Guidelines for Internet Congestion Control at Endpoints
draft-fairhurst-tsvwg-cc-00

Abstract

This document provides guidance on the design of methods to avoid congestion collapse and to provide congestion control. Recommendations and requirements on this topic are distributed across many documents in the RFC series. It seeks to gather and consolidate these recommendations. This is intended to provide input to the design of new congestion control methods in protocols, such as IETF QUIC.

The present document is for discussion and comment by the IETF.

Status of This Memo

This Internet-Draft is submitted in full conformance with the provisions of BCP 78 and BCP 79.

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF). Note that other groups may also distribute working documents as Internet-Drafts. The list of current Internet-Drafts is at https://datatracker.ietf.org/drafts/current/.

Internet-Drafts are draft documents valid for a maximum of six months and may be updated, replaced, or obsoleted by other documents at any time. It is inappropriate to use Internet-Drafts as reference material or to cite them other than as "work in progress."

This Internet-Draft will expire on January 2, 2020.

Copyright Notice

Copyright (c) 2019 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
1. Introduction

The IETF has specified Internet transports (e.g., TCP [ID.ietf-tcpm-rfc793bis], UDP [RFC0768], UDP-Lite [RFC3828], SCTP [RFC4960], and DCCP [RFC4340]) as well as protocols layered on top of these transports (e.g., RTP, QUIC [I-D.ietf-quic-transport], SCTP/UDP [RFC6951], DCCP/UDP) and transports that work directly over the IP network layer. These transports are implemented in endpoints (Internet hosts or routers acting as endpoints) and are designed to detect and react to network congestion.

Recommendations and requirements on this topic are distributed across many documents in the RFC series. This document seeks to gather and consolidate these recommendations. This is intended to provide input to the design of new congestion control methods in protocols.
2. Terminology

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in [RFC2119].

Other terminology is directly copied the cited RFCs.

3. Principles of Congestion Control

This section describes principles for providing congestion control.

3.1. A Diversity of Path Characteristics

The path between endpoints (sometimes called "internet Hosts") consists of the endpoint protocol stack (implementing the transport) and a succession of links and network devices (routers or middleboxes) that provide connectivity across the network.

Internet transports do not usually rely upon prior reservation of capacity along the path they use. In the absence of such a resource reservation, endpoints are unable to determine a safe rate start or continue their transmission. The use of an Internet path therefore requires a combination of end-to-end transport mechanisms to detect and respond to changes in the capacity available across the network path. Buffering (an increase in latency) or loss (discard of a packet) arises when the traffic arriving at a link or network exceed the resources available. A network device that does not support Active Queue Management (AQM) [RFC7567] typically uses a drop-tail policy to drop excess IP packets when its queue becomes full. Although losses are not always due to congestion (loss may be due to link corruption, receiver overrun, etc. [RFC3819]), endpoints have to conservatively presume that loss is potentially due to congestion and reduce the sending rate of their flows to reflect the available capacity.

A path that is not congested can still experience an increased latency when the path multiplexes the traffic of multiple flows, and/or when the level of traffic is (transiently) higher than the currently available capacity. As with loss, latency can also be incurred for other reasons [RFC3819] (e.g. Quality of service scheduling, link radio resource management/bandwidth on demand, transient outages, link retransmission, and connection/resource setup below the IP layer).

The use of a path impacts any flows (possibly from or to other endpoints) that share (multiplex their data) over a common network device or link.
Principles include:

- The design of a congestion controller needs to consider the wide range of path characteristics presented by the variety of Internet paths. Transports MUST be designed such that they operate safely and effectively over common paths.

- An endpoint cannot assume that a particular packet header is passed transparently by the path or a particular forwarding treatment applies. The supported set of packet headers and forwarding can also change once a flow has commenced.

- A design MUST assume path characteristics can change over relatively short intervals of time (i.e. characteristics discovered do not necessarily remain valid for multiple Round Trip Times, RTTs). In particular, they need to measure and adapt to the path(s) they use path capacity, and the estimated RTT used for timers.

- A design MUST assume that the set of network devices encountered along a path can change with time. This MUST be robust to reconfiguration of network devices, reset of devices.

3.2. Flow Multiplexing and Congestion

It is normal to observe some perturbation in latency or loss to traffic when it shares a common network bottleneck with other traffic. This impact needs to be considered and Internet flows ought to implement appropriate safeguards to avoid inappropriate impact on other flows that share the resources along a path. Congestion control methods satisfy this requirement and therefore also avoid congestion collapse [@ARTICLE{author = {Bob Briscoe}, title = {Flow Rate Fairness: Dismantling a Religion}, journal = {ACM CCR}, year = {2007} ].

Internet transports should react to avoid congestion that impacts other flows sharing the path, and need to be designed to avoid starving other flows of capacity. This could include methods seeking to equally distribute resources between sharing flows, but this is explicitly not a requirement for design of network devices.

The Requirements for Internet Hosts [RFC1122] formally mandates that endpoints perform congestion control. "Because congestion control is critical to the stable operation of the Internet, applications and other protocols that choose to use UDP as an Internet transport must employ mechanisms to prevent congestion collapse and to establish some degree of fairness with concurrent traffic [RFC2914]. They may also need to implement additional mechanisms, depending on how they..."
use UDP" [RFC8085].  [RFC2309] also discussed the dangers of congestion-unresponsive flows and states that "all UDP-based streaming applications should incorporate effective congestion avoidance mechanisms." [RFC7567] and [RFC8085] reaffirm this.

An endpoint can become aware of congestion by various means. A signal that indicates congestion on the end-to-end network path, must result in a congestion control reaction by the transport to reduce the maximum rate permitted by the sending endpoint [RFC8087].

The general recommendation in the UDP Guidelines [RFC8085] is therefore that applications SHOULD leverage existing congestion control techniques, such as those defined for TCP [RFC5681], TFRC [RFC5348], SCTP [RFC4960], and other IETF-defined transports. This is because there are many trade offs and details that can have a serious impact on the performance of congestion control for the application they support and other traffic that seeks to share the resources along the path over which they communicate.

Section 3.6 notes that by default, IETF specifications target deployment on the general Internet. Experience has however shown that successful protocols developed in one specific context or for a particular application tend to become used in a wider range of contexts. Experience has however shown that successful protocols developed in one specific context or for a particular application tend to become used in a wider range of contexts.

Principles include:

- [RFC1122] mandates that endpoints perform congestion control.

- Transports need to avoid inducing flow starvation to other flows sharing resources along the path they use.

- "If an application or protocol chooses not to use a congestion-controlled transport protocol, it SHOULD control the rate at which it sends UDP datagrams to a destination host, in order to fulfill the requirements of [RFC2914]", as stated in [RFC8085].

- Transports that do not target Internet deployment need to be constrained to only operate in a controlled environment (e.g. see Section 3.6 of [RFC8085]) and provide appropriate mechanisms to prevent traffic accidentally leaving the controlled environment [RFC8084].
3.3. Avoiding Congestion Collapse

A significant pathology can arise when a poorly designed transport creates congestion. This can result in severe service degradation or "Internet meltdown". This phenomenon was first observed during the early growth phase of the Internet in the mid 1980s [RFC896] [RFC970]; it is technically called "congestion collapse" and was a key focus of [RFC2309].

- Endpoints MUST control their flows to avoid Congestion Collapse.
- Endpoints MUST employ exponential backoff to their traffic when they detect persistent congestion.
- Endpoints MUST treat a loss of all feedback (e.g., RTO expiry) as a tentative indication of congestion collapse, reacting until the path characteristics can again be confirmed.
- Network devices should provide mechanisms to avoid congestion collapse (e.g., priority forwarding of control information, and starvation detection and protection [RFC7567]).

4. Guidelines for performing Congestion Control

This section provides guidance for designers of a new transport protocol that decide to implement congestion control and its associated mechanisms.

4.1. Connection Initialization

When a connection or flow to a new destination is established, the endpoints have little information about the characteristics of the network path. This section describes how a flow starts transmission over such a path.

Flow Start: A new flow between a local and a remote endpoint cannot assume that capacity is available at the start of the flow, unless it uses a mechanism to explicitly reserve capacity. In the absence of a capacity signal, a flow MUST therefore start slowly.

The slow-start algorithm is the accepted standard for flow startup [RFC5681]. TCP uses the notion of an Initial Window (IW [RFC3390] updated by [RFC6928]) to define the initial volume of data that can be sent on a path. This is not the smallest burst, nor the smallest window - it is considered a safe starting point for a network that is not suffering persistent congestion, and applicable until feedback about the path is received.
initial sending rate needs to be viewed as tentative until the capacity is confirmed to be available.

Initial RTO: When a flow sends the first packet it typically has no way to know the actual RTT of the path it uses. The values used to initialise the Retransmission Timeout (RTO) is therefore a trade off that has important consequences on the overall Internet stability [RFC6928] [RFC8085]. In the absence of any knowledge about the latency of a path, the RTO MUST be conservatively set to no less than 1 second. Values shorter than 1 second can be problematic (see the appendix of [RFC6298]).

Initial RTO Expiry: If the RTO timer expires while awaiting completion of the connection setup (in TCP, the ACK of a SYN segment), and the implementation is using an RTO less than 3 seconds, the sender can resend the connection setup. The RTO MUST then be re-initialized to increase it to 3 seconds when data transmission begins (i.e., after the three-way handshake completes) [RFC6298] [RFC8085]. This conservative increase is necessary to avoid congestion collapse when many flows retransmit across a shared bottleneck with restricted capacity.

Initial Measured RTO: Once an RTT measurement is available (e.g., through reception of an acknowledgement), this value must be adjusted, and MUST take into account the RTT variance. For the first (sample this variance cannot be determined, and a sender must therefore initialise the variance to RTT/2 (see equation 2.2 of [RFC6928] and related text for UDP in section 3.1.1 of [RFC8085]).

Current State: A congestion controller MAY assume that recently used capacity between a pair of endpoint addresses is an indication of capacity available in the next RTT between the same endpoints (and react accordingly if this is not confirmed to be true).

Cached State: An endpoint that has recently used the same path between a local and remote endpoint could also have additional state that lets a flow take-over utilising the capacity that was previously consumed (e.g., in the last RTT) by another flow. In TCP, this mechanism is referred to as TCP Control Block (TCB) sharing [RFC2140] [ID.ietf-tcpm-2140bis]. This and other information can be used to suggest a faster initial sending rate, but MUST be viewed as tentative until the capacity is confirmed to be available. A sender MUST reduce its rate if the actual used capacity is not confirmed within the current RTT interval.
4.2. Timers and Retransmission

This section describes mechanisms to detect and provide retransmission, and to protect the network in the absence of timely feedback.

Loss Detection: Loss detection occurs after a sender determines there is no delivery confirmation within an expected period of time. Retransmission mechanisms MAY utilise a measure of the RTT of a path to detect loss before the period specified by the RTO [RFC8085].

Detection can also be performed using the time-ordering of transmission (as in TCP DupACK), or a combination of using a timer and ordering information to trigger retransmission of data [ID.ietf-tcpm-rack-05].

Retransmission: Retransmission of lost packets or messages is a common reliability mechanism. When a loss is detected, the sender can choose to retransmit the lost data, ignore the loss, or send other data. Any transmission consumes network capacity, therefore retransmissions MUST NOT increase the network load in response to congestion loss (which worsens that congestion) [RFC8085]. Any method that sends additional data following loss is responsible for congestion control of the retransmissions (and any other packets sent) as well as the original traffic.

Maintaining the RTO: Once an endpoint is communicating with a peer the RTO should MUST adjusted by measuring the RTT and its variance (see equation 2.3 of [RFC6928]). The RTO SHOULD be set based on recent observations [RFC8530].

RTO Expiry: Persistent lack of feedback detected by the RTO (or other means) must be used an indication of potential congestion. A failure to receive any specific response within a RTO interval could potentially be a result of a RTT change, change of path, excessive loss, or even congestion collapse.

If there is no response within the timeout period (often called the RTO interval), TCP collapses the congestion window to one segment [RFC5681]. Other transports must similarly respond when they detect loss of feedback.

RTO expiry require to exponentially increase the size of the timeout interval [RFC8085]. When the retransmission timer expires, the RTO MUST be set to RTO * 2 ("back off the timer") [RFC6298] [RFC8085]. A maximum value MAY be placed on the RTO. This maximum RTO MUST NOT be less than 60 seconds [RFC6298].
4.3. Using Path Capacity

This section describes how a sender needs to regulate the maximum volume of data in flight over the interval of the current RTT, and how it manages transmission of the capacity that it perceives is available.

Congestion Management: The capacity available to a flow could be expressed as the number of bytes in flight, the sending rate or a limit on the number of unacknowledged segments. In steady-state this congestion window reflects a safe limit to the sending rate that has not resulted in persistent congestion. A sender performing congestion management will usually optimise performance for its application by avoiding excessive loss or delay.

One common model views the path between two endpoints as a pipe, new packets enter the pipe at the sender, older one leaves at the receiver. The rate of data that leaves the pipe indicates the share of the capacity utilised by a flow. If on average (over an RTT the sending rate equals the sending rate, it indicates the capacity can safely be used in the next RTT. If the average receiving rate is less than the sending rate, then the path is either queuing packets, the RTT/path has changed, or there is packet loss.

Transient Path: Path capacity information is transient. A sender that fails to use capacity has no understanding whether that capacity remains available to use - or whether it has disappeared (e.g., to a change to a path with a smaller bottleneck, or more traffic has emerged that has consumed the previously available capacity). For this reason, a sender that is limited by the volume of application data available to send MUST NOT continue to grow the congestion window [RFC5681].

Standard TCP states that a TCP sender SHOULD set the congestion window to no more than the Restart Window (RW) before beginning transmission if the TCP sender has not sent data in an interval that exceeds the current retransmission timeout, i.e., when an application becomes idle [RFC5681]. Experimental specifications permit TCP senders to tentatively maintain a congestion window when application-limited, provided that they appropriately and rapidly collapse the window when potential congestion is detected [RFC7661]. This mechanism is called Congestion Window Validation (CWV).

Burst Mitigation: Even in the absence of congestion, statistical multiplexing of flows can result in transient effects for flows sharing common resources. A sender therefore SHOULD avoid
inducing excessive congestion to other flows (collateral damage),
or patterns of loss that result in denying a reasonable access to
the available capacity (sometimes called flow starvation). While
a congestion controller ought to limit sending at the granularity
of the current RTT, this can be insufficient to satisfy the goals
of preventing starvation and mitigating collateral damage. This
requires moderating the burst rate of the sender to avoid
significant periods where a flow(s) consume all buffer capacity at
the path bottleneck, which would otherwise prevent other flows
from gaining a reasonable share.

Endpoints SHOULD provide mechanisms to regulate the bursts of
transmission that the application/protocol sends to the network
(section 3.1.6. of [RFC8085]). ACK-Clocking [RFC5681] can help
mitigate bursts for protocols that receive continuous feedback of
reception (such as TCP). Sender pacing can mitigate this
[RFC8085], (See Section 4.6 of [RFC3449], and has been recommended
for TCP in conditions where ACK-clocking is not effective, (e.g.,
[RFC3742], [RFC7661]). SCTP [RFC4960] defines a maximum burst
length (Max.Burst) with a recommended value of 4 segments to limit
the SCTP burst size.

4.4. Responding to Potential Congestion

Internet flows SHOULD implement appropriate safeguards to avoid
inappropriate impact on other flows that share the resources along a
path. The safety and responsiveness of new proposals need to be
evaluated [RFC5166]. In determining an appropriate congestion
response, designs could take into consideration the size of the
packets that experience congestion [RFC4828].

Congestion Response: An endpoint MUST reduce the rate of
transmission when it detects loss (or some other indicator of
congestion) [RFC2914]. (i.e. a reduction needs to not depend on
reception of a signal from the remote endpoint, considering that
congestion indications could themselves be lost under congestion).

TCP Reno established a method that relies on multiplicative-
decrease to halve the sending rate while congestion is detected.
This response to loss is considered sufficient for safe Internet
operation, but other decrease factors have also been published in
the RFC series [RFC8312].

ECN Response: A congestion control design should provide the
necessary mechanisms to support Explicit Congestion Notification
(ECN) [RFC3168] [RFC5679], as described in section 3.1.7. of
[RFC8085]. This can provide help determine an appropriate
congestion window when supported by routers on the path [RFC7567] to enable rapid early indication of incipient congestion.

The early detection of incipient congestion justifies a different reaction to that for loss [RFC8311] [RFC8087]. Simple feedback of congestion experienced by ECN-marked packets [RFC3168] [RFC8511], relies only on an indication that congestion has been experienced within the last RTT. The reaction for traffic marked with ECT(0) when using this simple feedback of congestion was modified [RFC8511].

Further detail of the ECN marking can be obtained by providing more accurate receiver feedback [ID.-ietf-tcpm-accurate-ecn], enabling a faster reaction reducing the queuing latency [RFC8087]. Current work in progress [ID.ietf-tsvwg-l4s-arch] defines a reaction for packets marked with ECT(1), building on the style of feedback provided by [ID.-ietf-tcpm-accurate-ecn].

Protection from Path Change: Congestion control, like loss recovery, requires timely feedback. Congestion control MUST NOT solely rely on the presence of feedback to perform safely. The only way to surely confirm that a local endpoint has successfully communicated with a remote endpoint is to utilise a timer Section 4.2 to detect a lack of response that could result from a change in the path or the path characteristics. Congestion controllers that are unable to react one (or at most a few RTT) after a congestion indication should observe the guidance in 3.3 of the UDP Guidelines [RFC8085].

Persistent Congestion: Endpoints MUST reduce the rate further below that reflected by the restart window, if the RTO continues to expire.

Persistent congestion can result in congestion collapse and MUST be aggressively avoided [RFC2914]. [RFC8085] provides guidelines for a sender that does not or is unable too adapt the congestion window. A suitable method (e.g., TFRC) continues to reduce the sending rate under persistent congestion, to one packet per round-trip time and then exponentially backs off the time between single packet transmissions if congestion continues to persist [RFC2914].

4.5. Using More Capacity

In the absence of persistent congestion, endpoints are permitted to increase their congestion window and hence their sending rate, providing that there is (or is expected to be) additional data available to send across the path.
TCP Reno [RFC5681] defines an algorithm, known as the AIMD (additive-increase/multiplicative-decrease) that allows a sender to exponentially increase the congestion window each RTT from the initial window to the first detected congestion event. This is designed to allow new flows to rapidly acquire a suitable congestion window. Where the bandwidth delay product (BDP) is large it can take many RTTs to find a suitable share of the path capacity, such paths benefit from methods that more rapidly increase the congestion window (e.g., TCP Cubic [RFC8312]), but need to be designed to also react rapidly to any detected congestion.

Increasing Congestion Window: A sender SHOULD stop increasing its congestion window as soon as it receives indication of congestion and MUST NOT continue to increase its rate for more than a RTT after a congestion indication is received. When increasing the congestion window, a sender can transmit faster than the last known safe rate.

Any increase above the last confirmed rate needs to be regarded as tentative and the sender reduce their rate below the last confirmed safe rate when they experience congestion.

Congestion: An endpoint MUST utilise a method that assures the sender will keep the rate below the previously confirmed safe rate for multiple RTTs after an observed congestion event. In TCP this is performed by using linear increase from a slow start threshold that is re-initialised when congestion is experienced.

Avoiding Overshoot: Overshoot of the congestion window beyond the point of congestion can significantly impact other flows sharing resources along a path. It is important to note that as endpoints experience more paths with a large BDP and a wider range of potential path RTT, that variability or changes in the path can have very significant impacts on appropriate dynamics for increasing the congestion window (see also burst mitigation Section 4.3).

4.6. Network Signals

An endpoint can utilise signals from the network to help determine how to control the traffic it sends.

Network Signals: The assumptions that network devices on path may change motivates the use of soft-state when designing protocols that interact with network devices (e.g., ECN). To protect from changes in the path.
Transport mechanisms need to be robust to potential black-holing of signals, as it must also be robust to loss of packets.

Mechanisms MUST NOT solely rely on ICMP messages or other specific signalling messages to perform safely when these are not received (section 5.2 of [RFC8085]). This can include context-sensitive treatment of "soft" signals provided to the endpoint [RFC5461].

Validation of ICMP: ICMP messages [RFC0792] MUST to be validated before they are used. Other path signals must similarly be validated to protect from malicious use.

4.7. Protection of Protocol Mechanisms

Off Path Attack: A design MUST protect from off-path attack [RFC8085] where an attack on the congestion control can lead to a DoS vulnerability for the flow being controlled and/or other flows that share network resources along the path.

OffOn Path Attack: A protocol can be designed to protect from on-path attacks, but this requires more complexity and the use of encryption/authentication mechanisms (e.g., IPsec [RFC4301], QUIC [I-D.ietf-quic-transport]).

5. IETF guidelines on evaluation of Congestion Control

The IETF has provided guidance [RFC5033] for considering alternate congestion control algorithms. The IRTF has described set of metrics and related trade-off between metrics that can be used to compare, contrast, and evaluate congestion control techniques [RFC5166].

6. Acknowledgements

Nicholas Kuhn helped develop the first draft of these guidelines. Tom Jones reviewed the first version of this draft. Gorry Fairhurst and Tom Jones were funded at the University of Aberdeen by the European Space Agency.

The views expressed are solely those of the author(s).

7. IANA Considerations

This memo includes no request to IANA.

RFC Editor Note: If there are no requirements for IANA, the section will be removed during conversion into an RFC by the RFC Editor.
8. Security Considerations

The security considerations for the use of transports are provided in the references section of the cited RFCs. Security guidance for applications using UDP is provided in the UDP Usage Guidelines [RFC8085].

Section Section 4.6 supports current best practice to validate ICMP messages prior to use. Section Section 4.7 describes general requirements relating to the design of safe protocols and their protection from on and off path attack.

9. References

9.1. Normative References


9.2. Informative References


"TCP Alternative Backoff with ECN (ABE)", RFC 8511,
DOI 10.17487/RFC8511, December 2018,

Appendix A.  Revision Notes

Note to RFC-Editor: please remove this entire section prior to
publication.

Individual draft -00:

  o  Comments and corrections are welcome directly to the authors or
      via the IETF TSVWG, working group mailing list.

  o  This update is proposed for initial WG comments.

  o  If there is interest in progressing this document, the next
      version will include more complete referencing to cited material.

Author’s Address

Godred Fairhurst
University of Aberdeen
School of Engineering
Fraser Noble Building
Aberdeen AB24 3U
UK

Email: gorry@erg.abdn.ac.uk