ccwg
Internet-Draft
Intended status: Standards Track
Expires: 29 August 2025

G. Zhao
Z. Du
China Mobile

25 February 2025

Improvement of Congestion Control Methods Based on Bandwidth Measurement
draft-zhao-ccwg-bcmi-00

Abstract

This document discusses how the Congestion Control algorithm integrates and utilizes the bandwidth, rate recommendations, and constraints etc. provided by bandwith measurement to achieve better congestion control.This document discusses how the Congestion Control algorithm.

Table of Contents

1. Introduction 2
1.1. Terminology 3
1.2. Requirements Language 3
2. Overview 3
3. Bandwidth measurement methods 3
3.1. Available measurement through hop-by-hop 3
3.2. Throughput advice by network elements 4
3.3. Throughput advice from control or management plane . . . 4
4. 4.Example: RENO-type congestion control algorithms with Bandwidth Measurement 5
4.1. Slow start phase 5
4.2. Congestion avoidance phase 6
4.3. Fast Recovery Phase 6
4.4. ECN Message Handling 6
5. BBR with Bandwidth Measurement 6
6. IANA Considerations 7
7. Security Considerations 7
8. Contributors 7
9. Acknowledgements 7
10. References 7
10.1. Normative References 7
10.2. Informative References 7
Authors' Addresses 8

1. Introduction

The congestion control algorithm of TCP[RFC5681] adjusts the size of the congestion window dynamically to control the sending rate, in order to adapt to different network environments and congestion conditions. RENO-type congestion control algorithms control packet sending rates based on received ACK packets, adjust congestion windows to control sending rates, and use lost packets as congestion control signals to perform network congestion control, featuring slow start, congestion avoidance, fast retransmit and fast recovery. CUBIC is a typical RENO-type congestion control algorithm, currently the default congestion control algorithm in Linux, Windows, and other operating systems. It uses a cubic function as the congestion window growth function during the congestion avoidance phase to improve network bandwidth utilization.

The BBR congestion control algorithm mainly adjusts the size of the congestion window by periodically probing the bottleneck bandwidth (bandwidth and delay) of the link, thereby achieving higher bandwidth utilization and lower transmission latency. BBR[I-D.ietf-ccwg-bbr] consists of four stages: Startup, Drain, Probe Bandwidth, and Probe RTT. The latest iteration of BBR has refined the Probe Bandwidth

phase by incorporating new stages including cruise, refill, up, and down. These enhancements aim to improve the fairness of BBR flows when sharing the network with other traffics, while also adding capabilities such as improved packet loss management.

The available bandwidth or throughput advice of the network link cloud be used at congestion control algorithms' phases to improve the effect of congestion control, such as slow start, congestion avoidance and quick recovery etc. These methods can quickly and accurately obtain the available bandwidth of the link and adjust the data sending rate according to the available bandwidth or throughput advice, achieving rapid convergence of sending rates, avoiding network congestion, and making full use of network bandwidth.

1.1. Terminology

* ABW: available bandwidth of link * CC: congestion control * RTT: round-trip time * CWND: congestion window size.

1.2. Requirements Language

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.

2. Overview

The ABW(available bandwidth) of links can be applied in existing CC algorithms to optimize their throughput performance, such as TCP Reno and CUBIC. The sending rate and congestion window can be dynamically adjusted during the CC's slow-start and loss recovery phases. The BBR algorithm, which detects link bottleneck bandwidth based on rate and round-trip time (RTT), can utilize the ABW to obtain the bottleneck bandwidth of the link and optimize data throughput efficiency. Alternatively, a completely new CC algorithm can be designed based on ABW to predict and avoid congestion in advance.

3. Bandwidth measurement methods

3.1. Available measurement through hop-by-hop

The daft [draft-shi-ippm-congestion-measurement-data-02] specifies a method to measure available bandwidth. To obtain the available bandwidth of network links by traversing the minimum available bandwidth of sending nodes, transit nodes, and receiving nodes.

3.2. Throughput advice by network elements

The scone WG aims to establish a mechanism for network elements capable of rate-limiting a UDP 4-tuple to communicate an upper bound on achievable bitrate, termed "throughput advice". The throughput advice serves as a guideline to enhance user experience and represents the maximum bitrate manageable by a single network element for that user’s current connection. This mechanism will allow an application to receive notifications containing throughput advice for both upstream and downstream traffic from any network elements. Currently, there are mainly three methods: a.TRAIN[draft-thomson-scone-train-protocol-00] b.NRLPs[draft-brw-scone-rate-policy-discovery-02] c.Throughput Advice[draft-brw-scone-throughput-advice-blob-02]

3.3. Throughput advice from control or management plane

If the network providing the transport service and the peer nodes (i.e., the sender and the receiver) are all under the control of the same entity, for example, a telecom operator that owns both the cloud infrastructure and the network, the control plane or the management plane of the entity can provide a throughput advice to the APPs.

In this scenario, the administrator should be aware of the available bandwidth of the network and the requirement of the flows of the APPs. For example, in the night, the network will be light-loaded, and the administrator can configure part of the bandwidth for a group of flows that need a high throughput. Each flow can be allocated a certain number of bandwidth. In other words, the APP on the sender can obtain a suggested sending rate.

In this scenario, the administrator should be aware of the available bandwidth of the network and the requirement of the flows of the APPs. For example, in the night, the network will be light-loaded, and the administrator can configure part of the bandwidth for a group of flows that need a high throughput. Each flow can be allocated a certain number of bandwidth. In other words, the APP on the sender can obtain a suggested sending rate.Thus, the APP could continue sending traffic at that rate. It is the responsibility of the administrator that the network should have enough bandwidth for it. Additionally, the APP can also send traffic at a higher rate if the CC algorithm finds that a larger rate is available.

The sender should be able to communicate with a specific node that be aware of the available resource information, such as a control node in the control plane. A general procedure for the mechanism is described as below.

Firstly, the sender containing the APP would send a request to the control node. The request should contain the ID information of the source node (i.e., the sender) and the destination node (i.e., the receiver), and an expected sending rate.

Secondly, after receiving the request, the control node responds a suggested rate to the source node. Thus, after receiving the response, the sender can send traffic at this suggested rate. Optionally, the control node can update the rate.

Thirdly, the sender can release the resource after the transmission is completed.

4. 4.Example: RENO-type congestion control algorithms with Bandwidth Measurement

The RENO-type congestion control algorithms cloud be enhanced to leverage available bandwidth measurement mainly includes these three parts: slow start phase, congestion avoidance phase and fast recovery phase. The available bandwidth of the link is obtained hop-by-hop with the data packet.

4.1. Slow start phase

During the slow start phase of congestion control, the congestion window can continue to grow exponentially. When determining whether to exit the slow start phase, it is possible to base this decision on the available bandwidth of the current link. If the size of the congestion window for the next iteration is greater than or equal to CWNDtarget, then the slow start phase is terminated. This approach helps to avoid buffer overflow issues that might occur by using packet loss signals as the trigger for exiting the slow start phase.

CWNDtarget = 2 * ABW * RTT

Certainly, one could directly bypass the slow start phase and set the congestion window size equal to the available bandwidth, allowing the flow to quickly reach its reasonable sending rate. However, this approach may compromise the fairness of other traffic flows in the network.

4.2. Congestion avoidance phase

During the congestion avoidance phase, after receiving data with the current available bandwidth of the link, the difference between the actual sending rate and the available bandwidth is compared, and different strategies are adopted to adjust the next congestion window size based on the difference. The available bandwidth of a link can be periodically probed based on the size of the RTT (Round-Trip Time).

When receiving a data packet with the current available bandwidth of the link, such as an ACK data packet with the current link size, parse the current available bandwidth of the link. Then, compare the actual CWND with the CWNDtarget. If the difference is within a certain range, the next CWND size is eaqual to CWNDtarget, or it approaches CWNDtarget using a method of linear increase or decrease. If the difference exceeds a certain range, it could approache CWNDtarget using exponential change.

4.3. Fast Recovery Phase

When packet loss occurs on the link, decrease the congestion window size based on the packet loss rate threshold, and continue for a period of time, for example, reduce 0.5 of the current congestion window size, lasting for a specific period of time, and then enter the fast recovery phase. During fast recovery CWND, we can either directly set the congestion window to the current CWNDtarget calculated based on available bandwidth or use the following methods.

CWNDnext=(CWNDcurr+CWNDtarget)/2

CWNDnext represents the next congestion window size, CWNDcurr represents the current congestion window size, i.e. After five iterations, the size of the congestion window becomes close to the value of CWNDtarget, and the size for the next CWND iteration will be equal to CWNDtarget.

4.4. ECN Message Handling

TBD.

5. BBR with Bandwidth Measurement

TBD.

6. IANA Considerations

TBD.

7. Security Considerations

TBD.

8. Contributors

The following people have substantially contributed to this document:

Zhiqiang Li
lizhiqiangyjy@chinamobile.com

Hongwei Yang
yanghongwei@chinamoblie.com

Kehan Yao
yaokehan@chinamobile.com

9. Acknowledgements

TBD.

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>.

[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>.

10.2. Informative References

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

[I-D.ietf-ccwg-bbr]
Cardwell, N., Swett, I., and J. Beshay, "BBR Congestion Control", Work in Progress, Internet-Draft, draft-ietf-ccwg-bbr-01, 21 October 2024, <https://datatracker.ietf.org/doc/html/draft-ietf-ccwg-bbr-01>.

Authors' Addresses

Guangyu Zhao
China Mobile
No.32 XuanWuMen West Street
Beijing
100053
China
Email: zhaoguangyu@chinamobile.com

Zongpeng Du
China Mobile
No.32 XuanWuMen West Street
Beijing
100053
China
Email: duzongpeng@chinamobile.com