Network Working Group
Internet-Draft
Intended status: Informational
Expires: May 18, 2017

C. Perkins
University of Glasgow

November 14, 2016

RTP Control Protocol (RTCP) Feedback for Congestion Control in Interactive Multimedia Conferences

draft-ietf-rmcat-rtp-cc-feedback-03

Abstract

This memo discusses the types of congestion control feedback that it is possible to send using the RTP Control Protocol (RTCP), and their suitability of use in implementing congestion control for unicast multimedia applications.

Status of This Memo

Copyright Notice

Table of Contents

1. Introduction

The coming deployment of WebRTC systems raises the prospect that high quality video conferencing will see extremely wide use. To ensure the stability of the network in the face of this use, WebRTC systems will need to use some form of congestion control for their RTP-based media traffic. To develop such congestion control, it is necessary to understand the sort of congestion feedback that can be provided within the framework of RTP [RFC3550] and the RTP Control Protocol (RTCP). It then becomes possible to determine if this is sufficient for congestion control, or if some form of RTP extension is needed.

This memo considers the congestion feedback that can be sent using RTCP under the RTP/SAVPF profile [RFC5124] (the secure version of the RTP/AVPF profile [RFC4585]). This profile was chosen as it forms the basis for media transport in WebRTC [I-D.ietf-rtcweb-rtp-usage] systems. Nothing in this memo is specific to the secure version of the profile, or to WebRTC, however.

2. Possible Models for RTCP Feedback

Several questions need to be answered when providing RTCP reception quality feedback for congestion control purposes. These include:

- How often is feedback needed?
- How much overhead is acceptable?
- How much, and what, data does each report contain?

The key question is how often does the receiver need to send feedback on the reception quality it is experiencing, and hence the congestion state of the network? Traditional congestion control protocols, such as TCP, send acknowledgements with every packet (or, at least, every couple of packets). That is straight-forward and low overhead when traffic is bidirectional and acknowledgements can be piggybacked onto return path data packets. It can also be acceptable, and can have reasonable overhead, to send separate acknowledgement packets when those packets are much smaller than data packets. It becomes a problem, however, when there is no return traffic on which to piggyback acknowledgements, and when acknowledgements are similar in size to data packets; this can be the case for some forms of media traffic, especially for voice over IP (VoIP) flows, but less so for video.

When considering multimedia traffic, it might make sense to consider less frequent feedback. For example, it might be possible to send a feedback packet once per video frame, or every few frames, or once per network round trip time (RTT). This could still give sufficiently frequent feedback for the congestion control loop to be stable and responsive while keeping the overhead reasonable when the feedback cannot be piggybacked onto returning data. In this case, it is important to note that RTCP can send much more detailed feedback than simple acknowledgements. For example, if it were useful, it could be possible to use an RTCP extended report (XR) packet [RFC3611] to send feedback once per RTT comprising a bitmap of lost and received packets, with reception times, over that RTT. As long as feedback is sent frequently enough that the control loop is stable, and the sender is kept informed when data leaves the network (to provide an equivalent to ACK clocking in TCP), it is not necessary to report on every packet at the instant it is received (indeed, it is unlikely that a video codec can react instantly to a rate change anyway, and there is little point in providing feedback more often than the codec can adapt).

The amount of overhead due to congestion control feedback that is considered acceptable has to be determined. RTCP data is sent in separate packets to RTP data, and this has some cost in terms of additional header overhead compared to protocols that piggyback feedback on return path data packets. The RTP standards have long said that a 5% overhead for RTCP traffic generally acceptable, while providing the ability to change this fraction. Is this still the case for congestion control feedback? Or is there a desire to either see more responsive feedback and congestion control, possibility with a higher overhead, or is lower overhead wanted, accepting that this might reduce responsiveness of the congestion control algorithm?

Finally, the details of how much, and what, data is to be sent in each report will affect the frequency and/or overhead of feedback. There is a fundamental trade-off that the more frequently feedback packets are sent, the less data can be included in each packet to keep the overhead constant. Does the congestion control need high rate but simple feedback (e.g., like TCP acknowledgements), or is it acceptable to send more complex feedback less often?

3. What Feedback is Achievable With RTCP?

3.1. Scenario 1: Voice Telephony

In many ways, point-to-point voice telephony is the simplest scenario for congestion control, since there is only a single media stream to control. It's complicated, however, by severe bandwidth constraints on the feedback, to keep the overhead manageable.

Assume a two-party point-to-point voice-over-IP call, using RTP over UDP/IP. A rate adaptive speech codec, such as Opus, is used, encoded into RTP packets in frames of duration Tf seconds (Tf = 20ms in many cases, but values up to 60ms are not uncommon). The congestion control algorithm requires feedback every Nr frames, i.e., every Nr * Tf seconds, to ensure effective control. Both parties in the call send speech data or comfort noise with sufficient frequency that they are counted as senders for the purpose of the RTCP reporting interval calculation.

RTCP feedback packets can be full, compound, RTCP feedback packets, or non-compound RTCP packets. A compound RTCP packet is sent once for every Nnc non-compound RTCP packets.

Compound RTCP packets contain a Sender Report (SR) packet and a Source Description (SDES) packet, and an RTP Congestion Control Feedback (RC2F) packet [I-D.dt-rmcat-feedback-message]. Non-compound RTCP packets contain only the RC2F packet. Since each participant sends only a single media stream, the extensions for RTCP report aggregation [I-D.ietf-avtcore-rtp-multi-stream] and reporting group optimisation [I-D.ietf-avtcore-rtp-multi-stream-optimisation] are not used.

Within each compound RTCP packet, the SR packet will contain a sender information block (28 octets) and a single reception report block (24 octets), for a total of 52 octets. A minimal SDES packet will contain a header (4 octets) and a single chunk containing an SSRC (4 octets) and a CNAME item, and if the recommendations for choosing the CNAME [RFC7022] are followed, the CNAME item will comprise a 2 octet header, 16 octets of data, and 2 octets of padding, for a total SDES packet size of 28 octets. The RC2F packets contains an XR block header and SSRC (8 octets), a block type and timestamp (8 octets), the SSRC, beginning and ending sequence numbers (8 octets), and 2*Nr octets of reports, for a total of 24 + 2*Nr octets. If IPv4 is used, with no IP options, the UDP/IP header will be 28 octets in size. This gives a total compound RTCP packet size of Sc = 132 + 2*Nr octets.

The non-compound RTCP packets will comprise just the RC2F packet with a UDP/IP header. It can be seen that these packets will be Snc = 48 + 2*Nr octets in size.

The RTCP reporting interval calculation ([RFC3550], Section 6.2) for a two-party session where both participants are senders, reduces to Trtcp = n * Srtcp/Brtcp where Srtcp = (Sc + Nnc * Snc)/(1 + Nnc) is the average RTCP packet size in octets, Brtcp is the bandwidth allocated to RTCP in octets per second, and n is the number of participants (n=2 in this scenario).

To ensure a report is sent every Nr frames, it is necessary to set the RTCP reporting interval Trtcp = Nr * Tf, which when substituted into the previous gives Nr * Tf = n * Srtcp/Brtcp.

Solving for the RTCP bandwidth, Brtcp, and expanding the definition of Srtcp gives Brtcp = (n * (Sc + Nnc * Snc))/(Nr * Tf * (1 + Nnc)).

If we assume every report is a compound RTCP packet (i.e., Nnc = 0), the frame duration Tf = 20ms, and an RTCP report is sent for every second frame (i.e., 25 RTCP reports per second), this expression gives the needed RTCP bandwidth Brtcp = 53.1kbps. Increasing the frame duration, or reducing the frequency of reports, reduces the RTCP bandwidth, as shown below:

Tf (seconds)	Nr (frames)	rtcp_bw (kbps)
20ms	2	53.1
20ms	4	27.3
20ms	8	14.5
20ms	16	8.01
60ms	2	17.7
60ms	4	9.1
60ms	8	4.8
60ms	16	2.66

Table 1: Required RTCP bandwidth for VoIP feedback

The final row of the table (60ms frames, report every 16 frames) sends RTCP reports once per second, giving an RTCP bandwidth of 2.66kbps.

The overhead can be reduced by sending some reports in non-compound RTCP packets [RFC5506]. For example, if we alternate compound and non-compound RTCP packets, i.e., Nnc = 1, the calculation gives:

Tf (seconds)	Nr (frames)	rtcp_bw (kbps)
20ms	2	36.7
20ms	4	19.1
20ms	8	10.4
20ms	16	6.0
60ms	2	12.2
60ms	4	6.4
60ms	8	3.5
60ms	16	2.0

Table 2: Required RTCP bandwidth for VoIP feedback (alternating compound and non-compound reports)

The RTCP bandwidth needed for 60ms frames, reporting every 16 frames (once per second), can be seen to drop to 2.01kbps. This calculation can be repeated for other patterns of compound and non-compound RTCP packets, feedback frequency, and frame duration, as needed.

Note: To achieve the RTCP transmission intervals above the RTP/SAVPF profile with T_rr_interval=0 is used, since even when using the reduced minimal transmission interval, the RTP/SAVP profile would only allow sending RTCP at most every 0.11s (every third frame of video). Using RTP/SAVPF with T_rr_interval=0 however is capable of fully utilizing the configured 5% RTCP bandwidth fraction.

3.2. Scenario 2: Point-to-Point Video Conference

Consider a point to point video call between two end systems. There will be four RTP flows in this scenario, two audio and two video, with all four flows being active for essentially all the time (the audio flows will likely use voice activity detection and comfort noise to reduce the packet rate during silent periods, and does not cause the transmissions to stop).

Assume all four flows are sent in a single RTP session, each using a separate SSRC; the RTCP reports from co-located audio and video SSRCs at each end point are aggregated [I-D.ietf-avtcore-rtp-multi-stream]; the optimisations in [I-D.ietf-avtcore-rtp-multi-stream-optimisation] are used; and congestion control feedback is sent [I-D.dt-rmcat-feedback-message].

When all members are senders, the RTCP timing rules in Section 6.2 and 6.3 of [RFC3550] and [RFC4585] reduce to:

rtcp_interval = avg_rtcp_size * n / rtcp_bw

where n is the number of members in the session, the avg_rtcp_size is measured in octets, and the rtcp_bw is the bandwidth available for RTCP, measured in octets per second (this will typically be 5% of the session bandwidth).

The average RTCP size will depend on the amount of feedback that is sent in each RTCP packet, on the number of members in the session, on the size of source description (RTCP SDES) information sent, and on the amount of congestion control feedback sent in each packet.

As a baseline, each RTCP packet will be a compound RTCP packet that contains an aggregate of a compound RTCP packet generated by the video SSRC and a compound RTCP packet generated by the audio SSRC. Since the RTCP reporting group extensions are used, one of these SSRCs will be a reporting SSRC, and the other will delegate its reports to that.

The aggregated compound RTCP packet from the non-reporting SSRC will contain an RTCP SR packet, an RTCP SDES packet, and an RTCP RGRS packet. The RTCP SR packet contains the 28 octet header and sender information, but no report blocks (since the reporting is delegated). The RTCP SDES packet will comprise a header (4 octets), originating SSRC (4 octets), a CNAME chunk, a terminating chunk, and any padding. If the CNAME follows [RFC7022] and [I-D.ietf-rtcweb-rtp-usage] it will be 18 octets in size, and will need 1 octet of padding, making the SDES packet 28 octets in size. The RTCP RGRS packet will be 12 octets in size. This gives a total of 28 + 28 + 12 = 68 octets.

The aggregated compound RTCP packet from the reporting SSRC will contain an RTCP SR packet, an RTCP SDES packet, and an RTCP XR congestion control feedback packet. The RTCP SR packet will contain two report blocks, one for each of the remote SSRCs (the report for the other local SSRC is suppressed by the reporting group extension), for a total of 28 + (2 * 24) = 76 octets. The RTCP SDES packet will comprise a header (4 octets), originating SSRC (4 octets), a CNAME chunk, an RGRP chunk, a terminating chunk, and any padding. If the CNAME follows [RFC7022] and [I-D.ietf-rtcweb-rtp-usage] it will be 18 octets in size. The RGRP chunk similarly comprises 18 octets, and 3 octets of padding are needed, for a total of 48 octets. The RTCP XR congestion control feedback report comprises an 8 octet XR header, an 8 octet RC2F header, then for each of the remote audio and video SSRCs, an 8 octet report header, and 2 octets per packet reported upon, and padding to a 4 octet boundary, if needed; that is 8 + 8 + 8

+ (2 * Nv) + 8 + (2 * Na) where Nv is the number of video packets per report, and Na is the number of audio packets per report.

The complete compound RTCP packet contains the RTCP packets from both the reporting and non-reporting SSRCs, an SRTP authentication tag, and a UDP/IPv4 header. The size of this RTCP packet is therefore: 252 + (2 * Nv) + (2 * Na) octets. Since the aggregate RTCP packet contains reports from two SSRCs, the RTCP packet size is halved before use [I-D.ietf-avtcore-rtp-multi-stream]. Accordingly, we define Sc = (252 + (2 * Nv) + (2 * Na))/2 for this scenario.

How many packets does the RTCP XR congestion control feedback packet report on? This is obviously highly dependent on the choice of codec and encoding parameters, and might be quite bursty if the codec sends I-frames from which later frames are predicted. For now though, assume constant rate media with an MTU around 1500 octets, with reports for both audio and video being aggregated and sent to align with video frames. This gives the following, assuming Nr =1 and Nnc = 0 (i.e., send a compound RTCP packet for each video frame, and no non-compound packets), and using the calculation from Scenario 1: Brtcp = (n * (Sc + Nnc * Snc))/(Nr * Tf * (1 + Nnc))

Data Rate (kbps)	Video Frame Rate	Video Packets per Report: Nv	Audio Packets per Report: Na	Required RTCP bandwidth: Brtcp (kbps)
100	8	1	6	33.3 (33%)
200	16	1	3	65.0 (33%)
350	30	1	2	120.1 (35%)
700	30	2	2	121.9 (17%)
700	60	1	1	240.0 (34%)
1024	30	3	2	122.8 (12%)
1400	60	2	1	241.8 (17%)
2048	30	6	2	125.6 (6%)
2048	60	3	1	243.8 (12%)
4096	30	12	2	131.3 (3%)
4096	60	6	1	294.4 (6%)

Table 3: Required RTCP bandwidth, reporting on every frame

The RTCP bandwidth needed scales inversely with Nr. That is, it is halved if Nr=2 (report on every second packet), is reduced to one-third if Nr=3 (report on every third packet), and so on.

The needed RTCP bandwidth scales as a percentage of the data rate following the ratio of the frame rate to the data rate. As can be seen from the table above, the RTCP bandwidth needed is a significant fraction of the media rate, if reporting on every frame for low rate video. This can be solved by reporting less often at lower rates. For example, to report on every frame of 100kbps/8fps video requires the RTCP bandwidth to be 21% of the media rate; reporting every fourth frame (i.e., twice per second) reduces this overhead to 5%.

Use of reduced size RTCP [RFC5506] would allow the SR and SDES packets to be omitted from some reports. These "non-compound" (actually, compound but reduced size in this case) RTCP packets would contain an RTCP RGRS packet from the non-reporting SSRC, and an RTCP SDES RGRP packet and a congestion control feedback packet from the reporting SSRC. This will be 12 + 28 + 12 + 8 + 2*Nv + 8 + 2*Na octets, plus UDP/IP header. That is, Snc = (96 + 2*Nv + 2*Na)/2. Repeating the analysis above, but alternating compound and non-compound reports, i.e., setting Nnc = 1, gives:

Data Rate (kbps)	Video Frame Rate	Video Packets per Report: Nv	Audio Packets per Report: Na	Required RTCP bandwidth: Brtcp (kbps)
100	8	1	6	23.5 (23%)
200	16	1	3	45.5 (23%)
350	30	1	2	84.4 (24%)
700	30	2	2	85.3 (12%)
700	60	1	1	166.9 (24%)
1024	30	3	2	86.2 (8%)
1400	60	2	1	168.8 (12%)
2048	30	6	2	89.1 (4%)
2048	60	3	1	170.6 (8%)
4096	30	12	2	94.7 (2%)
4096	60	6	1	176.3 (4%)

Table 4: Required RTCP bandwidth, reporting on every frame, with reduced-size reports

The use of reduced-size RTCP gives a noticeable reduction in the needed RTCP bandwidth, and can be combined with reporting every few frames rather than every frames. Overall, it is clear that the RTCP overhead can be reasonable across the range of data and frame rates, if RTCP is configured carefully.

3.3. Scenario 3: Group Video Conference

(tbd)

3.4. Scenario 4: Screen Sharing

(tbd)

4. Discussion and Conclusions

RTCP as it is currently specified cannot be used to send per-packet congestion feedback. RTCP can, however, be used to send congestion feedback on each frame of video sent, provided the session bandwidth exceeds a couple of megabits per second (the exact rate depending on the number of session participants, the RTCP bandwidth fraction, and what RTCP extensions are enabled, and how much detail of feedback is needed). For lower rate sessions, the overhead of reporting on every frame becomes high, but can be reduced to something reasonable by sending reports once per N frames (e.g., every second frame), or by sending non-compound RTCP reports in between the regular reports.

If it is desired to use RTCP in something close to it's current form for congestion feedback in WebRTC, the multimedia congestion control algorithm needs be designed to work with feedback sent every few frames, since that fits within the limitations of RTCP. That feedback can be a little more complex than just an acknowledgement, provided care is taken to consider the impact of the extra feedback on the overhead, possibly allowing for a degree of semantic feedback, meaningful to the codec layer as well as the congestion control algorithm.

The format described in [I-D.dt-rmcat-feedback-message] seems sufficient for the needs of congestion control feedback. There is little point optimising this format: the main overhead comes from the UDP/IP headers and the other RTCP packets included in the compound packets, and can be lowered by using the [RFC5506] extensions and sending reports less frequently.

Further study of the scenarios of interest is needed, to ensure that the analysis presented is applicable to other media topologies, and to sessions with different data rates and sizes of membership.

5. Security Considerations

An attacker that can modify or spoof RTCP congestion control feedback packets can manipulate the sender behaviour to cause denial of service. This can be prevented by authentication and integrity protection of RTCP packets, for example using the secure RTP profile [RFC3711][RFC5124], or by other means as discussed in [RFC7201].

6. IANA Considerations

There are no actions for IANA.

7. Acknowledgements

Thanks to Magnus Westerlund and the members of the RMCAT feedback design team for their feedback.

8. Informative References

[I-D.dt-rmcat-feedback-message]
Sarker, Z., Perkins, C., Singh, V., and M. Ramalho, "RTP Control Protocol (RTCP) Feedback for Congestion Control", draft-dt-rmcat-feedback-message-01 (work in progress), October 2016.

[I-D.ietf-avtcore-rtp-multi-stream]
Lennox, J., Westerlund, M., Wu, Q., and C. Perkins, "Sending Multiple RTP Streams in a Single RTP Session", draft-ietf-avtcore-rtp-multi-stream-11 (work in progress), December 2015.

[I-D.ietf-avtcore-rtp-multi-stream-optimisation]
Lennox, J., Westerlund, M., Wu, Q., and C. Perkins, "Sending Multiple RTP Streams in a Single RTP Session: Grouping RTCP Reception Statistics and Other Feedback", draft-ietf-avtcore-rtp-multi-stream-optimisation-12 (work in progress), March 2016.

[I-D.ietf-rtcweb-rtp-usage]
Perkins, C., Westerlund, M., and J. Ott, "Web Real-Time Communication (WebRTC): Media Transport and Use of RTP", draft-ietf-rtcweb-rtp-usage-26 (work in progress), March 2016.

[RFC3550] Schulzrinne, H., Casner, S., Frederick, R., and V. Jacobson, "RTP: A Transport Protocol for Real-Time Applications", STD 64, RFC 3550, DOI 10.17487/RFC3550, July 2003, <http://www.rfc-editor.org/info/rfc3550>.

[RFC3611] Friedman, T., Ed., Caceres, R., Ed., and A. Clark, Ed., "RTP Control Protocol Extended Reports (RTCP XR)", RFC 3611, DOI 10.17487/RFC3611, November 2003, <http://www.rfc-editor.org/info/rfc3611>.

[RFC3711] Baugher, M., McGrew, D., Naslund, M., Carrara, E., and K. Norrman, "The Secure Real-time Transport Protocol (SRTP)", RFC 3711, DOI 10.17487/RFC3711, March 2004, <http://www.rfc-editor.org/info/rfc3711>.

[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, DOI 10.17487/RFC4585, July 2006, <http://www.rfc-editor.org/info/rfc4585>.

[RFC5124] Ott, J. and E. Carrara, "Extended Secure RTP Profile for Real-time Transport Control Protocol (RTCP)-Based Feedback (RTP/SAVPF)", RFC 5124, DOI 10.17487/RFC5124, February 2008, <http://www.rfc-editor.org/info/rfc5124>.

[RFC5506] Johansson, I. and M. Westerlund, "Support for Reduced-Size Real-Time Transport Control Protocol (RTCP): Opportunities and Consequences", RFC 5506, DOI 10.17487/RFC5506, April 2009, <http://www.rfc-editor.org/info/rfc5506>.

[RFC7022] Begen, A., Perkins, C., Wing, D., and E. Rescorla, "Guidelines for Choosing RTP Control Protocol (RTCP) Canonical Names (CNAMEs)", RFC 7022, DOI 10.17487/RFC7022, September 2013, <http://www.rfc-editor.org/info/rfc7022>.

[RFC7201] Westerlund, M. and C. Perkins, "Options for Securing RTP Sessions", RFC 7201, DOI 10.17487/RFC7201, April 2014, <http://www.rfc-editor.org/info/rfc7201>.

Author's Address

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

Email: csp@csperkins.org