# RFC 9392
# Sending RTP Control Protocol (RTCP) Feedback for Congestion Control in Interactive Multimedia Conferences

## Abstract

This memo discusses the rate at which congestion control feedback can be sent using the RTP Control Protocol (RTCP) and the suitability of RTCP for implementing congestion control for unicast multimedia applications.

## Status of This Memo

## Copyright Notice

## Table of Contents

## 1.  Introduction

The deployment of WebRTC systems [RFC8825] has resulted in high-quality video conferencing seeing extremely wide use. To ensure the stability of the network in the face of this use, WebRTC systems need to use some form of congestion control for their RTP-based media traffic [RFC2914] [RFC8083] [RFC8085] [RFC8834], allowing them to adapt and adjust the media data they send to match changes in the available network capacity. In addition to ensuring the stable operation of the network, such adaptation is critical to ensuring a good user experience, since it allows the sender to match the media to the network capacity, rather than forcing the receiver to compensate for uncontrolled packet loss when the available capacity is exceeded.

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.

As this memo will show, if it is desired to use RTCP in something close to its current form for congestion feedback, the multimedia congestion control algorithm needs to be designed to work with detailed feedback sent every few frames, rather than per-frame acknowledgement, to match the constraints of RTCP.

This memo considers unicast 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 because it forms the basis for media transport in WebRTC [RFC8834] systems. However, nothing in this memo is specific to the secure version of the profile or to WebRTC. It is also assumed that the congestion control feedback mechanism described in [RFC8888] and common RTCP extensions for efficient feedback [RFC5506] [RFC8108] [RFC8861] [RFC8872] are used.

## 1.1.  Terminology

Nr:    number of frames between feedback reports

Nrs:   number of reduced-size RTCP packets send for every compound RTCP packet

Na:    number of audio packets per report

Nv:    number of video packets per reports

Sc:    size of a compound RTCP packet

Srs:   size of a reduced-size RTCP packet

Tf:    duration of a media frame in seconds

Rf:    frame rate 1/Tf

# 2.  Considerations for RTCP Feedback

Several questions need to be answered when providing RTCP 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?

However, the key question is as follows: how often does the receiver need to send feedback on the reception quality it is experiencing and hence the congestion state of the network?

Widely used transport protocols, such as TCP, send acknowledgements frequently. For example, a TCP receiver will send an acknowledgement at least once every 0.5 seconds or when new data equal to twice the maximum segment size has been received [RFC9293]. That has relatively 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.

Frequent acknowledgements can become a problem, however, when there is no return traffic on which to piggyback feedback or if separate feedback and data packets are sent and the feedback is similar in size to the data being acknowledged. This can be the case for some forms of media traffic, especially for Voice over IP (VoIP) flows, leading to high overhead when using a transport protocol that sends frequent feedback. Approaches like in-network filtering of acknowledgements that have been proposed to reduce acknowledgement overheads on highly asymmetric links (e.g., as mentioned in [RFC3449]) can also reduce the feedback frequency and overhead for multimedia traffic, but this so-called "stretch-ACK" behavior is nonstandard and not guaranteed.

Accordingly, when implementing congestion control for RTP-based multimedia traffic, it might make sense to give the option of sending congestion feedback less often than TCP does. For example, it might be possible to send a feedback packet once per video frame, 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; the feedback could comprise 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 acknowledgement (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, and there is little point in providing feedback more often than the codec can adapt. This suggests that an RTP receiver needs to be configured to provide feedback at a rate that matches the rate of adaptation of the sender. In the best case, this will match the media frame rate but might often be slower.

Reducing the feedback frequency compared to TCP will reduce feedback overhead but will lead multimedia flows to adapt to congestion more slowly than TCP, raising concerns about inter-flow fairness. Similar concerns are noted in [RFC5348], and accordingly, the congestion control algorithm described therein aims for "reasonable" fairness and a sending rate that is "generally within a factor of two" of what TCP would achieve under the same conditions. It is to be noted, however, that TCP exhibits inter-flow unfairness when flows with differing round-trip times compete, and stretch acknowledgements due to in-network traffic manipulation are not uncommon and also raise fairness concerns. Implementations need to balance potential unfairness against feedback overhead.

Generating and processing feedback consumes resources at the sender and receiver. The feedback packets also incur forwarding costs, contribute to link utilization, and can affect the timing of other traffic on the network. This can affect performance on some types of networks that can be impacted by the rate, timing, and size of feedback packets, as well as the overall volume of feedback bytes.

The amount of overhead due to congestion control feedback that is considered acceptable has to be determined. RTCP feedback 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 is generally acceptable. Is this still the case for congestion control feedback? Is there a desire to provide more responsive feedback and congestion control, possibly 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 a high rate but simple feedback (e.g., like TCP acknowledgements), or is it acceptable to send more complex feedback less often? Is it useful for the congestion control to receive frequent feedback, perhaps to provide more accurate round-trip time estimates, or to provide robustness in case feedback packets are lost, even if the media sending rate cannot quickly be changed? Or is low-rate feedback, resulting in slowly responsive changes to the sending rate, acceptable? Different combinations of the congestion control algorithm and media codec might require different trade-offs, and the correct trade-off for interactive, self-paced, real-time multimedia traffic might not be the same as that for TCP congestion control.

# 3.   What Feedback is Achievable with RTCP?

The following sections illustrate how the RTCP congestion control feedback report [RFC8888] can be used in different scenarios and illustrate the overheads of this approach.

## 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 VoIP call, using RTP over UDP/IP. A rate-adaptive speech codec, such as Opus, is used, encoded into RTP packets in frames of a duration of Tf seconds (Tf = 0.020 s in many cases, but values up to 0.060 s 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 reduced-size RTCP packets [RFC5506]. A compound RTCP packet is sent once for every Nrs reduced-size RTCP packets.

Compound RTCP packets contain a Sender Report (SR) packet, a Source Description (SDES) packet, and an RTP Congestion Control Feedback (CCFB) packet [RFC8888]. Reduced-size RTCP packets contain only the CCFB packet. Since each participant sends only a single RTP media stream, the extensions for RTCP report aggregation [RFC8108] and reporting group optimization [RFC8861] 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), a single chunk containing a synchronization source (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 CCFB packets contain an RTCP header and SSRC (8 octets), a report timestamp (4 octets), the other party's SSRC, beginning and ending sequence numbers (8 octets), and 2 * Nr octets of reports, for a total of 20 + (2 * Nr) octets. The compound Secure RTCP (SRTCP) packet will include 4 octets of trailer, followed by an 80-bit (10-octet) authentication tag if HMAC-SHA1 authentication is used. 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 = 142 + (2 * Nr) octets.

The reduced-size RTCP packets will comprise just the CCFB packet, SRTCP trailer and authentication tag, and a UDP/IP header. It can be seen that these packets will be Srs = 62 + (2 * Nr) octets in size.

The RTCP reporting interval calculation (Sections 6.2 and 6.3 of [RFC3550] and [RFC4585]) for a two-party session where both participants are senders reduces to:

```
    Trtcp = n * Srtcp / Brtcp
```

where Srtcp = (Sc + Nrs * Srs) / (1 + Nrs) 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 in the RTP session (in this scenario, n = 2).

To ensure an RTCP report containing congestion control feedback is sent after every Nr frames of audio, it is necessary to set the RTCP reporting interval to Trtcp = Nr * Tf, which when substituted into the previous, gives Nr * Tf = n * Srtcp / Brtcp. Solving this to give the RTCP bandwidth (Brtcp) and expanding the definition of Srtcp gives:

```
    Brtcp = (n * (Sc + Nrs * Srs)) / (Nr * Tf * (1 + Nrs))
```

If we assume every report is a compound RTCP packet (i.e., Nrs = 0), the frame duration is Tf = 20 ms, and an RTCP report is sent for every second frame (i.e., 25 RTCP reports per second), this gives an RTCP feedback bandwidth of Brtcp = 57 kbps. Increasing the frame duration or reducing the frequency of reports will reduce the RTCP bandwidth, as shown in Table 1.

| Tf (seconds) | Nr (frames) | rtcp_bw (kbps) |
|---|---|---|
| 0.020 | 2 | 57.0 |
| 0.020 | 4 | 29.3 |
| 0.020 | 8 | 15.4 |

| Tf (seconds) | Nr (frames) | rtcp_bw (kbps) |
|---|---|---|
| 0.020 | 16 | 8.5 |
| 0.060 | 2 | 19.0 |
| 0.060 | 4 | 9.8 |
| 0.060 | 8 | 5.1 |
| 0.060 | 16 | 2.8 |

*Table 1: RTCP Bandwidth Needed for VoIP Feedback (Compound Reports Only)*

The final row of Table 1 (60 ms frames, reporting every 16 frames) sends RTCP reports once per second, giving an RTCP bandwidth overhead of 2.8 kbps.

The overhead can be reduced by sending some reports in reduced-size RTCP packets [RFC5506]. For example, if we alternate compound and reduced-size RTCP packets, i.e., Nrs = 1, the calculation gives the results shown in Table 2.

| Tf (seconds) | Nr (frames) | rtcp_bw (kbps) |
|---|---|---|
| 0.020 | 2 | 41.4 |
| 0.020 | 4 | 21.5 |
| 0.020 | 8 | 11.5 |
| 0.020 | 16 | 6.5 |
| 0.060 | 2 | 13.8 |
| 0.060 | 4 | 7.2 |
| 0.060 | 8 | 3.8 |
| 0.060 | 16 | 2.2 |

*Table 2: Required RTCP Bandwidth for VoIP Feedback (Alternating Compound and Reduced-Size Reports)*

The RTCP bandwidth needed for 60 ms frames, reporting every 16 frames (once per second), can be seen to drop to 2.2 kbps. This calculation can be repeated for other patterns of compound and reduced-size 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 0.11 s (every third frame of video). Using RTP/SAVPF with T_rr_interval=0, however, enables full utilization of the configured 5% RTCP bandwidth fraction.

The use of IPv6 will increase the overhead by 20 octets per packet, due to the increased size of the IPv6 header compared to IPv4, assuming no IP options in either case. This increases the size of compound packets to Sc = 162 + (2 * Nr) octets and reduced-size packets to Srs = 82 + (2 * Nr). Rerunning the calculations from Table 1 with these packet sizes gives the results shown in Table 3. As can be seen, there is a significant increase in overhead due to the use of IPv6.

| Tf (seconds) | Nr (frames) | rtcp_bw (kbps) |
|---|---|---|
| 0.020 | 2 | 64.8 |
| 0.020 | 4 | 33.2 |
| 0.020 | 8 | 17.4 |
| 0.020 | 16 | 9.5 |
| 0.060 | 2 | 21.6 |
| 0.060 | 4 | 11.1 |
| 0.060 | 8 | 5.8 |
| 0.060 | 16 | 3.2 |

*Table 3: RTCP Bandwidth Needed for VoIP Feedback (Compound Reports Only) Using IPv6*

Repeating the calculations from Table 2 using IPv6 gives the results shown in Table 4. As can be seen, the overhead still increases with IPv6 when a mix of compound and reduced-size reports is used, but the effect is less pronounced than with compound reports only.

| Tf (seconds) | Nr (frames) | rtcp_bw (kbps) |
|---|---|---|
| 0.020 | 2 | 49.2 |
| 0.020 | 4 | 25.4 |
| 0.020 | 8 | 13.5 |
| 0.020 | 16 | 7.5 |
| 0.060 | 2 | 16.4 |

| Tf (seconds) | Nr (frames) | rtcp_bw (kbps) |
|---|---|---|
| 0.060 | 4 | 8.5 |
| 0.060 | 8 | 4.5 |
| 0.060 | 16 | 2.5 |

*Table 4: Required RTCP Bandwidth for VoIP Feedback (Alternating Compound and Reduced-Size Reports) Using IPv6*

## 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, but this 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 the co-located audio and video SSRCs at each end point are aggregated [RFC8108], the optimizations in [RFC8861] are used, and RTCP congestion control feedback is sent [RFC8888].

As in Section 3.1, when all members are senders, the RTCP reporting interval calculation in Sections 6.2 and 6.3 [RFC3550] and in [RFC4585] reduces to:

```
Trtcp = n * Srtcp / Brtcp
```

where n is the number of members in the session, Srtcp is the average RTCP packet size in octets, and Brtcp is the RTCP bandwidth in octets per second.

The average RTCP packet size (Srtcp) depends on the amount of feedback sent in each RTCP packet, the number of members in the session, the size of source description (RTCP SDES) information sent, and 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. When the RTCP reporting group extensions are used, one of these SSRCs will be a reporting SSRC, to which the other SSRC will have delegated its reports. No reduced-size RTCP packets are sent.

The aggregated compound RTCP packet from the non-reporting SSRC will contain an RTCP SR packet, an RTCP SDES packet, and an RTCP Reporting Group Reporting Sources (RGRS) packet. The RTCP SR packet contains the 28-octet UDP/IP header (assuming IPv4 with no options) and sender information but no report blocks (since the reporting is delegated). The RTCP SDES packet will comprise a header (4 octets), the originating SSRC (4 octets), a CNAME chunk, a terminating chunk, and any padding. If the CNAME follows [RFC7022] and [RFC8834], the CNAME chunk will

be 18 octets in size and will be followed by one octet of padding and one terminating null octet to align the SDES packet to a 32-bit boundary ([RFC3550], Section 6.5), 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 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, a Reporting Group (RGRP) chunk, a terminating chunk, and any padding. If the CNAME follows [RFC7022] and [RFC8834], it will be 18 octets in size. The RGRP chunk similarly comprises 18 octets, the terminating chunk is comprised of 1 octet, and 3 octets of padding are needed, for a total of 48 octets. The RTCP congestion control feedback (CCFB) report comprises an 8-octet RTCP header and SSRC, a 4-octet report timestamp, and for each of the remote audio and video SSRCs, an 8-octet report header, 2 octets per packet reported upon, and padding to a 4-octet boundary if needed; that is, $8 + 4 + 8 + (2 * Nv) + 8 + (2 * Na)$, 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 SRTCP trailer and authentication tag, and a UDP/IPv4 header. The size of this RTCP packet is therefore $262 + (2 * Nv) + (2 * Na)$ octets. Since the aggregate RTCP packet contains reports from two SSRCs, the RTCP packet size is halved before use [RFC8108]. Accordingly, the size of the RTCP packets is:

```
Srtcp = (262 + (2 * Nv) + (2 * Na)) / 2
```

How many RTP packets does the RTCP XR congestion control feedback packet, included in these compound RTCP packets, report on? That is, what are the values of $Nv$ and $Na$? This depends on the RTCP reporting interval (Trtcp), the video bit rate and frame rate (Rf), the audio bit rate and framing interval, and whether the receiver chooses to send congestion control feedback in each RTCP packet it sends.

To simplify the calculation, assume it is desired to send one RTCP report for each frame of video received (i.e., Trtcp = 1 / Rf) and to include a congestion control feedback packet in each report. Assume that video has a constant bit rate and frame rate and that each frame of video has to fit into a 1500-octet MTU. Further, assume that the audio takes negligible bandwidth and that the audio framing interval can be varied within reasonable bounds, so that an integral number of audio frames align with video frame boundaries.

Table 5 shows the resulting values of $Nv$ and $Na$ (the number of video and audio packets covered by each congestion control feedback report) for a range of data rates and video frame rates, assuming congestion control feedback is sent once per video frame. The table also shows the result of inverting the RTCP reporting interval calculation to find the corresponding RTCP bandwidth (Brtcp). The RTCP bandwidth is given in kbps and as a fraction of the data rate.

It can be seen that, for example, with a data rate of 1024 kbps and a video sent at 30 frames per second, the RTCP congestion control feedback report sent for each video frame will include reports on 3 video packets and 2 audio packets. The RTCP bandwidth needed to sustain this reporting rate is 127.5 kbps (12% of the data rate). This assumes an audio framing interval of 16.67 ms, so that 2 audio packets are sent for each video frame.

| Data Rate (kbps) | Video Frame Rate: Rf | Video Packets per Report: Nv | Audio Packets per Report: Na | Required RTCP Bandwidth: Brtcp (kbps) |
|---|---|---|---|---|
| 100 | 8 | 1 | 6 | 34.5 (34%) |
| 200 | 16 | 1 | 3 | 67.5 (33%) |
| 350 | 30 | 1 | 2 | 125.6 (35%) |
| 700 | 30 | 2 | 2 | 126.6 (18%) |
| 700 | 60 | 1 | 1 | 249.4 (35%) |
| 1024 | 30 | 3 | 2 | 127.5 (12%) |
| 1400 | 60 | 2 | 1 | 251.2 (17%) |
| 2048 | 30 | 6 | 2 | 130.3 ( 6%) |
| 2048 | 60 | 3 | 1 | 253.1 (12%) |
| 4096 | 30 | 12 | 2 | 135.9 ( 3%) |
| 4096 | 60 | 6 | 1 | 258.8 ( 6%) |

*Table 5: Required RTCP Bandwidth, Reporting on Every Frame*

Use of reduced-size RTCP [RFC5506] would allow the SR and SDES packets to be omitted from some reports. These reduced-size 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 the SRTCP trailer and authentication tag and a UDP/IP header. That is, the size of the reduced-size packets would be (110 + (2 * Nv) + (2 * Na)) / 2 octets. Repeating the analysis above, but alternating compound and reduced-size reports, gives the results shown in Table 6.

| Data Rate (kbps) | Video Frame Rate: Rf | Video Packets per Report: Nv | Audio Packets per Report: Na | Required RTCP Bandwidth: Brtcp (kbps) |
|---|---|---|---|---|
| 100 | 8 | 1 | 6 | 25.0 (25%) |

| Data Rate (kbps) | Video Frame Rate: Rf | Video Packets per Report: Nv | Audio Packets per Report: Na | Required RTCP Bandwidth: Brtcp (kbps) |
|---|---|---|---|---|
| 200 | 16 | 1 | 3 | 48.5 (24%) |
| 350 | 30 | 1 | 2 | 90.0 (25%) |
| 700 | 30 | 2 | 2 | 90.9 (12%) |
| 700 | 60 | 1 | 1 | 178.1 (25%) |
| 1024 | 30 | 3 | 2 | 91.9 ( 8%) |
| 1400 | 60 | 2 | 1 | 180.0 (12%) |
| 2048 | 30 | 6 | 2 | 94.7 ( 4%) |
| 2048 | 60 | 3 | 1 | 181.9 ( 8%) |
| 4096 | 30 | 12 | 2 | 100.3 ( 2%) |
| 4096 | 60 | 6 | 1 | 187.5 ( 4%) |

*Table 6: 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 frame. Overall, it is clear that the RTCP overhead can be reasonable across the range of data and frame rates if RTCP is configured carefully.

As discussed in Section 3.1, the reporting overhead will increase if IPv6 is used, due to the increased size of the IPv6 header. Table 7 shows the overhead in this case, compared to Table 6. As can be seen, the increase in overhead due to IPv6 rapidly becomes less significant as the data rate increases.

| Data Rate (kbps) | Video Frame Rate: Rf | Video Packets per Report: Nv | Audio Packets per Report: Na | Required RTCP Bandwidth: Brtcp (kbps) |
|---|---|---|---|---|
| 100 | 8 | 1 | 6 | 27.5 (27%) |
| 200 | 16 | 1 | 3 | 53.5 (26%) |
| 350 | 30 | 1 | 2 | 99.4 (28%) |
| 700 | 30 | 2 | 2 | 100.3 (14%) |
| 700 | 60 | 1 | 1 | 196.9 (28%) |

| Data Rate (kbps) | Video Frame Rate: Rf | Video Packets per Report: Nv | Audio Packets per Report: Na | Required RTCP Bandwidth: Brtcp (kbps) |
|---|---|---|---|---|
| 1024 | 30 | 3 | 2 | 101.2 ( 9%) |
| 1400 | 60 | 2 | 1 | 198.8 (14%) |
| 2048 | 30 | 6 | 2 | 104.1 ( 5%) |
| 2048 | 60 | 3 | 1 | 200.6 ( 9%) |
| 4096 | 30 | 12 | 2 | 109.7 ( 2%) |
| 4096 | 60 | 6 | 1 | 206.2 ( 5%) |

*Table 7: Required RTCP Bandwidth, Reporting on Every Frame, with Reduced-Size Reports, Using IPv6*

# 4.  Discussion and Conclusions

Practical systems will generally send some non-media traffic on the same path as the media traffic. This can include Session Traversal Utilities for NAT (STUN) / Traversal Using Relays around NAT (TURN) packets to keep alive NAT bindings [RFC8445], WebRTC data channel packets [RFC8831], etc. Such traffic also needs congestion control, but the means by which this is achieved is out of the scope of this memo.

RTCP, as it is currently specified, cannot be used to send per-packet congestion feedback with reasonable overhead.

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 depends on the number of session participants, the RTCP bandwidth fraction, 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 reduced-size RTCP reports in between the regular reports. The improved compression of new video codecs exacerbates the reporting overhead for a given video quality level, although this is to some extent countered by the use of higher-quality video over time.

If it is desired to use RTCP in something close to its current form for congestion feedback in WebRTC, the multimedia congestion control algorithm needs to be designed to work with feedback sent every few frames, since that fits within the limitations of RTCP. The provided feedback will be more detailed than just an acknowledgement, however, and will provide a loss bitmap, relative arrival time, and received Explicit Congestion Notification (ECN) marks for each

packet sent. This will allow congestion control that is effective, if slowly responsive, to be implemented (there is guidance on providing effective congestion control in Section 3.1 of [RFC8085]).

The format described in [RFC8888] seems sufficient for the needs of congestion control feedback. There is little point optimizing 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 extensions described in [RFC5506] and sending reports less frequently. The use of header compression [RFC2508] [RFC3545] [RFC5795] can also be beneficial.

Further study of the scenarios of interest is needed to ensure that the analysis presented is applicable to other media topologies [RFC7667] 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 behavior 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 other means as discussed in [RFC7201].

## 6.  IANA Considerations

This document has no IANA actions.

## 7.  Normative References

[RFC2914]   Floyd, S., "Congestion Control Principles", BCP 41, RFC 2914, DOI 10.17487/
            RFC2914, September 2000, <https://www.rfc-editor.org/info/rfc2914>.

[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, <https://www.rfc-editor.org/info/rfc3550>.

[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, <https://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, <https://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, <https://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, <https://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, <https://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, <https://www.rfc-editor.org/info/rfc7201>.

[RFC8083] Perkins, C. and V. Singh, "Multimedia Congestion Control: Circuit Breakers for Unicast RTP Sessions", RFC 8083, DOI 10.17487/RFC8083, March 2017, <https://www.rfc-editor.org/info/rfc8083>.

[RFC8085] Eggert, L., Fairhurst, G., and G. Shepherd, "UDP Usage Guidelines", BCP 145, RFC 8085, DOI 10.17487/RFC8085, March 2017, <https://www.rfc-editor.org/info/rfc8085>.

[RFC8108] Lennox, J., Westerlund, M., Wu, Q., and C. Perkins, "Sending Multiple RTP Streams in a Single RTP Session", RFC 8108, DOI 10.17487/RFC8108, March 2017, <https://www.rfc-editor.org/info/rfc8108>.

[RFC8825] Alvestrand, H., "Overview: Real-Time Protocols for Browser-Based Applications", RFC 8825, DOI 10.17487/RFC8825, January 2021, <https://www.rfc-editor.org/info/rfc8825>.

[RFC8834] Perkins, C., Westerlund, M., and J. Ott, "Media Transport and Use of RTP in WebRTC", RFC 8834, DOI 10.17487/RFC8834, January 2021, <https://www.rfc-editor.org/info/rfc8834>.

[RFC8861] Lennox, J., Westerlund, M., Wu, Q., and C. Perkins, "Sending Multiple RTP Streams in a Single RTP Session: Grouping RTP Control Protocol (RTCP) Reception Statistics and Other Feedback", RFC 8861, DOI 10.17487/RFC8861, January 2021, <https://www.rfc-editor.org/info/rfc8861>.

[RFC8872] Westerlund, M., Burman, B., Perkins, C., Alvestrand, H., and R. Even, "Guidelines for Using the Multiplexing Features of RTP to Support Multiple Media Streams", RFC 8872, DOI 10.17487/RFC8872, January 2021, <https://www.rfc-editor.org/info/rfc8872>.

[RFC8888] Sarker, Z., Perkins, C., Singh, V., and M. Ramalho, "RTP Control Protocol (RTCP) Feedback for Congestion Control", RFC 8888, DOI 10.17487/RFC8888, January 2021, <https://www.rfc-editor.org/info/rfc8888>.

## 8. Informative References

[RFC2508]

Casner, S. and V. Jacobson, "Compressing IP/UDP/RTP Headers for Low-Speed Serial Links", RFC 2508, DOI 10.17487/RFC2508, February 1999, <https://www.rfc-editor.org/info/rfc2508>.

[RFC3449]    Balakrishnan, H., Padmanabhan, V., Fairhurst, G., and M. Sooriyabandara, "TCP Performance Implications of Network Path Asymmetry", BCP 69, RFC 3449, DOI 10.17487/RFC3449, December 2002, <https://www.rfc-editor.org/info/rfc3449>.

[RFC3545]    Koren, T., Casner, S., Geevarghese, J., Thompson, B., and P. Ruddy, "Enhanced Compressed RTP (CRTP) for Links with High Delay, Packet Loss and Reordering", RFC 3545, DOI 10.17487/RFC3545, July 2003, <https://www.rfc-editor.org/info/rfc3545>.

[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, <https://www.rfc-editor.org/info/rfc3611>.

[RFC5348]    Floyd, S., Handley, M., Padhye, J., and J. Widmer, "TCP Friendly Rate Control (TFRC): Protocol Specification", RFC 5348, DOI 10.17487/RFC5348, September 2008, <https://www.rfc-editor.org/info/rfc5348>.

[RFC5795]    Sandlund, K., Pelletier, G., and L. Jonsson, "The RObust Header Compression (ROHC) Framework", RFC 5795, DOI 10.17487/RFC5795, March 2010, <https://www.rfc-editor.org/info/rfc5795>.

[RFC7667]    Westerlund, M. and S. Wenger, "RTP Topologies", RFC 7667, DOI 10.17487/RFC7667, November 2015, <https://www.rfc-editor.org/info/rfc7667>.

[RFC8445]    Keranen, A., Holmberg, C., and J. Rosenberg, "Interactive Connectivity Establishment (ICE): A Protocol for Network Address Translator (NAT) Traversal", RFC 8445, DOI 10.17487/RFC8445, July 2018, <https://www.rfc-editor.org/info/rfc8445>.

[RFC8831]    Jesup, R., Loreto, S., and M. Tüxen, "WebRTC Data Channels", RFC 8831, DOI 10.17487/RFC8831, January 2021, <https://www.rfc-editor.org/info/rfc8831>.

[RFC9293]    Eddy, W., Ed., "Transmission Control Protocol (TCP)", STD 7, RFC 9293, DOI 10.17487/RFC9293, August 2022, <https://www.rfc-editor.org/info/rfc9293>.

# Acknowledgements

# Author's Address

**Colin Perkins**
University of Glasgow
School of Computing Science
Glasgow
G12 8QQ
United Kingdom
Email: csp@csperkins.org