RMCAT WG
Internet-Draft
Intended status: Informational
Expires: May 9, 2019

V. Singh
callstats.io
J. Ott
Technical University of Munich
S. Holmer
Google

November 5, 2018

Evaluating Congestion Control for Interactive Real-time Media
draft-ietf-rmcat-eval-criteria-08

Abstract

The Real-time Transport Protocol (RTP) is used to transmit media in telephony and video conferencing applications. This document describes the guidelines to evaluate new congestion control algorithms for interactive point-to-point real-time media.

Status of This Memo

Copyright Notice

Table of Contents

1. Introduction

This memo describes the guidelines to help with evaluating new congestion control algorithms for interactive point-to-point real time media. The requirements for the congestion control algorithm are outlined in [I-D.ietf-rmcat-cc-requirements]). This document builds upon previous work at the IETF: Specifying New Congestion Control Algorithms [RFC5033] and Metrics for the Evaluation of Congestion Control Algorithms [RFC5166].

The guidelines proposed in the document are intended to help prevent a congestion collapse, promote fair capacity usage and optimize the media flow's throughput. Furthermore, the proposed algorithms are expected to operate within the envelope of the circuit breakers defined in RFC8083 [RFC8083].

This document only provides broad-level criteria for evaluating a new congestion control algorithm. The minimal requirement for congestion control proposals is to produce or present results for the test scenarios described in [I-D.ietf-rmcat-eval-test] (Basic Test Cases). Additionally, proponents may produce evaluation results for the wireless test scenarios [I-D.ietf-rmcat-wireless-tests].

2. Terminology

The terminology defined in RTP [RFC3550], RTP Profile for Audio and Video Conferences with Minimal Control [RFC3551], RTCP Extended Report (XR) [RFC3611], Extended RTP Profile for RTCP-based Feedback (RTP/AVPF) [RFC4585] and Support for Reduced-Size RTCP [RFC5506] apply.

3. Metrics

This document specifies testing criteria for evaluating congestion control algorithms for RTP media flows. Proposed algorithms are to prove their performance by means of simulation and/or emulation experiments for all the cases described.

Each experiment is expected to log every incoming and outgoing packet (the RTP logging format is described in Section 3.1). The logging can be done inside the application or at the endpoints using PCAP (packet capture, e.g., tcpdump, wireshark). The following are calculated based on the information in the packet logs:

1. Sending rate, Receiver rate, Goodput (measured at 200ms intervals)

2. Packets sent, Packets received

3. Bytes sent, bytes received

4. Packet delay

5. Packets lost, Packets discarded (from the playout or de-jitter buffer)

6. If using, retransmission or FEC: post-repair loss

7. Self-Fairness and Fairness with respect to cross traffic: Experiments testing a given congestion control proposal must report on relative ratios of the average throughput (measured at coarser time intervals) obtained by each RTP media stream. In the presence of background cross-traffic such as TCP, the report must also include the relative ratio between average throughput of RTP media streams and cross-traffic streams.
   During static periods of a test (i.e., when bottleneck bandwidth is constant and no arrival/departure of streams), these report on relative ratios serve as an indicator of how fair the RTP streams share bandwidth amongst themselves and against cross-traffic streams. The throughput measurement interval should be set at a few values (for example, at 1s, 5s, and 20s) in order to measure fairness across different time scales.
   As a general guideline, the relative ratio between congestion controlled RTP flows with the same priority level and similar path RTT should be bounded between (0.333 and 3.) For example, see the test scenarios described in [I-D.ietf-rmcat-eval-test].

8. Convergence time: The time taken to reach a stable rate at startup, after the available link capacity changes, or when new flows get added to the bottleneck link.

9. Instability or oscillation in the sending rate: The frequency or number of instances when the sending rate oscillates between an high watermark level and a low watermark level, or vice-versa in a defined time window. For example, the watermarks can be set at 4x interval: 500 Kbps, 2 Mbps, and a time window of 500ms.

10. Bandwidth Utilization, defined as ratio of the instantaneous sending rate to the instantaneous bottleneck capacity. This metric is useful only when a congestion controlled RTP flow is by itself or competing with similar cross-traffic.

Note that the above metrics are all objective application-independent metrics. Refer to Section 3, in [I-D.ietf-netvc-testing] for objective metrics for evaluating codecs.

From the logs the statistical measures (min, max, mean, standard deviation and variance) for the whole duration or any specific part of the session can be calculated. Also the metrics (sending rate, receiver rate, goodput, latency) can be visualized in graphs as variation over time, the measurements in the plot are at 1 second intervals. Additionally, from the logs it is possible to plot the histogram or CDF of packet delay.

3.1. RTP Log Format

Having a common log format simplifies running analyses across and comparing different measurements. The log file SHOULD be tab or comma separated containing the following details:

```
Send or receive timestamp (unix)
RTP payload type
SSRC
RTP sequence no
RTP timestamp
marker bit
payload size
```

If the congestion control implements, retransmissions or FEC, the evaluation should report both packet loss (before applying error-resilience) and residual packet loss (after applying error-resilience).

4. List of Network Parameters

The implementors initially are encouraged to choose evaluation settings from the following values:

4.1. One-way Propagation Delay

Experiments are expected to verify that the congestion control is able to work across a broad range of path characteristics, also including challenging situations, for example over trans-continental and/or satellite links. Tests thus account for the following different latencies:

1. Very low latency: 0-1ms

2. Low latency: 50ms

3. High latency: 150ms

4. Extreme latency: 300ms

4.2. End-to-end Loss

Many paths in the Internet today are largely lossless but, with wireless networks and interference, towards remote regions, or in scenarios featuring high/fast mobility, media flows may exhibit substantial packet loss. This variety needs to be reflected appropriately by the tests.

To model a wide range of lossy links, the experiments can choose one of the following loss rates, the fractional loss is the ratio of packets lost and packets sent.

1. no loss: 0%

2. 1%

3. 5%

4. 10%

5. 20%

4.3. Drop Tail Router Queue Length

Routers SHOULD be configured to use Drop Trail queues in the experiments due to their (still) prevalent nature. Experimentation with AQM schemes is encouraged but not mandatory.

The router queue length is measured as the time taken to drain the FIFO queue. It has been noted in various discussions that the queue length in the current deployed Internet varies significantly. While the core backbone network has very short queue length, the home gateways usually have larger queue length. Those various queue lengths can be categorized in the following way:

1. QoS-aware (or short): 70ms

2. Nominal: 300-500ms

3. Buffer-bloated: 1000-2000ms

Here the size of the queue is measured in bytes or packets and to convert the queue length measured in seconds to queue length in bytes:

QueueSize (in bytes) = QueueSize (in sec) x Throughput (in bps)/8

4.4. Loss generation model

Many models for generating packet loss are available, some yield correlated, others independent losses; losses can also be extracted from packet traces. As a (simple) minimum loss model with minimal parameterization (i.e., the loss rate), independent random losses must be used in the evaluation.

It is known that independent loss models may reflect reality poorly and hence more sophisticated loss models could be considered. Suitable models for correlated losses includes the Gilbert-Elliot model and losses generated by modeling a queue including its (different) drop behaviors.

4.5. Jitter models

This section defines jitter models for the purposes of this document. When jitter is to be applied to both the congestion controlled RTP flow and any competing flow (such as a TCP competing flow), the competing flow will use the jitter definition below that does not allow for re-ordering of packets on the competing flow (see NR-RBPDV definition below).

Jitter is an overloaded term in communications. Its meaning is typically associated with the variation of a metric (e.g., delay) with respect to some reference metric (e.g., average delay or minimum delay). For example, RFC 3550 jitter is a smoothed estimate of jitter which is particularly meaningful if the underlying packet delay variation was caused by a Gaussian random process.

Because jitter is an overloaded term, we instead use the term Packet Delay Variation (PDV) to describe the variation of delay of individual packets in the same sense as the IETF IPPM WG has defined PDV in their documents (e.g., RFC 3393) and as the ITU-T SG16 has defined IP Packet Delay Variation (IPDV) in their documents (e.g., Y.1540).

Most PDV distributions in packet network systems are one-sided distributions (the measurement of which with a finite number of measurement samples result in one-sided histograms). In the usual packet network transport case there is typically one packet that transited the network with the minimum delay, then a majority of packets also transit the system within some variation from this minimum delay, and then a minority of the packets transit the network with delays higher than the median or average transit time (these are outliers). Although infrequent, outliers can cause significant deleterious operation in adaptive systems and should be considered in rate adaptation designs for RTP congestion control.

In this section we define two different bounded PDV characteristics, 1) Random Bounded PDV and 2) Approximately Random Subject to No-Reordering Bounded PDV.

The former, 1) Random Bounded PDV is presented for information only, while the latter, 2) Approximately Random Subject to No-Reordering Bounded PDV, must be used in the evaluation.

4.5.1. Random Bounded PDV (RBPDV)

The RBPDV probability distribution function (PDF) is specified to be of some mathematically describable function which includes some practical minimum and maximum discrete values suitable for testing. For example, the minimum value, x_min, might be specified as the minimum transit time packet and the maximum value, x_max, might be defined to be two standard deviations higher than the mean.

Since we are typically interested in the distribution relative to the mean delay packet, we define the zero mean PDV sample, z(n), to be z(n) = x(n) - x_mean, where x(n) is a sample of the RBPDV random variable x and x_mean is the mean of x.

We assume here that s(n) is the original source time of packet n and the post-jitter induced emission time, j(n), for packet n is j(n) = {[z(n) + x_mean] + s(n)}. It follows that the separation in the post-jitter time of packets n and n+1 is {[s(n+1)-s(n)] - [z(n)-z(n+1)]}. Since the first term is always a positive quantity, we note that packet reordering at the receiver is possible whenever the second term is greater than the first. Said another way, whenever the difference in possible zero mean PDV sample delays (i.e., [x_max-x_min]) exceeds the inter-departure time of any two sent packets, we have the possibility of packet re-ordering.

There are important use cases in real networks where packets can become re-ordered such as in load balancing topologies and during route changes. However, for the vast majority of cases there is no packet re-ordering because most of the time packets follow the same path. Due to this, if a packet becomes overly delayed, the packets after it on that flow are also delayed. This is especially true for mobile wireless links where there are per-flow queues prior to base station scheduling. Owing to this important use case, we define another PDV profile similar to the above, but one that does not allow for re-ordering within a flow.

4.5.2. Approximately Random Subject to No-Reordering Bounded PDV (NR-RPVD)

No Reordering RPDV, NR-RPVD, is defined similarly to the above with one important exception. Let serial(n) be defined as the serialization delay of packet n at the lowest bottleneck link rate (or other appropriate rate) in a given test. Then we produce all the post-jitter values for j(n) for n = 1, 2, ... N, where N is the length of the source sequence s to be offset-ed. The exception can be stated as follows: We revisit all j(n) beginning from index n=2, and if j(n) is determined to be less than [j(n-1)+serial(n-1)], we redefine j(n) to be equal to [j(n-1)+serial(n-1)] and continue for all remaining n (i.e., n = 3, 4, .. N). This models the case where the packet n is sent immediately after packet (n-1) at the bottleneck link rate. Although this is generally the theoretical minimum in that it assumes that no other packets from other flows are in-between packet n and n+1 at the bottleneck link, it is a reasonable assumption for per flow queuing.

We note that this assumption holds for some important exception cases, such as packets immediately following outliers. There are a multitude of software controlled elements common on end-to-end Internet paths (such as firewalls, ALGs and other middleboxes) which stop processing packets while servicing other functions (e.g., garbage collection). Often these devices do not drop packets, but rather queue them for later processing and cause many of the outliers. Thus NR-RPVD models this particular use case (assuming serial(n+1) is defined appropriately for the device causing the outlier) and thus is believed to be important for adaptation development for congestion controlled RTP streams.

4.5.3. Recommended distribution

Whether Random Bounded PDV or Approximately Random Subject to No-Reordering Bounded PDV, it is recommended that z(n) is distributed according to a truncated Gaussian for the above jitter models:

z(n) ~ |max(min(N(0, std^2), N_STD * std), -N_STD * std)|

where N(0, std^2) is the Gaussian distribution with zero mean and standard deviation std. Recommended values:

- std = 5 ms
- N_STD = 3

5. WiFi or Cellular Links

[I-D.ietf-rmcat-wireless-tests] describes the test cases to simulate networks with wireless links. The document describes mechanism to simulate both cellular and WiFi networks.

6. Traffic Models

6.1. TCP traffic model

Long-lived TCP flows will download data throughout the session and are expected to have infinite amount of data to send or receive. This roughly applies, for example, when downloading software distributions.

Each short TCP flow is modeled as a sequence of file downloads interleaved with idle periods. Not all short TCP flows start at the same time, i.e., some start in the ON state while others start in the OFF state.

The short TCP flows can be modeled as follows: 30 connections start simultaneously fetching small (30-50 KB) amounts of data. This covers the case where the short TCP flows are not fetching a video file.

The idle period between bursts of starting a group of TCP flows is typically derived from an exponential distribution with the mean value of 10 seconds.

[These values were picked based on the data available at http://httparchive.org/interesting.php as of October 2015].

Many different TCP congestion control schemes are deployed today. Therefore, experimentation with a range of different schemes, especially including CUBIC, is encouraged. Experiments MUST document in detail which congestion control schemes they tested against and which parameters were used.

6.2. RTP Video model

[I-D.ietf-rmcat-video-traffic-model] describes two types of video traffic models for evaluating candidate algorithms for RTP congestion control. The first model statistically characterizes the behavior of a video encoder. Whereas the second model uses video traces.

For example, test sequences are available at: [xiph-seq] and [HEVC-seq]. The currently chosen video streams are: Foreman and FourPeople.

6.3. Background UDP

Background UDP flow is modeled as a constant bit rate (CBR) flow. It will download data at a particular CBR rate for the complete session, or will change to particular CBR rate at predefined intervals. The inter packet interval is calculated based on the CBR and the packet size (is typically set to the path MTU size, the default value can be 1500 bytes).

Note that new transport protocols such as QUIC may use UDP but, due to their congestion control algorithms, will exhibit behavior conceptually similar in nature to TCP flows above and can thus be subsumed by the above, including the division into short- and long-lived flows. As QUIC evolves independently of TCP congestion control algorithms, its future congestion control SHOULD be considered as competing traffic as appropriate.

7. Security Considerations

Security issues have not been discussed in this memo.

8. IANA Considerations

There are no IANA impacts in this memo.

9. Contributors

The content and concepts within this document are a product of the discussion carried out in the Design Team.

Michael Ramalho provided the text for the Jitter model.

10. Acknowledgments

Much of this document is derived from previous work on congestion control at the IETF.

The authors would like to thank Harald Alvestrand, Anna Brunstrom, Luca De Cicco, Wesley Eddy, Lars Eggert, Kevin Gross, Vinayak Hegde, Randell Jesup, Mirja Kuehlewind, Karen Nielsen, Piers O'Hanlon, Colin Perkins, Michael Ramalho, Zaheduzzaman Sarker, Timothy B. Terriberry, Michael Welzl, Mo Zanaty, and Xiaoqing Zhu for providing valuable feedback on earlier versions of this draft. Additionally, also thank the participants of the design team for their comments and discussion related to the evaluation criteria.

11. References

11.1. Normative References

[I-D.ietf-rmcat-cc-requirements]
Jesup, R. and Z. Sarker, "Congestion Control Requirements for Interactive Real-Time Media", draft-ietf-rmcat-cc-requirements-09 (work in progress), December 2014.

[I-D.ietf-rmcat-wireless-tests]
Sarker, Z., Johansson, I., Zhu, X., Fu, J., Tan, W., and M. Ramalho, "Evaluation Test Cases for Interactive Real-Time Media over Wireless Networks", draft-ietf-rmcat-wireless-tests-05 (work in progress), June 2018.

[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>.

[RFC3551] Schulzrinne, H. and S. Casner, "RTP Profile for Audio and Video Conferences with Minimal Control", STD 65, RFC 3551, DOI 10.17487/RFC3551, July 2003, <https://www.rfc-editor.org/info/rfc3551>.

[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>.

[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>.

[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>.

[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>.

11.2. Informative References

[HEVC-seq]
HEVC, "Test Sequences", http://www.netlab.tkk.fi/~varun/test_sequences/ .

[I-D.ietf-netvc-testing]
Daede, T., Norkin, A., and I. Brailovskiy, "Video Codec Testing and Quality Measurement", draft-ietf-netvc-testing-07 (work in progress), July 2018.

[I-D.ietf-rmcat-eval-test]
Sarker, Z., Singh, V., Zhu, X., and M. Ramalho, "Test Cases for Evaluating RMCAT Proposals", draft-ietf-rmcat-eval-test-07 (work in progress), October 2018.

[I-D.ietf-rmcat-video-traffic-model]
Zhu, X., Cruz, S., and Z. Sarker, "Video Traffic Models for RTP Congestion Control Evaluations", draft-ietf-rmcat-video-traffic-model-06 (work in progress), November 2018.

[RFC5033] Floyd, S. and M. Allman, "Specifying New Congestion Control Algorithms", BCP 133, RFC 5033, DOI 10.17487/RFC5033, August 2007, <https://www.rfc-editor.org/info/rfc5033>.

[RFC5166] Floyd, S., Ed., "Metrics for the Evaluation of Congestion Control Mechanisms", RFC 5166, DOI 10.17487/RFC5166, March 2008, <https://www.rfc-editor.org/info/rfc5166>.

[RFC5681] Allman, M., Paxson, V., and E. Blanton, "TCP Congestion Control", RFC 5681, DOI 10.17487/RFC5681, September 2009, <https://www.rfc-editor.org/info/rfc5681>.

[SA4-LR] S4-050560, 3GPP., "Error Patterns for MBMS Streaming over UTRAN and GERAN", 3GPP S4-050560, 5 2008.

[TCP-eval-suite]
Lachlan, A., Marcondes, C., Floyd, S., Dunn, L., Guillier, R., Gang, W., Eggert, L., Ha, S., and I. Rhee, "Towards a Common TCP Evaluation Suite", Proc. PFLDnet. 2008, August 2008.

[xiph-seq]
Daede, T., "Video Test Media Set", https://people.xiph.org/~tdaede/sets/ .

Appendix A. Application Trade-off

Application trade-off is yet to be defined. see RMCAT requirements [I-D.ietf-rmcat-cc-requirements] document. Perhaps each experiment should define the application's expectation or trade-off.

A.1. Measuring Quality

No quality metric is defined for performance evaluation, it is currently an open issue. However, there is consensus that congestion control algorithm should be able to show that it is useful for interactive video by performing analysis using a real codec and video sequences.

Appendix B. Change Log

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

B.1. Changes in draft-ietf-rmcat-eval-criteria-07

Updated the draft according to the discussion at IETF-101.

- Updated the discussion on fairness. Thanks to Xiaoqing Zhu for providing text.

- Fixed a simple loss model and provided pointers to more sophisticated ones.

- Fixed the choice of the jitter model.

B.2. Changes in draft-ietf-rmcat-eval-criteria-06

- Updated Jitter.

B.3. Changes in draft-ietf-rmcat-eval-criteria-05

- Improved text surrounding wireless tests, video sequences, and short-TCP model.

B.4. Changes in draft-ietf-rmcat-eval-criteria-04

- Removed the guidelines section, as most of the sections are now covered: wireless tests, video model, etc.

- Improved Short TCP model based on the suggestion to use httparchive.org.

B.5. Changes in draft-ietf-rmcat-eval-criteria-03

- Keep-alive version.
- Moved link parameters and traffic models from eval-test

B.6. Changes in draft-ietf-rmcat-eval-criteria-02

- Incorporated fairness test as a working test.
- Updated text on mimimum evaluation requirements.

B.7. Changes in draft-ietf-rmcat-eval-criteria-01

- Removed Appendix B.
- Removed Section on Evaluation Parameters.

B.8. Changes in draft-ietf-rmcat-eval-criteria-00

- Updated references.
- Resubmitted as WG draft.

B.9. Changes in draft-singh-rmcat-cc-eval-04

- Incorporate feedback from IETF 87, Berlin.
- Clarified metrics: convergence time, bandwidth utilization.
- Changed fairness criteria to fairness test.
- Added measuring pre- and post-repair loss.
- Added open issue of measuring video quality to appendix.
- clarified use of DropTail and AQM.
- Updated text in "Minimum Requirements for Evaluation"

B.10. Changes in draft-singh-rmcat-cc-eval-03

- Incorporate the discussion within the design team.
- Added a section on evaluation parameters, it describes the flow and network characteristics.
- Added Appendix with self-fairness experiment.

o Changed bottleneck parameters from a proposal to an example set.

o

B.11. Changes in draft-singh-rmcat-cc-eval-02

o Added scenario descriptions.

B.12. Changes in draft-singh-rmcat-cc-eval-01

o Removed QoE metrics.

o Changed stability to steady-state.

o Added measuring impact against few and many flows.

o Added guideline for idle and data-limited periods.

o Added reference to TCP evaluation suite in example evaluation scenarios.

Authors' Addresses

Varun Singh
CALLSTATS I/O Oy
Runeberginkatu 4c A 4
Helsinki 00100
Finland

Email: varun@callstats.io
URI: https://www.callstats.io/about

Joerg Ott
Technical University of Munich
Faculty of Informatics
Boltzmannstrasse 3
Garching bei Muenchen, DE 85748
Germany

Email: ott@in.tum.de

Stefan Holmer
Google
Kungsbron 2
Stockholm 11122
Sweden

Email: holmer@google.com