

TSVWG Working Group
Internet-Draft
Intended status: Experimental
Expires: September 4, 2018

L. Han
Y. Qu
Huawei
T. Nadeau
Lucid Vision
March 3, 2018

A New Congestion Control in Bandwidth Guaranteed Network
draft-han-tsvwg-cc-00

Abstract

In bandwidth guaranteed networks, network resources are reserved before a TCP session starts transmitting data. This draft proposes a new TCP congestion control algorithm used in bandwidth guaranteed networks. It is an extension to the current TCP standards.

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 September 4, 2018.

Copyright Notice

Copyright (c) 2018 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

Internet-Draft

New Congestion Control

March 2018

the Trust Legal Provisions and are provided without warranty as described in the Simplified BSD License.

Table of Contents

1.	Introduction	2
2.	Terminology and Notation	3
3.	Bandwidth Guaranteed Network	4
4.	New Congestion Control	5
4.1.	Receiver Advertised Window Size	5
4.2.	MinBandwidthWND and MaxBandwidthWND	5
4.3.	Congestion Avoidance	6
4.4.	Fast Retransmit and Fast Recovery	7
4.5.	Timeout	8
4.6.	Idle Recovery	8
5.	IANA Considerations	8
6.	Security Considerations	8
7.	References	8
7.1.	Normative References	9
7.2.	Informative References	9
	Acknowledgments	11
	Authors' Addresses	11

[1.](#) Introduction

The original IP protocol suite was designed to support best-effort data transmission. With the development of the Internet, congestion became a real problem. To avoid congestion in the Internet, TCP uses congestion-avoidance algorithms to keep hosts from pumping too much traffic into the network. Over the past 40 years there have been various algorithms and optimizations proposed to solve this problem, including TCP-RENO [[RFC5681](#)], TCP-NewReno [[RFC6582](#)] [[RFC6675](#)], TCP-Cubic [[RFC8312](#)] and BBR [[I-D.cardwell-iccr-g-bbr-congestion-control](#)] etc.

In bandwidth guaranteed networks, network resources are reserved before transmitting data. This draft proposes a new congestion control algorithm that should be used in bandwidth guaranteed networks to improve TCP throughput. The following is a list of key differences between this new algorithm and classic TCP congestion control [[RFC5681](#)]:

It doesn't have a slow start, after a TCP session is successfully

initiated its congestion window (cwnd) jumps to CIR and the host is allowed to transmit data. This is based on the assumption that network resources have been reserved in bandwidth guaranteed networks.

During congestion avoidance, cwnd stays between CIR (Committed Information Rate) and PIR (Peak Information Rate). If there is no packet loss due to congestion, cwnd has a flat top rate as PIR.

OAM is used together with duplicate ACKs to detect whether a packet loss is due to congestion or random failure.

This draft is organized as follows. [Section 2](#) defines terminologies used in this draft. [Section 3](#) provides background information for Bandwidth Guaranteed Networks. [Section 4](#) explains the details of the new congestion control algorithm.

[2](#). Terminology and Notation

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](#)].

Some of the following terms are defined the same as [[RFC5681](#)], and they are copied here for readability.

FULL-SIZED SEGMENT: A segment that contains the maximum number of data bytes permitted (i.e., a segment containing SMSS bytes of data).

RECEIVER WINDOW (rwnd): The most recently advertised receiver window.

CONGESTION WINDOW (cwnd): A TCP state variable that limits the amount of data a TCP can send. At any given time, a TCP MUST NOT send data with a sequence number higher than the sum of the highest acknowledged sequence number and the minimum of cwnd and rwnd.

Sender Maximum Segment Size (SMSS): The SMSS is the size of the largest segment that the sender can transmit. This value can be

based on the maximum transmission unit of the network, the path MTU discovery [RFC1191, [RFC4821](#)] algorithm, RMSS (see next item), or other factors. The size does not include the TCP/IP headers and options.

RECEIVER MAXIMUM SEGMENT SIZE (RMSS): The RMSS is the size of the largest segment the receiver is willing to accept. This is the value specified in the MSS option sent by the receiver during connection startup. Or, if the MSS option is not used, it is 536 bytes [[RFC1122](#)]. The size does not include the TCP/IP headers and options.

INITIAL WINDOW (IW): The initial window is the size of the sender's congestion window after the three-way handshake is completed.

RESTART WINDOW (RW): The restart window is the size of the congestion window after a TCP restarts transmission after an idle period.

ssthresh: Slow Start Threshold.

OAM: Operations, Administrations, and Maintenance.

RTT: Round-Trip Time.

CIR: Committed Information Rate.

PIR: Peak Information Rate.

[3.](#) Bandwidth Guaranteed Network

With the development of new applications, such as AR/VR, the network is required to provide bandwidth guaranteed services. There have been various solutions, including out-of-band signaling protocols such as RSVP [[RFC2205](#)] and NSIS [[RFC4080](#)], and in-band-signaling as proposed in [[I-D.han-6man-in-band-signaling-for-transport-qos](#)]. The common objective of all these solutions is to have network resources/bandwidth reserved before data is transmitted. The details of how the resource is reserved are out of the scope of this draft, however it is assumed that in bandwidth guaranteed networks there have been

network resources (bandwidths, queues etc.) dedicated to the TCP flows, and data is guaranteed at CIR rate. When data rate is between CIR and PIR shared resources are used, and traffic above CIR rate is not guaranteed. No traffic above PIR rate will be allowed to enter the network.

The proposed congestion control also requires that OAM (Operations, administration and management) is used to constantly report on the network condition parameters. Before a TCP session is started, important network parameters need to be detected by OAM, such as number of hops, Round Trip Time (RTT). This might be done through setting up a measuring TCP connection. The measuring TCP connection does not have user data, and it is only used to measure the key network parameters. As the network status is constantly changing, after a TCP session is established, these parameters need to be updated. This requires a sender to periodically or consistently embed TCP data packet with OAM

[[I-D.han-6man-in-band-signaling-for-transport-qos](#)]

[[I-D.ietf-ippm-ioam-data](#)] to detect current buffer depth, RTT etc.

Han, et al.

Expires September 4, 2018

[Page 4]

Internet-Draft

New Congestion Control

March 2018

It is important that OAM needs to be able to detect if any device's buffer depth has exceeded the pre-configured threshold, as this is an indication of potential congestion and packet drop. When this happens, OAM should send a possible congestion alarm to the TCP sender. In case the retransmit timer expires on this TCP sender, if a possible congestion alarm has been received it means a packet is dropped due to congestion. Otherwise it is possible that this packet drop might due to some physical failure. The OAM details are out of the scope of this draft. Please refer to other related drafts.

In summary, in bandwidth guaranteed networks resources are reserved before transmitting data, and OAM is used to get network statistics. The new congestion control proposed in this draft is to be used in this kind of bandwidth guaranteed networks.

[4.](#) New Congestion Control

[RFC5681] defines a set of TCP congestion algorithms: slow start, congestion avoidance, fast retransmit and fast recovery. The proposed congestion control in this draft is an extension to [RFC 5681](#), and it only differs in the congestion control algorithm on the sender side.

[4.1.](#) Receiver Advertised Window Size

Receiver's advertised window (rwnd) is a receiver-side limit on the amount of outstanding data, so a sender should not send data more than this window size. It is calculated as the following:

$$\text{rwnd} = \text{AdvertisedWND} = \text{MaxRcvBuffer} - (\text{LastByteRcvd} - \text{LastByteRead})$$

[4.2.](#) MinBandwidthWND and MaxBandwidthWND

Same as [[RFC5681](#)], on the sender side, the congestion window (cwnd) is the sender-side limit on the amount of data that the sender can transmit before receiving an acknowledgement (ACK). Considering both the sender and the receiver side, the effective sending window is always the minimum of cwnd and rwnd:

$$\text{EffectiveWND} = \min(\text{cwnd}, \text{rwnd})$$

A TCP sender MUST NOT send data more than the minimum of cwnd and rwnd.

Slow-start is commonly used in TCP at the beginning of a transfer or after a loss repair as the network conditions are unknown, hence this slow probing is necessary to determine the available network capacity in order to avoid inappropriately sending large burst of data into

the network and cause congestion. A detailed discussion about initial window setting is provided in [[RFC3390](#)].

RTT is the time taken to send a packet to the destination plus receiving a response packet(ACK). Since the network status is constantly changing, RTT also varies. [[RFC6298](#)] specifies how RTT should be sampled and updated. In this new algorithm RTT is updated using the following formula:

$$\text{RTT} = a * \text{old RTT} + (1-a) * \text{new RTT} \quad (0 < a < 1) \quad (1)$$

The initial RTT can be achieved using a measure TCP connection, or configured based on historical data.

In bandwidth guaranteed network since resources are already allocated

and the network status is known through OAM [[I-D.han-6man-in-band-signaling-for-transport-qos](#)], it is safe to remove slow-start and allow a host to start sending traffic at the rate of CIR after the TCP session is established.

There are two important window sizes, the MinBandwidthWND and the MaxBandwidthWND are calculated as below:

$$\text{MinBandwidthWND} = \text{CIR} * \text{RTT/MSS} \quad (2)$$

$$\text{MaxBandwidthWND} = \text{PIR} * \text{RTT/MSS} \quad (3)$$

In bandwidth guaranteed networks, after a TCP session is established, the sender can start transmitting data at an initial window size, which is equal to MinBandwidthWND:

```

cwnd = MinBandwidthWND
IW = min (cwnd, rwnd)
```

If the receiver window (rwnd) is not a limiting factor, the sender will start sending data at CIR rate. This is a key difference from the classic TCP slow-start, which usually starts from sending one or two packets [[RFC5681](#)].

[4.3.](#) Congestion Avoidance

In TCP-Reno, a TCP enters congestion avoidance mode after slow-start. In bandwidth guaranteed networks, there is no slow-start, so a TCP enters congestion avoidance mode right after the initial start.

During congestion avoidance, for approximately per round-trip time when a valid ACK packet is received, cwnd is increased by one until it reaches MaxBandwidthWND.

```

If (cwnd < MaxBandwidthWND) {
    cwnd +=1;
} else {
    cwnd = MaxBandwidthWND;
}
```

Once the cwnd reaches MaxBandwidthWND , it stays constant at MaxBandwidthWND until packet loss is detected. This is another major

difference from [\[RFC5681\]](#). In [\[RFC5681\]](#) congestion avoidance period, the cwnd keeps increasing until a TCP sender detects segment loss. However, in this new congestion control algorithm, the cwnd stays constant at MaxBandwidthWND until there is packet loss detected.

This means a TCP sender is never allowed to send data at a rate larger than PIR, and it's different from TCP Reno.

4.4. Fast Retransmit and Fast Recovery

Same as defined [\[RFC5681\]](#), a TCP receiver SHOULD send an immediate duplicate ACK when an out-of-order segment arrives. The TCP sender detects and repair loss based on incoming duplicate ACKs. If 3 duplicate ACKs are received, the sender uses it as an indication that a segment has been lost, and will perform a retransmission of the lost segment.

In TCP-Reno [\[RFC5681\]](#), after the fast retransmit of what appears to be the lost segment, fast recovery is used to continue to transmit new segments at a reduced rate ssthresh.

In the new congestion control algorithm, upon receiving duplicate ACKs the fast retransmit and fast recovery follow the below rules:

- o When a sender receives the first and second duplicate ACKs, same as [\[RFC5681\]](#), the cwnd is not changed, and the sender continues to send traffic.
- o When a sender receives the third duplicated ACK, if the retransmission timer has not expired and a previous OAM congestion alarm has been received it is likely a segment is lost due to congestion. The sender will perform a retransmission of the lost segment, and the cwnd is set to be MinBandwidthWND.
- o When a sender receives the third duplicated ACK, but no previous OAM congestion alarm has been received, then it is considered that a segment is lost due to random failure not congestion. In this case the cwnd is not changed.

cwnd is set to be ssthresh, which is usually half of the old cwnd. In this new congestion control, in case there is a segment loss detected as described above, the new cwnd is set to be MinBandwidthWND as in equation (2).

[4.5.](#) Timeout

If a retransmission timer [[RFC6298](#)] in a TCP sender expires, in bandwidth guaranteed networks no matter duplicate ACK received or not, this most likely indicates a physical failure.

In this case, the cwnd is set to be one, and the TCP sender will retransmit the lost segment. This packet also services the function of probing network status. If there is really a network failure, no ACK will be received and the retransmission timer will expire again. Upon receiving an expected ACK after the retransmission, it means the network has recovered, and the cwnd will be set to be MinBandwidthWND as in equation (2).

[4.6.](#) Idle Recovery

It is defined in [[RFC5681](#)] that a TCP session should use slow start to restart transmission after a long idle period more than one retransmission timeout, and the RW (Restart Window) is the minimum of IW and cwnd.

In this proposal, the same rule is still followed. However due to the fact that there is no slow start needed in bandwidth guaranteed networks, and the IW in this new congestion control is set to be MinBandwidthWND, a TCP sender can start transmitting data at CIR rate after a long idle.

[5.](#) IANA Considerations

NA.

[6.](#) Security Considerations

This proposal makes no change to the underlying security of TCP. More information about TCP security concerns can be found in [[RFC5681](#)].

[7.](#) References

7.1. 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>>.

7.2. Informative References

- [RFC2205] Braden, R., Ed., Zhang, L., Berson, S., Herzog, S., and S. Jamin, "Resource ReSerVation Protocol (RSVP) -- Version 1 Functional Specification", [RFC 2205](#), DOI 10.17487/RFC2205, September 1997, <<https://www.rfc-editor.org/info/rfc2205>>.
- [RFC3390] Allman, M., Floyd, S., and C. Partridge, "Increasing TCP's Initial Window", [RFC 3390](#), DOI 10.17487/RFC3390, October 2002, <<https://www.rfc-editor.org/info/rfc3390>>.
- [RFC4080] Hancock, R., Karagiannis, G., Loughney, J., and S. Van den Bosch, "Next Steps in Signaling (NSIS): Framework", [RFC 4080](#), DOI 10.17487/RFC4080, June 2005, <<https://www.rfc-editor.org/info/rfc4080>>.
- [RFC4960] Stewart, R., Ed., "Stream Control Transmission Protocol", [RFC 4960](#), DOI 10.17487/RFC4960, September 2007, <<https://www.rfc-editor.org/info/rfc4960>>.
- [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>>.
- [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>>.
- [RFC6582] Henderson, T., Floyd, S., Gurtov, A., and Y. Nishida, "The NewReno Modification to TCP's Fast Recovery Algorithm", [RFC 6582](#), DOI 10.17487/RFC6582, April 2012, <<https://www.rfc-editor.org/info/rfc6582>>.
- [RFC6675] Blanton, E., Allman, M., Wang, L., Jarvinen, I., Kojo, M., and Y. Nishida, "A Conservative Loss Recovery Algorithm Based on Selective Acknowledgment (SACK) for TCP", [RFC 6675](#), DOI 10.17487/RFC6675, August 2012, <<https://www.rfc-editor.org/info/rfc6675>>.

Internet-Draft

New Congestion Control

March 2018

- [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>>.
- [I-D.cardwell-iccr-g-bbr-congestion-control]
Cardwell, N., Cheng, Y., Yeganeh, S., and V. Jacobson, "BBR Congestion Control", [draft-cardwell-iccr-g-bbr-congestion-control-00](#) (work in progress), July 2017.
- [I-D.han-6man-in-band-signaling-for-transport-qos]
Han, L., Li, G., Tu, B., Xuefei, T., Li, F., Li, R., Tantsura, J., and K. Smith, "IPv6 in-band signaling for the support of transport with QoS", [draft-han-6man-in-band-signaling-for-transport-qos-00](#) (work in progress), October 2017.
- [I-D.ietf-ippm-ioam-data]
Brockners, F., Bhandari, S., Pignataro, C., Gredler, H., Leddy, J., Youell, S., Mizrahi, T., Mozes, D., Lapukhov, P., Chang, R., and d. daniel.bernier@bell.ca, "Data Fields for In-situ OAM", [draft-ietf-ippm-ioam-data-01](#) (work in progress), October 2017.

Internet-Draft

New Congestion Control

March 2018

Acknowledgments

The authors wish to thank xxxx for their helpful comments and suggestions.

Authors' Addresses

Lin Han
Huawei
2330 Central Expressway
Santa Clara CA 95050
USA

EMail: lin.han@huawei.com

Yingzhen Qu
Huawei
2330 Central Expressway
Santa Clara CA 95050
USA

EMail: yingzhen.qu@huawei.com

Thomas Nadeau
Lucid Vision
Hampton NH 03842
USA

EMail: tnadeau@lucidvision.com

