Stream:	Internet Engineering Task Force (IETF)
RFC:	9743
BCP:	133
Obsoletes:	5033
Category:	Best Current Practice
Published:	March 2025
ISSN:	2070-1721
Authors:	M. Duke, Ed.	G. Fairhurst, Ed.
---	---	---
	Google LLC	*University of Aberdeen*

RFC 9743
Specifying New Congestion Control Algorithms

Abstract

This document replaces RFC 5033, which discusses the principles and guidelines for standardizing new congestion control algorithms. It seeks to ensure that proposed congestion control algorithms operate without harm and efficiently alongside other algorithms in the global Internet. It emphasizes the need for comprehensive testing and validation to prevent adverse interactions with existing flows. This document provides a framework for the development and assessment of congestion control mechanisms, promoting stability across diverse network environments. It obsoletes RFC 5033 to reflect changes in the congestion control landscape.

Status of This Memo

This memo documents an Internet Best Current Practice.

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). Further information on BCPs is available in 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/rfc9743.

Copyright Notice

Table of Contents

1. Introduction

This document provides guidelines for the IETF to use when evaluating a proposed congestion control algorithm that differs from the general congestion control principles outlined in [RFC2914]. The guidance is intended to be useful to authors proposing congestion control algorithms and for the IETF community when evaluating whether a proposal is appropriate for publication in the RFC Series and for deployment in the Internet.

This document obsoletes [RFC5033], which was published in 2007 as a Best Current Practice for evaluating proposed congestion control algorithms for publication in Experimental or Proposed Standard RFCs.

The IETF specifies standard Internet congestion control algorithms in the RFC Series. These congestion control algorithms can suffer performance challenges when used in differing environments (e.g., high-speed networks, cellular and Wi-Fi wireless technologies, and long-distance satellite links), and also when flows carry specific workloads (e.g., Voice over IP (VoIP), gaming, and videoconferencing).

When [RFC5033] was published, TCP [RFC9293] was the primary focus of IETF congestion control efforts, with proposals typically discussed within the Internet Congestion Control Research Group (ICCRG). Concurrently, the Datagram Congestion Control Protocol (DCCP) [RFC4340] was developed to define new congestion control algorithms for datagram traffic, while the Stream Control Transmission Protocol (SCTP) [RFC9260] reused TCP congestion control algorithms.

Since then, several changes have occurred. The range of protocols utilizing congestion control algorithms has expanded to include QUIC [RFC9000] and RTP Media Congestion Avoidance Techniques (RMCAT) (e.g., [RFC8836]). Additionally, some alternative congestion control algorithms have been tested and deployed at scale without full IETF review. There is increased interest in specialized use cases, such as data centers (e.g., [RFC8257]), and in supporting a variety of upper-layer protocols and applications, such as real-time protocols. Moreover, the community has gained significant experience with congestion indications beyond packet loss.

Multicast congestion control is a considerably less mature field of study and is not in the scope of this document. However, Section 4 of [RFC8085] provides additional guidelines for multicast and broadcast usage of UDP.

Congestion control algorithms have been developed outside of the IETF, including at least two that saw large scale deployment. These include CUBIC [HRX08] and Bottleneck Bandwidth and Round-trip propagation time (BBR) [BBR].

CUBIC was documented in a research publication in 2007 [HRX08], and was then adopted as the default congestion control algorithm for the TCP implementation in Linux. It was already used in a significant fraction of TCP connections over the Internet before being documented in an Informational Internet-Draft in 2015, published as an Informational RFC in 2017 as [RFC8312] and then as a Proposed Standard in 2023 [RFC9438].

At the time of writing, BBR is being developed as an internal research project by Google, with the first implementation contributed to Linux kernel 4.19 in 2016. It was described in an Internet-Draft in 2018, which has been regularly updated to document the evolving versions of the algorithm [BBR]. BBR is currently widely used for Google services using either TCP or QUIC and is also widely deployed outside of Google.

We cannot say whether the original authors of [RFC5033] expected that developers would be waiting for IETF review before widely deploying a new congestion control algorithm over the Internet, but the examples of CUBIC and BBR illustrate that deployment of new algorithms is not, in fact, gated by the publication of the algorithm as an RFC.

Nevertheless, a specification for a congestion control algorithm provides a number of advantages:

- It can help implementers, operators, and other interested parties develop a shared understanding of how the algorithm works and how it is expected to behave in various scenarios and configurations.
- It can help potential contributors understand the algorithm, which can make it easier for them to suggest improvements and/or identify limitations. Furthermore, the specification can help multiple contributors align on a consensus change to the algorithm.
- A specification that is accessible to anyone can circumvent the issue that some implementers may be unable to read open-source reference implementations due to the constraints of some open-source licenses.

Beyond helping develop specific algorithm proposals, guidelines can also serve as a reminder to potential inventors and developers of the multiple facets of the congestion control problem.

The evaluation guidelines in this document are intended to be consistent with the congestion control principles from [RFC2914] related to preventing congestion collapse, considering fairness, and optimizing a flow's own performance in terms of throughput, delay, and loss. [RFC2914] also discusses the goal of avoiding a congestion control "arms race" among competing transport protocols.

This document does not give hard-and-fast requirements for an appropriate congestion control algorithm. Rather, the document provides a set of criteria that should be considered and weighed by the developers of alternative algorithms and by the IETF in the context of each proposal.

The high-order criterion for advancing any proposal within the IETF is a serious scientific study of the pros and cons that occur when the proposal is considered for publication by the IETF or before it is deployed at a large scale.

After initial studies, authors are encouraged to write a specification of their proposal for publication in the RFC Series. This allows others to understand and investigate the wealth of proposals in this space.

This document is intended to reduce the barriers to entry for new congestion control work to the IETF. As such, proponents of new congestion control algorithms ought not to interpret these criteria as a checklist of requirements before approaching the IETF. Instead, proponents are encouraged to think about these issues beforehand and have the willingness to do the work implied by the remainder of this document.

2. Specification of Requirements

The key words "**MUST**", "**MUST NOT**", "**REQUIRED**", "**SHALL**", "**SHALL NOT**", "**SHOULD**", "**SHOULD NOT**", "**RECOMMENDED**", "**NOT RECOMMENDED**", "**MAY**", and "**OPTIONAL**" in this document are to be interpreted as described in BCP 14 [RFC2119] [RFC8174] when, and only when, they appear in all capitals, as shown here.

3. Guidelines for Authors

3.1. Evaluation Guidelines

This document does not provide specific evaluation methods, short of Internet-scale deployment and measurement, to test the criteria described below. There are multiple possible approaches to evaluation. Each has a role, and the most appropriate approach depends on the criteria being evaluated and the maturity of the specification.

For many algorithms, an initial evaluation will consider individual protocol mechanisms in a simulator to analyze their stability and safety across a wide range of conditions, including overload. For example, [RFC8869] describes evaluation test cases for interactive real-time media over wireless networks. Such results could also be published or discussed in IRTF research groups, such as ICCRG and MAPRG.

Before a proposed congestion control algorithm is published as an Experimental or Standards Track RFC, the community **SHOULD** gain practical experience with implementations and experience using the algorithm. Implementations by independent teams can help provide assurance that a specification has avoided assumptions or ambiguity. An independent evaluation by multiple teams helps provide assurance that the design meets the evaluation criteria and can assess typical interactions with other traffic. This evaluation could use an emulated laboratory environment or a controlled experiment (within a limited domain or at the Internet scale). Evidence of results is normally considered by the working group in deciding if a specification is ready for publication and ought to be documented in any request for the working group to publish the specification.

Publication might occur without multiple implementations if a single implementation is widely used, open source, and shown to have a positive impact on the Internet, particularly if the target status is Experimental.

3.2. Document-Status Guidelines

The guidelines in this document apply to specifications of congestion control algorithms that seek publication as an RFC via the IETF Stream with an Experimental or Standards Track status. The evaluation of either status involves the same questions, but with different expectations for both the answers and the degree of certainty of those answers.

Specifications on congestion control algorithms without empirical evidence of Internet-scale deployment **MUST** seek Experimental status, unless they are not targeted for general use.

Specifications that seek to be published as Experimental IETF Stream RFCs ought to explain the reason for the status and what further information would be required to progress to a Standards Track RFC. For example, Section 12 of [RFC6928] provides "Usage and Deployment Recommendations" that describe the experiments expected by the TCPM Working Group. Section 4 of [RFC4614] provides other examples of extensions that were considered experimental when the specification was published (note that [RFC4614] has since been obsoleted by [RFC7414]).

Experimental specifications **SHOULD NOT** be deployed as a default. They **SHOULD** only be deployed in situations where they are being actively measured and where it is possible to deactivate them if there are signs of pathological behavior.

Specifications on congestion control algorithms with a record of measured Internet-scale deployment **MAY** directly seek Standards Track status if there is solid data that reflects that the algorithm is safe and the design is stable, guided by the considerations in Section 6. However, the existence of this data does not waive the other considerations in this document.

Each specification submitted for publication as an RFC is **REQUIRED** to include a statement in the abstract indicating whether or not there is IETF consensus that the proposed congestion control algorithm is considered safe for use on the Internet. Each such specification is also **REQUIRED** to include a statement in the abstract describing environments where the protocol is not recommended for deployment. There can be environments where the congestion control algorithm is deemed safe for use, but it is still not recommended for use because it does not perform well for the user.

Examples of such statements exist in [RFC3649], which specifies HighSpeed TCP and includes a statement in the abstract stating that the proposed congestion control algorithm is experimental but may be deployed in the Internet. In contrast, the Quick-Start document [RFC4782] includes a paragraph in the abstract stating that the mechanism is only being proposed for use in controlled environments. The abstract specifies environments where the Quick-Start request could give false positives (and therefore would be unsafe for incremental deployment where some routers forward but do not process the option). The abstract also specifies environments where packets containing the Quick-Start request could be dropped in the network; in such an environment, Quick-Start would not be unsafe to deploy, but deployment is not recommended because it could lead to unnecessary delays for the connections attempting to use Quick-Start. The Quick-Start method is discussed as an example in [RFC9049].

Strictly speaking, documents for publication as Informational RFCs from the IETF Stream need not meet all of the criteria in this document, as they do not carry a formal recommendation from the IETF community. Instead, the community judges the publication of these Informational RFCs based on the value of their addition to the information captured by the RFC Series.

Although it is out of scope for this document, proponents of a new algorithm could alternatively seek publication of their specification as an Informational or Experimental RFC via the Internet Research Task Force (IRTF) Stream. In general, these algorithms are expected to be less mature than ones that follow the procedures in this document for publication via the IETF Stream. Authors documenting deployed congestion control algorithms that cannot be changed by IETF or IRTF review are invited to seek publication of their specification as an Informational RFC via the Independent Submission Stream.

4. Specifying Algorithms for Use in Controlled Environments

Algorithms can be designed for general Internet deployment or for use in controlled environments [RFC8799]. Within a controlled environment, an operator can ensure that flows are isolated from other Internet flows or they might allow these flows to share resources with other Internet flows. A data center is an example of a controlled environment that often deploys fabrics with rich signaling from switches to endpoints.

Algorithms that rely on specific functions or configurations in the network need to provide a reference or specification for these functions (such as an RFC or another stable specification). For publication of a specification of one of these algorithms to proceed, the IETF will need to consider whether a working group exists that can properly assess the network-layer aspects and their interaction with the congestion control.

In evaluating a new proposal for use in a controlled environment, the IETF needs to understand the usage, e.g., how the usage is scoped to the controlled environment, whether the algorithm will share resources with Internet traffic, and consider what could happen if used in a protocol that is bridged across an Internet path. Algorithms that are designed to be confined to a controlled environment and are not intended for use in the general Internet might instead seek real-world data for those environments. In such cases, the evaluation criteria in the remainder of this document might not apply.

5. Evaluation Criteria

As previously noted, authors of a specification on a congestion control algorithm are expected to conduct a comprehensive evaluation of the advantages and disadvantages of any congestion control algorithms presented to the IETF community. The following guidelines are intended to assist authors and the community in this endeavor. While these guidelines provide a helpful framework, they should not be regarded as an exhaustive checklist as concerns beyond the scope of these guidelines may also arise.

When considering a proposed congestion control algorithm, the community **MUST** consider the criteria in the following sections. These criteria will be evaluated in various domains (see Sections 6 and 7).

Some of the sections below will list criteria that **SHOULD** be met. It could happen that these criteria are not, in fact, met by the proposal. In such cases, the community **MUST** document whether not meeting the criteria is acceptable, for example, if there are practical limitations on carrying out an evaluation of the criteria.

The requirement that the community consider a criterion does not imply that the result needs to be described in an RFC: there is no formal requirement to document the results, although normal IETF policies for archiving proceedings will provide a record.

This document, except where otherwise noted, does not provide normative guidance on the acceptable thresholds for any of these criteria. Instead, the community will use these evaluations as an input when considering whether to progress the proposed algorithm.

5.1. Single Algorithm Behavior

The criteria in the following subsections evaluate the congestion control algorithm when one or more flows using that algorithm share a bottleneck link (i.e., with no flows using a differing congestion control algorithm).

5.1.1. Protection Against Congestion Collapse

A congestion control algorithm should either stop sending when the packet drop rate exceeds some threshold [RFC3714] or include some notion of "full backoff". For "full backoff", at some point, the algorithm would reduce the sending rate to one packet per round-trip time; then, it would exponentially back off the time between single packet transmissions if the congestion persists. Exactly when either "full backoff" or a pause in sending comes into play will be algorithm specific. However, as discussed in [RFC2914] and [RFC8961], this requirement is crucial to protect the network in times of extreme (and persistent) congestion.

If full backoff is used, this test does not require that the mechanism be identical to that of TCP (see [RFC6298] and [RFC8961]). For example, this does not preclude full backoff mechanisms that would give flows with different round-trip times comparable capacity during backoff.

5.1.2. Protection Against Bufferbloat

A congestion control algorithm should try to avoid maintaining excessive queues in the network. Exactly how the algorithm achieves this is algorithm specific; see [RFC8961] and [RFC8085] for requirements.

"Bufferbloat" refers to the building of excessive queues in the network [BUFFERBLOAT]. Many network routers are configured with very large buffers. The Standards Track RFCs [RFC5681] and [RFC9438] describing the Reno and CUBIC congestion control algorithms (respectively) send at progressively higher rates until a First In, First Out (FIFO) buffer completely fills; then packet losses occur. Every connection passing through that bottleneck experiences increased latency due to the high buffer occupancy. This adds unwanted latency that negatively impacts highly interactive applications such as videoconferencing or games, but it also affects routine web browsing and video playing.

This problem has been widely discussed since 2011 [BUFFERBLOAT], but was not discussed in the congestion control principles published in September 2002 [RFC2914]. The Reno and CUBIC congestion control algorithms do not address this problem, but a new congestion control algorithm has the opportunity to improve the state of the art.

5.1.3. Protection Against High Packet Loss

A congestion control algorithm should try to avoid causing excessively high rates of packet loss. To accomplish this, it should avoid excessive increases in sending rate and reduce its sending rate if experiencing high packet loss.

The first version of the BBR algorithm [BBRv1] failed this requirement. Experimental evaluation [BBRv1-EVALUATION] showed that it caused a sustained rate of packet loss when multiple BBRv1 flows shared a bottleneck and the buffer size was less than roughly one and a half times the Bandwidth Delay Product (BDP). This was unsatisfactory, and, indeed, further versions provided a fix for this aspect of BBR [BBR].

This requirement does not imply that the algorithm should react to packet losses in exactly the same way as congestion control algorithms described in current Standards Track RFCs (e.g., [RFC5681]).

5.1.4. Fairness Within the Proposed Congestion Control Algorithm

When multiple competing flows all use the same proposed congestion control algorithm, the specification should explore how the capacity is shared among the competing flows. Capacity fairness can be important when a small number of similar flows compete to fill a bottleneck. However, it can also not be useful, for example, when comparing flows that seek to send at different rates or if some of the flows do not last sufficiently long to approach asymptotic behavior.

5.1.5. Short Flows

A great deal of congestion control analysis concerns the steady-state behavior of long flows. However, many Internet flows are relatively short lived. Many short-lived flows today remain in the "slow start" mode of operation [RFC5681] that commonly features exponential congestion window growth because the flow never experiences congestion (e.g., packet loss).

A proposed congestion control algorithm **MUST** consider how new and short-lived flows affect long-lived flows, and vice versa.

5.2. Mixed Algorithm Behavior

The mixed algorithm behavior criteria evaluate the interaction of the proposed congestion control algorithms being specified with commonly deployed congestion control algorithms.

In contexts where differing congestion control algorithms are used, it is important to understand whether the proposed congestion control algorithm could result in more harm than algorithms published in previous Standards Track RFCs (e.g., [RFC5681], [RFC9002], and [RFC9438]) to flows sharing a common bottleneck. The measure of harm is not restricted to unequal capacity, but also ought to consider metrics such as the introduced latency or an increase in packet loss. An evaluation **MUST** assess the potential to cause starvation, including assurance that a loss of all feedback (e.g., detected by expiry of a retransmission time out) results in backoff.

5.2.1. Existing General-Purpose Congestion Control

A proposed congestion control algorithm **MUST** be evaluated when competing against standard IETF congestion controls (e.g., [RFC5681], [RFC9002], and [RFC9438]). A proposed congestion control algorithm that has a significantly negative impact on flows using standard congestion control might be suspect, and this aspect should be part of the community's decision making with

regards to the suitability of the proposed congestion control algorithm. The community should also consider other non-standard congestion control algorithms that are known to be widely deployed.

Note that this guideline is not a requirement for strict Reno or CUBIC friendliness as a prerequisite for a proposed congestion control mechanism to advance to Experimental or Standards Track status. As an example, HighSpeed TCP is a congestion control mechanism that is specified in an Experimental RFC and is not TCP friendly in all environments. When a new congestion control algorithm is deployed, the existing major algorithm deployments need to be considered to avoid severe performance degradation. Note that this guideline does not constrain the interaction with flows that are not best effort.

As an example from an Experimental RFC, fairness with standard TCP is discussed in Sections 4 and 6 of [RFC3649], and using spare capacity is discussed in Sections 6, 11.1, and 12 of [RFC3649].

5.2.2. Real-Time Congestion Control

General-purpose algorithms need to coexist in the Internet with real-time congestion control algorithms, which in general have finite throughput requirements (i.e., they do not seek to utilize all available capacity) and more strict latency bounds. See [RFC8836] for a description of the characteristics of this use case and the resulting requirements.

[RFC8868] provides suggestions for real-time congestion control design and [RFC8867] suggests test cases. [RFC9392] describes some considerations for the RTP Control Protocol (RTCP). In particular, real-time flows can use less frequent feedback (acknowledgment) than that provided by reliable transports. This document does not change the Informational status of those RFCs.

A proposed congestion control algorithm **SHOULD** consider coexistence with widely deployed real-time congestion control algorithms. Regrettably, at the time of writing (2024), many algorithms with detailed public specifications are not widely deployed, while many widely deployed real-time congestion control algorithms have incomplete public specifications. It is hoped that this situation will change.

To the extent that behavior of widely deployed algorithms is understood, proponents of a proposed congestion control algorithm can analyze and simulate a proposal's interaction with those algorithms. To the extent that they are not, experiments can be conducted where possible.

Real-time flows can be directed into distinct queues via Differentiated Services Code Points (DSCPs) or other mechanisms, which can substantially reduce the interplay with other traffic. However, a proposal targeting general Internet use cannot assume this is always the case.

Section 7.2 describes the impact of network transport circuit breaker algorithms. [RFC8083] also defines a minimal set of RTP circuit breakers that operate end-to-end across a path. This identifies conditions under which a sender needs to stop transmitting media data to protect the network from excessive congestion. It is expected that, in the absence of long-lived excessive congestion, RTP applications running on best-effort IP networks will be able to operate without triggering these circuit breakers.

5.2.3. Short and Long Flows

The effect on short-lived and long-lived flows using other common congestion control algorithms **MUST** be evaluated, as in Section 5.1.5.

5.3. Other Criteria

5.3.1. Differences with Congestion Control Principles

A proposed congestion control algorithm **MUST** clearly explain any deviations from [RFC2914] and [RFC7141].

5.3.2. Incremental Deployment

A congestion control algorithm proposal **MUST** discuss whether it allows for incremental deployment in the targeted environment. For a mechanism targeted for deployment in the current Internet, the proposal **SHOULD** discuss what is known (if anything) about the correct operation of the mechanisms with some of the equipment in the current Internet (e.g., routers, transparent proxies, WAN optimizers, intrusion detection systems, home routers, and the like).

Similarly, if the proposed congestion control algorithm is intended only for specific environments (and not the global Internet), the proposal **SHOULD** consider how this intention is to be realized. The IETF community will have to address the question of whether the scope can be enforced by stating the restrictions or whether additional protocol mechanisms are required to enforce this scoping. The answer will necessarily depend on the proposed change.

As an example from an Experimental RFC, deployment issues of Quick-Start are discussed in Sections 10.3 and 10.4 of [RFC4782].

6. General Use

The criteria in Section 5 will be evaluated in the scenarios described in the following subsections. Unless a proposed congestion control algorithm specification of the IETF Stream explicitly forbids use on the public Internet, there **MUST** be IETF consensus that it meets the criteria in these scenarios for the proposed congestion control algorithm to progress.

The evaluation of each scenario **SHOULD** occur over a representative range of bandwidths, delays, and queue depths. Of course, the set of parameters representative of the public Internet will change over time.

These criteria are intended to capture a statistically dominant set of Internet conditions. In the case that a proposed congestion control algorithm has been tested at Internet scale, the results from that deployment are often useful for answering these questions.

6.1. Paths with Tail-Drop Queues

The performance of a congestion control algorithm is affected by the queue discipline applied at the bottleneck link. The drop-tail queue discipline (using a FIFO buffer) **MUST** be evaluated. See Section 7.1 for evaluation of other queue disciplines.

6.2. Tunnel Behavior

When a proposed congestion control algorithm relies on explicit signals from the path, the proposal **MUST** consider the effect of traffic passing through a tunnel, where routers may not be aware of the flow.

Designers of tunnels and similar encapsulations might need to consider nested congestion control interactions, for example, when the Explicit Congestion Notification (ECN) is used by both an IP and lower-layer technology [ECN-ENCAPS].

6.3. Wired Paths

Wired networks are usually characterized by extremely low rates of packet loss except for those due to queue drops. They tend to have stable aggregate capacity, usually higher than other types of links, and low non-queueing delay. Because the properties are relatively simple, wired links are typically used as a "baseline" case even if they are not always the bottleneck link in the modern Internet.

6.4. Wireless Paths

While the early Internet was dominated by wired links, the properties of wireless links have become important to Internet performance. In particular, a proposed congestion control algorithm should be evaluated in situations where some packet losses are due to radio effects rather than router queue drops. The link capacity varies over time due to changing link conditions, and media-access delays and link-layer retransmission lead to increased jitter in round-trip times. See [RFC3819] and Section 16 of [TOOLS] for further discussion of wireless properties.

7. Special Cases

The criteria in Section 5 will be evaluated in the scenarios described in the following subsections, unless the proposed congestion control algorithm specifically excludes its use in a scenario. For these specific use cases, the IETF community **MAY** allow a proposal to progress even if the criteria indicate an unsatisfactory result for these scenarios.

In general, measurements from Internet-scale deployments might not expose the properties of operation in each of these scenarios because they are not as ubiquitous as the general-use scenarios.

7.1. Active Queue Management (AQM)

The proposed congestion control algorithm **SHOULD** be evaluated under a variety of bottleneck-queue disciplines. The effect of an AQM discipline can be hard to detect by Internet evaluation. At a minimum, a proposal should reason about an algorithm's response to various AQM disciplines. Simulation or empirical results are, of course, valuable.

Some of the AQM techniques that might have an impact on a proposed congestion control algorithm include:

- Flow Queue CoDel (FQ-CoDel) [RFC8290];
- Proportional Integral controller Enhanced (PIE) [RFC8033]; and
- Low Latency, Low Loss, and Scalable Throughput (L4S) [RFC9332].

A proposed congestion control algorithm that sets one of the two Explicit Congestion Transport (ECT) codepoints in the IP header can gain the benefits of receiving Explicit Congestion Notification - Congestion Experienced (ECN-CE) signals from an on-path AQM [RFC8087]. Use of ECN (see [RFC3168] and [RFC9332]) requires the congestion control algorithm to react when it receives a packet with an ECN-CE marking. This reaction needs to be evaluated to confirm that the algorithm conforms with the requirements of the ECT codepoint that was used.

Note that evaluation of AQM techniques -- as opposed to their impact on a specific proposed congestion control algorithm -- is out of scope of this document. [RFC7567] describes design considerations for AQMs.

7.2. Operation with the Envelope Set by Network Circuit Breakers

Some equipment in the network uses an automatic mechanism to continuously monitor the use of resources by a flow or aggregate set of flows [RFC8084]. Such a network transport circuit breaker can automatically detect excessive congestion; when detected, it can terminate (or significantly reduce the rate of) the flow(s). A well-designed congestion control algorithm ought to react before the flow uses excessive resources; therefore, it will operate within the envelope set by network transport circuit breaker algorithms.

7.3. Paths with Varying Delay

An Internet path can include simple links, where the minimum delay is the propagation delay, and any additional delay can be attributed to link buffering. This cannot be assumed. An Internet path can also include complex subnetworks where the minimum delay changes over various time scales, resulting in a minimum delay that is not stationary.

Varying delay occurs when a subnet changes the forwarding path to optimize capacity, resilience, etc. It could also arise when a subnet uses a capacity-management method where the available resource is periodically distributed among the active nodes. A node might then have to buffer data until an assigned transmission opportunity or until the physical path changes (e.g., when the length of a wireless path changes or when the physical layer changes its mode of operation).

Variation also arises when traffic with a higher priority DSCP preempts transmission of traffic with a lower class. In these cases, the delay varies as a function of external factors, and attempting to infer congestion from an increase in the delay results in reduced throughput. This variation in the delay over short timescales (jitter) might not be distinguishable from jitter that results from other effects.

A proposed congestion control algorithm **SHOULD** be evaluated to ensure its operation is robust when there is a significant change in the minimum delay.

7.4. Internet of Things and Constrained Nodes

The "Internet of Things" (IoT) is a broad concept, but when evaluating a proposed congestion control algorithm, it is often associated with unique characteristics. For example, IoT nodes might be more constrained in power, CPU, or other parameters than conventional Internet hosts. This might place limits on the complexity of any given algorithm. These power and radio constraints might make the volume of control packets in a given algorithm a key evaluation metric.

Extremely low-power links can lead to very low throughput and a low bandwidth-delay product, which is well below the standard operating range of most Internet flows.

Furthermore, many IoT applications do not a have a human in the loop; therefore, they might have weaker latency constraints because they do not relate to a user experience. Congestion control algorithms still may need to share the path with other flows with different constraints.

7.5. Paths with High Delay

A proposed congestion control algorithm ought not to presume that all general Internet paths have a low delay. Some paths include links that contribute much more delay than for a typical Internet path. Satellite links often have delays longer than is typical for wired paths [RFC2488] and high-delay-bandwidth products [RFC3649].

Paths can also present a variable delay as described in Section 7.3.

7.6. Misbehaving Nodes

A proposed congestion control algorithm **SHOULD** explore how the algorithm performs with non-compliant senders, receivers, or routers. In addition, the proposal should explore how a proposed congestion control algorithm performs with outside attackers. This can be particularly important for proposed congestion control algorithms that involve explicit feedback from routers along the path.

As an example from an Experimental RFC, performance with misbehaving nodes and outside attackers is discussed in Sections 9.4, 9.5, and 9.6 of [RFC4782]. This includes discussion of:

- misbehaving senders and receivers;
- collusion between misbehaving routers;
- misbehaving middleboxes; and

- the potential use of Quick-Start to attack routers or to tie up available Quick-Start bandwidth.

7.7. Extreme Packet Reordering

A proposed congestion control algorithm ought not to presume that all general Internet paths reliably deliver packets in order. [RFC4653] discusses the effect of extreme packet reordering.

7.8. Transient Events

A proposed congestion control algorithm **SHOULD** consider how it would perform in the presence of transient events such as a sudden onset of congestion, a routing change, or a mobility event. Routing changes, link disconnections, intermittent link connectivity, and mobility are discussed in more detail in Section 16 of [TOOLS].

As an example from an Experimental RFC, a response to transient events is discussed in Section 9.2 of [RFC4782].

7.9. Sudden Changes in the Path

An IETF transport is not tied to a specific Internet path or type of path. The set of routers that form a path can and do change with time. This will cause the properties of the path to change with respect to time. A proposed congestion control algorithm **MUST** evaluate the impact of changes in the path and be robust to changes in path characteristics on the interval of common Internet rerouting intervals.

7.10. Multipath Transport

Multipath transport protocols permit more than one path to be differentiated and used by a single connection at the sender. A multipath sender can schedule which packets travel on which of its active paths. This enables a trade-off in timeliness and reliability. There are various ways that multipath techniques can be used.

One example use is to provide failover from one path to another when the original path is no longer viable or provides inferior performance. Designs need to independently track the congestion state of each path and demonstrate independent congestion control for each path being used. Authors of a proposed multipath congestion control algorithm that implements path failover **MUST** evaluate the harm to performance resulting from a change in the path and show that this does not result in flow starvation. Synchronization of failover (e.g., where multiple flows change their path on similar time frames) can also contribute to harm and/or reduce fairness. These effects also ought to be evaluated.

Another example use is concurrent multipath, where the transport protocol simultaneously schedules a flow to aggregate the capacity across multiple paths. The Internet provides no guarantee that different paths (e.g., using different endpoint addresses) are disjoint. This introduces additional implications. A congestion control algorithm proposal **MUST** evaluate the potential harm to other flows when the multiple paths share a common congested bottleneck or share resources that are coupled between different paths, such as an overall capacity limit. A

proposal **SHOULD** consider the potential for harm to other flows. Synchronization of congestion control mechanisms (e.g., where multiple flows change their behavior on similar time frames) can also contribute to harm and/or reduce fairness. These effects also ought to be evaluated.

At the time of writing (2024), there are currently no Standards Track RFCs for concurrent multipath, but there is an Experimental RFC [RFC6356] that specifies a concurrent multipath congestion control algorithm for Multipath TCP (MPTCP) [RFC8684].

7.11. Data Centers

Data centers are characterized by very low latencies (< 2 ms). Many workloads involve bursty traffic where many nodes complete a task at the same time. As a controlled environment, data centers often deploy fabrics that employ rich signaling from switches to endpoints. Furthermore, the operator can often limit the number of operating congestion control algorithms.

For these reasons, data center congestion controls are often distinct from those running elsewhere on the Internet (see Section 4). A proposed congestion control need not coexist well with all other algorithms if it is intended for data centers, but the proposal **SHOULD** indicate which are expected to safely coexist with it.

8. Security Considerations

This document does not represent a change to any aspect of the TCP/IP protocol suite; therefore, it does not directly impact Internet security. The implementation of various facets of the Internet's current congestion control algorithms do have security implications (e.g., as outlined in [RFC5681]).

Proposed congestion control algorithms **MUST** examine any potential security or privacy issues that may arise from their design.

9. IANA Considerations

This document has no IANA actions.

10. References

10.1. Normative References

[RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, DOI 10.17487/RFC2119, March 1997, <https://www.rfc-editor.org/info/rfc2119>.

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

[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>.

[RFC7141] Briscoe, B. and J. Manner, "Byte and Packet Congestion Notification", BCP 41, RFC 7141, DOI 10.17487/RFC7141, February 2014, <https://www.rfc-editor.org/info/rfc7141>.

[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>.

[RFC8084] Fairhurst, G., "Network Transport Circuit Breakers", BCP 208, RFC 8084, DOI 10.17487/RFC8084, March 2017, <https://www.rfc-editor.org/info/rfc8084>.

[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>.

[RFC8174] Leiba, B., "Ambiguity of Uppercase vs Lowercase in RFC 2119 Key Words", BCP 14, RFC 8174, DOI 10.17487/RFC8174, May 2017, <https://www.rfc-editor.org/info/rfc8174>.

[RFC8961] Allman, M., "Requirements for Time-Based Loss Detection", BCP 233, RFC 8961, DOI 10.17487/RFC8961, November 2020, <https://www.rfc-editor.org/info/rfc8961>.

[RFC9002] Iyengar, J., Ed. and I. Swett, Ed., "QUIC Loss Detection and Congestion Control", RFC 9002, DOI 10.17487/RFC9002, May 2021, <https://www.rfc-editor.org/info/rfc9002>.

[RFC9438] Xu, L., Ha, S., Rhee, I., Goel, V., and L. Eggert, Ed., "CUBIC for Fast and Long-Distance Networks", RFC 9438, DOI 10.17487/RFC9438, August 2023, <https://www.rfc-editor.org/info/rfc9438>.

10.2. Informative References

[BBR] Cardwell, N., Cheng, Y., Yeganeh, S. H., Swett, I., and V. Jacobson, "BBR Congestion Control", Work in Progress, Internet-Draft, draft-cardwell-iccrg-bbr-congestion-control-02, 7 March 2022, <https://datatracker.ietf.org/doc/html/draft-cardwell-iccrg-bbr-congestion-control-02>.

[BBRv1] Cardwell, N., Cheng, Y., Yeganeh, S. H., and V. Jacobson, "BBR Congestion Control", Work in Progress, Internet-Draft, draft-cardwell-iccrg-bbr-congestion-control-00, 3 July 2017, <https://datatracker.ietf.org/doc/html/draft-cardwell-iccrg-bbr-congestion-control-00>.

[BBRv1-EVALUATION] Hock, M., Bless, R., and M. Zitterbart, "Experimental evaluation of BBR congestion control", 2017 IEEE 25th International Conference on Network Protocols (ICNP), DOI 10.1109/ICNP.2017.8117540, 2017, <https://ieeexplore.ieee.org/document/8117540>.

[BUFFERBLOAT] Nichols, K. and J. Gettys, "Bufferbloat: Dark Buffers in the Internet", ACM Queue Volume 9, Issue 11, November 2011, <https://queue.acm.org/detail.cfm?id=2071893>.

[ECN-ENCAPS] Briscoe, B. and J. Kaippallimalil, "Guidelines for Adding Congestion Notification to Protocols that Encapsulate IP", BCP 89, RFC 9599, DOI 10.17487/RFC9599, August 2024, <https://www.rfc-editor.org/info/rfc9599>.

[HRX08] Ha, S., Rhee, I., and L. Xu, "CUBIC: a new TCP-friendly high-speed TCP variant", ACM SIGOPS Operating Systems Review, vol. 42, no. 5, pp. 64-74, DOI 10.1145/1400097.1400105, July 2008, <https://doi.org/10.1145/1400097.1400105>.

[RFC2488] Allman, M., Glover, D., and L. Sanchez, "Enhancing TCP Over Satellite Channels using Standard Mechanisms", BCP 28, RFC 2488, DOI 10.17487/RFC2488, January 1999, <https://www.rfc-editor.org/info/rfc2488>.

[RFC3168] Ramakrishnan, K., Floyd, S., and D. Black, "The Addition of Explicit Congestion Notification (ECN) to IP", RFC 3168, DOI 10.17487/RFC3168, September 2001, <https://www.rfc-editor.org/info/rfc3168>.

[RFC3649] Floyd, S., "HighSpeed TCP for Large Congestion Windows", RFC 3649, DOI 10.17487/RFC3649, December 2003, <https://www.rfc-editor.org/info/rfc3649>.

[RFC3714] Floyd, S., Ed. and J. Kempf, Ed., "IAB Concerns Regarding Congestion Control for Voice Traffic in the Internet", RFC 3714, DOI 10.17487/RFC3714, March 2004, <https://www.rfc-editor.org/info/rfc3714>.

[RFC3819] Karn, P., Ed., Bormann, C., Fairhurst, G., Grossman, D., Ludwig, R., Mahdavi, J., Montenegro, G., Touch, J., and L. Wood, "Advice for Internet Subnetwork Designers", BCP 89, RFC 3819, DOI 10.17487/RFC3819, July 2004, <https://www.rfc-editor.org/info/rfc3819>.

[RFC4340] Kohler, E., Handley, M., and S. Floyd, "Datagram Congestion Control Protocol (DCCP)", RFC 4340, DOI 10.17487/RFC4340, March 2006, <https://www.rfc-editor.org/info/rfc4340>.

[RFC4614] Duke, M., Braden, R., Eddy, W., and E. Blanton, "A Roadmap for Transmission Control Protocol (TCP) Specification Documents", RFC 4614, DOI 10.17487/RFC4614, September 2006, <https://www.rfc-editor.org/info/rfc4614>.

[RFC4653] Bhandarkar, S., Reddy, A. L. N., Allman, M., and E. Blanton, "Improving the Robustness of TCP to Non-Congestion Events", RFC 4653, DOI 10.17487/RFC4653, August 2006, <https://www.rfc-editor.org/info/rfc4653>.

[RFC4782] Floyd, S., Allman, M., Jain, A., and P. Sarolahti, "Quick-Start for TCP and IP", RFC 4782, DOI 10.17487/RFC4782, January 2007, <https://www.rfc-editor.org/info/rfc4782>.

[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>.

[RFC6298] Paxson, V., Allman, M., Chu, J., and M. Sargent, "Computing TCP's Retransmission Timer", RFC 6298, DOI 10.17487/RFC6298, June 2011, <https://www.rfc-editor.org/info/rfc6298>.

[RFC6356] Raiciu, C., Handley, M., and D. Wischik, "Coupled Congestion Control for Multipath Transport Protocols", RFC 6356, DOI 10.17487/RFC6356, October 2011, <https://www.rfc-editor.org/info/rfc6356>.

[RFC6928] Chu, J., Dukkipati, N., Cheng, Y., and M. Mathis, "Increasing TCP's Initial Window", RFC 6928, DOI 10.17487/RFC6928, April 2013, <https://www.rfc-editor.org/info/rfc6928>.

[RFC7414] Duke, M., Braden, R., Eddy, W., Blanton, E., and A. Zimmermann, "A Roadmap for Transmission Control Protocol (TCP) Specification Documents", RFC 7414, DOI 10.17487/RFC7414, February 2015, <https://www.rfc-editor.org/info/rfc7414>.

[RFC7567] Baker, F., Ed. and G. Fairhurst, Ed., "IETF Recommendations Regarding Active Queue Management", BCP 197, RFC 7567, DOI 10.17487/RFC7567, July 2015, <https://www.rfc-editor.org/info/rfc7567>.

[RFC8033] Pan, R., Natarajan, P., Baker, F., and G. White, "Proportional Integral Controller Enhanced (PIE): A Lightweight Control Scheme to Address the Bufferbloat Problem", RFC 8033, DOI 10.17487/RFC8033, February 2017, <https://www.rfc-editor.org/info/rfc8033>.

[RFC8087] Fairhurst, G. and M. Welzl, "The Benefits of Using Explicit Congestion Notification (ECN)", RFC 8087, DOI 10.17487/RFC8087, March 2017, <https://www.rfc-editor.org/info/rfc8087>.

[RFC8257] Bensley, S., Thaler, D., Balasubramanian, P., Eggert, L., and G. Judd, "Data Center TCP (DCTCP): TCP Congestion Control for Data Centers", RFC 8257, DOI 10.17487/RFC8257, October 2017, <https://www.rfc-editor.org/info/rfc8257>.

[RFC8290] Hoeiland-Joergensen, T., McKenney, P., Taht, D., Gettys, J., and E. Dumazet, "The Flow Queue CoDel Packet Scheduler and Active Queue Management Algorithm", RFC 8290, DOI 10.17487/RFC8290, January 2018, <https://www.rfc-editor.org/info/rfc8290>.

[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>.

[RFC8684] Ford, A., Raiciu, C., Handley, M., Bonaventure, O., and C. Paasch, "TCP Extensions for Multipath Operation with Multiple Addresses", RFC 8684, DOI 10.17487/RFC8684, March 2020, <https://www.rfc-editor.org/info/rfc8684>.

[RFC8799] Carpenter, B. and B. Liu, "Limited Domains and Internet Protocols", RFC 8799, DOI 10.17487/RFC8799, July 2020, <https://www.rfc-editor.org/info/rfc8799>.

[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>.

[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>.

[RFC8868] Singh, V., Ott, J., and S. Holmer, "Evaluating Congestion Control for Interactive Real-Time Media", RFC 8868, DOI 10.17487/RFC8868, January 2021, <https://www.rfc-editor.org/info/rfc8868>.

[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>.

[RFC9000] Iyengar, J., Ed. and M. Thomson, Ed., "QUIC: A UDP-Based Multiplexed and Secure Transport", RFC 9000, DOI 10.17487/RFC9000, May 2021, <https://www.rfc-editor.org/info/rfc9000>.

[RFC9049] Dawkins, S., Ed., "Path Aware Networking: Obstacles to Deployment (A Bestiary of Roads Not Taken)", RFC 9049, DOI 10.17487/RFC9049, June 2021, <https://www.rfc-editor.org/info/rfc9049>.

[RFC9260] Stewart, R., Tüxen, M., and K. Nielsen, "Stream Control Transmission Protocol", RFC 9260, DOI 10.17487/RFC9260, June 2022, <https://www.rfc-editor.org/info/rfc9260>.

[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>.

[RFC9332] De Schepper, K., Briscoe, B., Ed., and G. White, "Dual-Queue Coupled Active Queue Management (AQM) for Low Latency, Low Loss, and Scalable Throughput (L4S)", RFC 9332, DOI 10.17487/RFC9332, January 2023, <https://www.rfc-editor.org/info/rfc9332>.

[RFC9392] Perkins, C., "Sending RTP Control Protocol (RTCP) Feedback for Congestion Control in Interactive Multimedia Conferences", RFC 9392, DOI 10.17487/RFC9392, April 2023, <https://www.rfc-editor.org/info/rfc9392>.

[TOOLS] Floyd, S. and E. Kohler, "Tools for the Evaluation of Simulation and Testbed Scenarios", Work in Progress, Internet-Draft, draft-irtf-tmrg-tools-05, 23 February 2008, <https://datatracker.ietf.org/doc/html/draft-irtf-tmrg-tools-05>.

Acknowledgments

Sally Floyd and Mark Allman were the authors of this document's predecessor, [RFC5033], which served the community well for over a decade.

Thanks to Richard Scheffenegger for helping to get this revision process started.

The editors would like to thank Mohamed Boucadair, Neal Cardwell, Reese Enghardt, Jonathan Lennox, Matt Mathis, Zahed Sarker, Juergen Schoenwaelder, Dave Taht, Sean Turner, Michael Welzl, Magnus Westerlund, and Greg White for suggesting improvements to this document.

Discussions with Lars Eggert and Aaron Falk seeded the original [RFC5033]. Bob Briscoe, Gorry Fairhurst, Doug Leith, Jitendra Padhye, Colin Perkins, Pekka Savola, members of TSVWG, and participants at the TCP Workshop at Microsoft Research all provided feedback and contributions to that document. It also drew from [RFC5166].

Contributors

Christian Huitema
Private Octopus, Inc.
Email: huitema@huitema.net

Authors' Addresses

Martin Duke (EDITOR)
Google LLC
Email: martin.h.duke@gmail.com

Godred Fairhurst (EDITOR)
University of Aberdeen
Email: gorry@erg.abdn.ac.uk