "fmtp" source-level attribute as described in Section 6.3 of [RFC5576]. They MAY be specified in both places, in which case the parameter in the source-level attribute overrides the one found on the "a=fmtp" line. The value of any parameter which is not specified in a source-level source attribute MUST be taken from the "a=fmtp" line, if it is present there.

Below are some examples of SDP session descriptions for Opus:

Example 1: Standard mono session with 48000 Hz clock rate

```
m=audio 54312 RTP/AVP 101
a=rtpmap:101 opus/48000/2
```

Example 2: 16000 Hz clock rate, maximum packet size of 40 ms, recommended packet size of 40 ms, maximum average bitrate of 20000 bps, prefers to receive stereo but only plans to send mono, FEC is allowed, DTX is not allowed

```
m=audio 54312 RTP/AVP 101
a=rtpmap:101 opus/48000/2
a=fmtp:101 maxplaybackrate=16000; sprop-maxcapture=16000;
maxaveragebitrate=20000; stereo=1; useinbandfec=1; usedtx=0
a=ptime:40
a=maxptime:40
```

Example 3: Two-way full-band stereo preferred

```
m=audio 54312 RTP/AVP 101
a=rtpmap:101 opus/48000/2
a=fmtp:101 stereo=1; sprop-stereo=1
```

#### 6.2.1. Offer-Answer Model Considerations for Opus

When using the offer-answer procedure described in [RFC3264] to negotiate the use of Opus, the following considerations apply:

- o Opus supports several clock rates. For signaling purposes only the highest, i.e. 48000, is used. The actual clock rate of the corresponding media is signaled inside the payload and is not subject to this payload format description. The decoder MUST be capable to decode every received clock rate. An example is shown below:

```
m=audio 54312 RTP/AVP 100
a=rtpmap:100 opus/48000/2
```

- o The "ptime" and "maxptime" parameters are unidirectional receive-only parameters and typically will not compromise interoperability; however, dependent on the set values of the parameters the performance of the application may suffer. [RFC3264] defines the SDP offer-answer handling of the "ptime" parameter. The "maxptime" parameter MUST be handled in the same way.
- o The "minptime" parameter is a unidirectional receive-only parameters and typically will not compromise interoperability; however, dependent on the set values of the parameter the performance of the application may suffer and should be set with care.
- o The "maxplaybackrate" parameter is a unidirectional receive-only parameter that reflects limitations of the local receiver. The sender of the other side SHOULD NOT send with an audio bandwidth higher than "maxplaybackrate" as this would lead to inefficient use of network resources. The "maxplaybackrate" parameter does not affect interoperability. Also, this parameter SHOULD NOT be used to adjust the audio bandwidth as a function of the bitrates, as this is the responsibility of the Opus encoder implementation.
- o The "maxaveragebitrate" parameter is a unidirectional receive-only parameter that reflects limitations of the local receiver. The sender of the other side MUST NOT send with an average bitrate higher than "maxaveragebitrate" as it might overload the network and/or receiver. The "maxaveragebitrate" parameter typically will not compromise interoperability; however, dependent on the set value of the parameter the performance of the application may suffer and should be set with care.
- o The "sprop-maxcapturerate" and "sprop-stereo" parameters are unidirectional sender-only parameters that reflect limitations of the sender side. They allow the receiver to set up a reduced-complexity audio processing pipeline if the sender is not planning to use the full range of Opus's capabilities. Neither "sprop-maxcapturerate" nor "sprop-stereo" affect interoperability and the receiver MUST be capable of receiving any signal.
- o The "stereo" parameter is a unidirectional receive-only parameter.
- o The "cbr" parameter is a unidirectional receive-only parameter.

- o The "useinbandfec" parameter is a unidirectional receive-only parameter.
- o The "usedtx" parameter is a unidirectional receive-only parameter.
- o Any unknown parameter in an offer MUST be ignored by the receiver and MUST be removed from the answer.

#### 6.2.2. Declarative SDP Considerations for Opus

For declarative use of SDP such as in Session Announcement Protocol (SAP), [RFC2974], and RTSP, [RFC2326], for Opus, the following needs to be considered:

- o The values for "maxptime", "ptime", "minptime", "maxplaybackrate", and "maxaveragebitrate" should be selected carefully to ensure that a reasonable performance can be achieved for the participants of a session.
- o The values for "maxptime", "ptime", and "minptime" of the payload format configuration are recommendations by the decoding side to ensure the best performance for the decoder. The decoder MUST be capable to accept any allowed packet sizes to ensure maximum compatibility.
- o All other parameters of the payload format configuration are declarative and a participant MUST use the configurations that are provided for the session. More than one configuration may be provided if necessary by declaring multiple RTP payload types; however, the number of types should be kept small.

### 7. Security Considerations

All RTP packets using the payload format defined in this specification are subject to the general security considerations discussed in the RTP specification [RFC3550] and any profile from e.g. [RFC3711] or [RFC3551].

This payload format transports Opus encoded speech or audio data, hence, security issues include confidentiality, integrity protection, and authentication of the speech or audio itself. The Opus payload format does not have any built-in security mechanisms. Any suitable external mechanisms, such as SRTP [RFC3711], MAY be used.

This payload format and the Opus encoding do not exhibit any significant non-uniformity in the receiver-end computational load and thus are unlikely to pose a denial-of-service threat due to the receipt of pathological datagrams.

### 8. Acknowledgements

TBD

## 9. Normative References

- [RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, March 1997.
- [RFC2326] Schulzrinne, H., Rao, A., and R. Lanphier, "Real Time Streaming Protocol (RTSP)", RFC 2326, April 1998.
- [RFC2974] Handley, M., Perkins, C., and E. Whelan, "Session Announcement Protocol", RFC 2974, October 2000.
- [RFC3264] Rosenberg, J. and H. Schulzrinne, "An Offer/Answer Model with Session Description Protocol (SDP)", RFC 3264, June 2002.
- [RFC3550] Schulzrinne, H., Casner, S., Frederick, R., and V. Jacobson, "RTP: A Transport Protocol for Real-Time Applications", STD 64, RFC 3550, July 2003.
- [RFC3551] Schulzrinne, H. and S. Casner, "RTP Profile for Audio and Video Conferences with Minimal Control", STD 65, RFC 3551, 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.
- [RFC4288] Freed, N. and J. Klensin, "Media Type Specifications and Registration Procedures", RFC 4288, December 2005.
- [RFC4566] Handley, M., Jacobson, V., and C. Perkins, "SDP: Session Description Protocol", RFC 4566, July 2006.
- [RFC4855] Casner, S., "Media Type Registration of RTP Payload Formats", RFC 4855, February 2007.
- [RFC5576] Lennox, J., Ott, J., and T. Schierl, "Source-Specific Media Attributes in the Session Description Protocol (SDP)", RFC 5576, June 2009.
- [RFC6562] Perkins, C. and JM. Valin, "Guidelines for the Use of Variable Bit Rate Audio with Secure RTP", RFC 6562, March 2012.
- [RFC6716] Valin, JM., Vos, K., and T. Terriberry, "Definition of the Opus Audio Codec", RFC 6716, September 2012.

Authors' Addresses

Julian Spittka

Email: [jspittka@gmail.com](mailto:jspittka@gmail.com)

Koen Vos

Skype Technologies S.A.

3210 Porter Drive

Palo Alto, CA 94304

USA

Email: [koenvos74@gmail.com](mailto:koenvos74@gmail.com)

Jean-Marc Valin

Mozilla

650 Castro Street

Mountain View, CA 94041

USA

Email: [jmvalin@jmvalin.ca](mailto:jmvalin@jmvalin.ca)