Network Working Group
Internet-Draft
Intended status: Informational
Expires: January 9, 2017

C. Perkins
University of Glasgow

July 8, 2016

Using RTP Control Protocol (RTCP) Feedback for Unicast Multimedia Congestion Control

draft-ietf-rmcat-rtp-cc-feedback-01

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 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. Per-packet Feedback

RTCP packets are sent as separate packets to RTP media data, and the protocol includes no mechanism for piggybacking an RTCP packet onto an RTP data packet. In addition, the RTCP timing rules are based on the size of the RTP session, the number of active senders, the RTCP packet size, and the configured RTCP bandwidth fraction, with randomisation to prevent synchronisation of reports; accordingly the RTCP packet transmission times are extremely unlikely to line up with RTP packet transmission times. As a result, RTCP cannot be used to send per-packet feedback in it's current form.

All of these issues with using RTCP for per-packet feedback could be resolved in an update to the RTP protocol, of course. Such an update could change the RTCP timing rules, and might define a shim layer to allow multiplexing of RTP and RTCP into a single packet, or to extend the RTP header to piggyback feedback data. This sort of change would be a large, and almost certainly backwards incompatible, extension to the RTP protocol, and is unlikely to be completed quickly, but could be done if there was a need.

3.2. Per-frame Feedback

Consider one of the simplest scenarios for WebRTC: 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 [RFC3550], [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, then for each of the remote audio and video SSRCs, a 12 octet report header, and 2 octets per packet reported upon, and padding to a 4 octet boundary, if needed; that is 8 + 12 + (2 * video_packets_per_report) + 12 + (2 * audio_packets_per_report). = 32 + (2 * video_packets_per_report) + (2 * audio_packets_per_report). The compound RTCP packet will be 156 + (2 * video_packets_per_report) + (2 * audio_packets_per_report).

The resulting aggregate RTCP packet, containing both compound RTCP packets, will be sent in UDP/IPv4 with no IP options and using Secure RTP, which adds 20 (IPv4) + 8 (UDP) + 14 (SRTP with 80 bit Authentication tag) = 42 octets, the avg_rtcp_size will therefore be 42 + 68 + 156 + (2 * video_packets_per_report) + (2 * audio_packets_per_report). (FIXME: this ignores padding in RTCP) Since the aggregate RTCP packet contains reports from two SSRCs, the avg_rtcp_packet size is halved before use [I-D.ietf-avtcore-rtp-multi-stream]. The value n is this scenario is 4, and the rtcp_bw is assumed to be 5% of the session bandwidth.

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, assume video_packets_per_second = (video_bit_rate_bps / 8) / mtu and video_packets_per_report = video_packets_per_seconds / fps. For audio, assume 50 packets per second, with audio_packets_per_report based on the video frame rate (i.e., RTCP packets for the audio SSRC are aggregated with those from the video SSRC).

If it is desired to send RTCP feedback packets on average 30 times per second, to correspond to one RTCP report every frame for 30fps video, one can solve the above expressions to determine the session bandwidth needed to give an RTCP reporting interval of 1/30 second. This is approximately 2.5Mbps. That is, provided the video session bandwidth is greater than approximately 2.5Mbps, one can report on each packet arrival (with ECN marks and arrival time) for every frame of 30 fps video, using existing RTCP mechanisms. This is not out of line with the expected session bandwidth for this type of application, suggesting the RTCP feedback can be used to provide per-frame congestion control feedback for WebRTC-style applications.

> 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.

3.3. Per-RTT Feedback

The arguments made in Section 3.2 apply to this case as well. The network RTT will usually be larger than the media framing interval, so sending feedback per RTT is less of a load on RTCP than sending feedback per frame.

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). RTCP can likely also be used to send feedback on a per-RTT basis, provided the RTT is not too low.

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 roughly each frame or each RTT, rather than per packet, 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.

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

The security considerations of [RFC3550], [RFC4585], and [RFC5124] apply.

6. IANA Considerations

There are no actions for IANA.

7. Acknowledgements

Thanks to Magnus Westerlund for his feedback on Section 3.2.

8. Informative References

[I-D.dt-rmcat-feedback-message]
Sarker, Z., Perkins, D., Singh, V., and D. Ramalho, "RTP Control Protocol (RTCP) Feedback for Congestion Control", draft-dt-rmcat-feedback-message-00 (work in progress), July 2016.

[I-D.ietf-avtcore-rtp-multi-stream]
Lennox, J., Westerlund, M., Wu, Q., and D. 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 D. 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, D., 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>.

[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>.

[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>.

Author's Address

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

Email: csp@csperkins.org