

Internet Engineering Task Force (IETF)
Request for Comments: 8868
Category: Informational
ISSN: 2070-1721

V. Singh
callstats.io
J. Ott
Technical University of Munich
S. Holmer
Google
January 2021

Evaluating Congestion Control for Interactive Real-Time Media

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

This document is not an Internet Standards Track specification; it is published for informational purposes.

This document is a product of the Internet Engineering Task Force (IETF). It represents the consensus of the IETF community. It has received public review and has been approved for publication by the Internet Engineering Steering Group (IESG). Not all documents approved by the IESG are candidates for any level of Internet Standard; see Section 2 of RFC 7841.

Information about the current status of this document, any errata, and how to provide feedback on it may be obtained at <https://www.rfc-editor.org/info/rfc8868>.

Copyright Notice

Copyright (c) 2021 IETF Trust and the persons identified as the document authors. All rights reserved.

This document is subject to BCP 78 and the IETF Trust's Legal Provisions Relating to IETF Documents (<https://trustee.ietf.org/license-info>) in effect on the date of publication of this document. Please review these documents carefully, as they describe your rights and restrictions with respect to this document. Code Components extracted from this document must include Simplified BSD License text as described in Section 4.e of the Trust Legal Provisions and are provided without warranty as described in the Simplified BSD License.

Table of Contents

1. Introduction
2. Terminology
3. Metrics
 - 3.1. RTP Log Format
4. List of Network Parameters
 - 4.1. One-Way Propagation Delay
 - 4.2. End-to-End Loss
 - 4.3. Drop-Tail Router Queue Length
 - 4.4. Loss Generation Model
 - 4.5. Jitter Models
 - 4.5.1. Random Bounded PDV (RBPDV)
 - 4.5.2. Approximately Random Subject to No-Reordering Bounded PDV (NR-BPDV)
 - 4.5.3. Recommended Distribution
5. Traffic Models
 - 5.1. TCP Traffic Model

- 5.2. RTP Video Model
- 5.3. Background UDP
- 6. Security Considerations
- 7. IANA Considerations
- 8. References
 - 8.1. Normative References
 - 8.2. Informative References
- Contributors
- Acknowledgments
- Authors' Addresses

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 [RFC8836]. 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, to promote fair capacity usage, and to optimize the media flow's throughput. Furthermore, the proposed congestion control algorithms are expected to operate within the envelope of the circuit breakers defined in RFC 8083 [RFC8083].

This document only provides the broad set of network parameters and traffic models 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 [RFC8867] (Basic Test Cases), which also defines the specifics for the test cases. Additionally, proponents may produce evaluation results for the wireless test scenarios [RFC8869].

This document does not cover application-specific implications of congestion control algorithms and how those could be evaluated. Therefore, no quality metrics are defined for performance evaluation; quality metrics and the algorithms to infer those vary between media types. Metrics and algorithms to assess, e.g., the quality of experience, evolve continuously so that determining suitable choices is left for future work. However, there is consensus that each congestion control algorithm should be able to show that it is useful for interactive video by performing analysis using real codecs and video sequences and state-of-the-art quality metrics.

Beyond optimizing individual metrics, real-time applications may have further options to trade off performance, e.g., across multiple media; refer to the RMCAT requirements [RFC8836] document. Such trade-offs may be defined in the future.

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] applies.

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 [tcpdump], Wireshark [wireshark]). The following metrics 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 reports on relative ratios serve as an indicator of how fairly 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 1 s, 5 s, and 20 s) in order to measure fairness across different timescales.

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 [RFC8867].

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 500 ms.
10. Bandwidth utilization, defined as the 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 is competing with similar cross-traffic.

Note that the above metrics are all objective application-independent metrics. Refer to Section 3 of [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 one-second intervals. Additionally, from the logs, it is possible to plot the histogram or cumulative distribution function (CDF) of packet delay.

3.1. RTP Log Format

Having a common log format simplifies running analyses across different measurement setups and comparing their results.

Send or receive timestamp (Unix): <int>.<int> -- sec.usec decimal
RTP payload type <int> -- decimal

SSRC	<int>	-- hexadecimal
RTP sequence no	<int>	-- decimal
RTP timestamp	<int>	-- decimal
marker bit	0 1	-- character
Payload size	<int>	-- # bytes, decimal

Each line of the log file should be terminated with CRLF, CR, or LF characters. Empty lines are disregarded.

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

These data should suffice to compute the media-encoding independent metrics described above. Use of a common log will allow simplified post-processing and analysis across different implementations.

4. List of Network Parameters

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

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, including challenging situations, for example, over transcontinental and/or satellite links. Tests thus account for the following different latencies:

1. Very low latency: 0-1 ms
2. Low latency: 50 ms
3. High latency: 150 ms
4. Extreme latency: 300 ms

4.2. End-to-End Loss

Many paths in the Internet today are largely lossless; however, in scenarios featuring interference in wireless networks, sending to and receiving from remote regions, or 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-tail queues in the experiments due to their (still) prevalent nature. Experimentation with Active Queue Management (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 currently 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): 70 ms
2. Nominal: 300-500 ms
3. Buffer-bloated: 1000-2000 ms

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

$$\text{QueueSize (in bytes)} = \text{QueueSize (in sec)} \times \text{Throughput (in bps)} / 8$$

4.4. Loss Generation Model

Many models for generating packet loss are available: some generate correlated packet losses, others generate independent packet losses. In addition, packet 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 include the Gilbert-Elliott model [gilbert-elliott] and models that generate losses by modeling a queue with 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 reordering of packets on the competing flow (see NR-BPDV definition below).

Jitter is an overloaded term in communications. It is typically used to refer to the variation of a metric (e.g., delay) with respect to some reference metric (e.g., average delay or minimum delay). For example in RFC 3550, jitter is computed as the smoothed difference in packet arrival times relative to their respective expected arrival times, which is particularly meaningful if the underlying packet delay variation was caused by a Gaussian random process.

Because jitter is an overloaded term, we use the term Packet Delay Variation (PDV) instead to describe the variation of delay of individual packets in the same sense as the IETF IP Performance Metrics (IPPM) working group 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 results in one-sided histograms. In the usual packet network transport case, there is typically one packet that transited the network with the minimum delay; a (large) number of packets transit the network within some (smaller) positive variation from this minimum delay, and a (small) number 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 that 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 $z(n) = x(n) - x_{\text{mean}}$, 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:

$$j(n) = \{[z(n) + x_{\text{mean}}] + s(n)\}.$$

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 reordering.

There are important use cases in real networks where packets can become reordered, such as in load-balancing topologies and during route changes. However, for the vast majority of cases, there is no packet reordering 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 reordering within a flow.

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

No Reordering BPDV, NR-BPDV, is defined similarly to the above with one important exception. Let $\text{serial}(n)$ 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 $n = 1, 2, \dots, N$, where N is the length of the source sequence s to be offset. 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 $[j(n-1)+\text{serial}(n-1)]$, we redefine $j(n)$ to be equal to $[j(n-1)+\text{serial}(n-1)]$ and continue for all remaining n (i.e., $n = 3, 4, \dots, N$). 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, application-layer gateways, and other middleboxes) that 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-BPDV models this particular use case (assuming $\text{serial}(n+1)$ is defined appropriately for the device causing

the outlier) and 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:

$$z(n) \sim \left| \max(\min(N(0, \text{std}^2), N_STD * \text{std}), -N_STD * \text{std}) \right|$$

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

$\text{std} = 5 \text{ ms}$

$N_STD = 3$

5. Traffic Models

5.1. 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, evenly distributed. This covers the case where the short TCP flows are fetching web page resources rather than video files.

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 https://httparchive.org/reports/state-of-the-web?start=2015_10_01&end=2015_11_01&view=list as of October 2015.

Many different TCP congestion control schemes are deployed today. Therefore, experimentation with a range of different schemes, especially including CUBIC [RFC8312], is encouraged. Experiments must document in detail which congestion control schemes they tested against and which parameters were used.

5.2. RTP Video Model

[RFC8593] 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.

Sample video test sequences are available at [xiph-seq]. The following two video streams are the recommended minimum for testing: Foreman (CIF sequence) and FourPeople (720p); both come as raw video data to be encoded dynamically. As these video sequences are short (300 and 600 frames, respectively), they shall be stitched together repeatedly until the desired length is reached.

5.3. Background UDP

Background UDP flow is modeled as a constant bit rate (CBR) flow. It will download data at a particular CBR for the complete session, or will change to particular CBR at predefined intervals. The inter-

packet interval is calculated based on the CBR and the packet size (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; however, due to their congestion control algorithms, they will exhibit behavior conceptually similar in nature to TCP flows above and can thus be subsumed by the above, including the division into short-lived 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.

6. Security Considerations

This document specifies evaluation criteria and parameters for assessing and comparing the performance of congestion control protocols and algorithms for real-time communication. This memo itself is thus not subject to security considerations but the protocols and algorithms evaluated may be. In particular, successful operation under all tests defined in this document may suffice for a comparative evaluation but must not be interpreted that the protocol is free of risks when deployed on the Internet as briefly described in the following by example.

Such evaluations are expected to be carried out in controlled environments for limited numbers of parallel flows. As such, these evaluations are by definition limited and will not be able to systematically consider possible interactions or very large groups of communicating nodes under all possible circumstances, so that careful protocol design is advised to avoid incidentally contributing traffic that could lead to unstable networks, e.g., (local) congestion collapse.

This specification focuses on assessing the regular operation of the protocols and algorithms under consideration. It does not suggest checks against malicious use of the protocols -- by the sender, the receiver, or intermediate parties, e.g., through faked, dropped, replicated, or modified congestion signals. It is up to the protocol specifications themselves to ensure that authenticity, integrity, and/or plausibility of received signals are checked, and the appropriate actions (or non-actions) are taken.

7. IANA Considerations

This document has no IANA actions.

8. References

8.1. Normative References

- [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>>.
- [RFC8593] Zhu, X., Mena, S., and Z. Sarker, "Video Traffic Models for RTP Congestion Control Evaluations", RFC 8593, DOI 10.17487/RFC8593, May 2019, <<https://www.rfc-editor.org/info/rfc8593>>.
- [RFC8836] Jesup, R. and Z. Sarker, Ed., "Congestion Control Requirements for Interactive Real-Time Media", RFC 8836, DOI 10.17487/RFC8836, January 2021, <<https://www.rfc-editor.org/info/rfc8836>>.

8.2. Informative References

- [gilbert-elliott] Hasslinger, G. and O. Hohlfeld, "The Gilbert-Elliott Model for Packet Loss in Real Time Services on the Internet", 14th GI/ITG Conference - Measurement, Modelling and Evaluation [sic] of Computer and Communication Systems, March 2008, <<https://ieeexplore.ieee.org/document/5755057>>.
- [netvc-testing] Daede, T., Norkin, A., and I. Brailovski, "Video Codec Testing and Quality Measurement", Work in Progress, Internet-Draft, draft-ietf-netvc-testing-09, 31 January 2020, <<https://tools.ietf.org/html/draft-ietf-netvc-testing-09>>.
- [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>>.
- [RFC8312] Rhee, I., Xu, L., Ha, S., Zimmermann, A., Eggert, L., and R. Scheffenegger, "CUBIC for Fast Long-Distance Networks", RFC 8312, DOI 10.17487/RFC8312, February 2018, <<https://www.rfc-editor.org/info/rfc8312>>.
- [RFC8867] Sarker, Z., Singh, V., Zhu, X., and M. Ramalho, "Test Cases for Evaluating Congestion Control for Interactive Real-Time Media", RFC 8867, DOI 10.17487/RFC8867, January 2021, <<https://www.rfc-editor.org/info/rfc8867>>.
- [RFC8869] Sarker, Z., Zhu, X., and J. Fu, "Evaluation Test Cases for Interactive Real-Time Media over Wireless Networks", RFC 8869, DOI 10.17487/RFC8869, January 2021, <<https://www.rfc-editor.org/info/rfc8869>>.
- [tcpdump] "Homepage of tcpdump and libpcap", <<https://www.tcpdump.org/index.html>>.
- [wireshark] "Homepage of Wireshark", <<https://www.wireshark.org>>.
- [xiph-seq] Daede, T., "Video Test Media Set", <<https://media.xiph.org/video/derf/>>.

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 models (Section 4.5).

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 Kühlewind, 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 draft versions of this document. Additionally, thanks to the participants of the Design Team for their comments and discussion related to the evaluation criteria.

Authors' Addresses

Varun Singh
CALLSTATS I/O Oy
Rauhankatu 11 C
FI-00100 Helsinki
Finland

Email: varun.singh@iki.fi
URI: <https://www.callstats.io/>

Jörg Ott
Technical University of Munich
Department of Informatics
Chair of Connected Mobility
Boltzmannstrasse 3
85748 Garching
Germany

Email: ott@in.tum.de

Stefan Holmer
Google
Kungsbron 2
SE-11122 Stockholm
Sweden

Email: holmer@google.com