Workgroup: Network Working Group
Internet-Draft: draft-bagnulo-congress-cci-01
Published: 6 August 2023
Intended Status: Informational
Expires: 7 February 2024
Authors: M. Bagnulo
UC3M

Congestion Control Invariants

Abstract

This document initiates the discussion about Congestion Control Invariants, that is, mechanisms that several CCAs implement and that would benefit from a common specification for all CCAs to improve their interoperability

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 7 February 2024.

Copyright Notice

Copyright (c) 2023 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 Revised BSD License text as described in Section 4.e of the Trust Legal Provisions and are provided without warranty as described in the Revised BSD License.

Table of Contents

1. Introduction

Over the last decade, we have witnessed a refreshing spring in congestion control research, resulting in a number of novel congestion control algorithms (CCAs). Indeed, in addition to the traditional congestion control algorithms such as New Reno and Cubic, we can now observe in that at least, the following algorithms are being used in parts of the Internet:

BBR (Bottleneck Bandwidth and Round-trip propagation time) [I-D.cardwell-iccrg-bbr-congestion-control] is a model-based congestion control algorithm that attempts to improve the performance of Internet communications by reducing the delay (when bottleneck buffers are large) and increase the throughput (when bottleneck buffers are small).

LEDBAT/LEDBAT++ (Low Extra Delay Background Transport)) [I-D.irtf-iccrg-ledbat-plus-plus] is a CCA that implements a less-than-best-effort (LBE) traffic class. When LEDBAT()++) traffic shares a bottleneck with one or more TCP connections using Cubic or other loss-based congestion control algorithms, it reduces its sending rate earlier and more aggressively than competing flows, allowing Cubic traffic to use more of the available capacity.

DCTCP (Data-Center TCP) [I-D.ietf-tcpm-dctcp] is a CCA developed by Microsoft to reduce the latency for data center communications. DCTCP relies on AccECN to quantify the amount of inflight traffic that is experiencing congestion and reduced the sending rate accordingly. This allows DCTCP to operate with small queues, oscillating around the optimal operation point. while DCTP was originally designed for its use within data center networks, the L4S (Low Latency, Low Loss, and Scalable Throughput) architecture extends the use of DCTCP to the Internet.

MPTCP (MultiPath TCP) [RFC8684] is an extension to TCP to support multiple concurrent paths in a single TCP connection. MPTCP includes a novel CCA that allows the coupling of the CCAs used in the different paths [RFC6356]. Through the coupled CCA, MPTCP manages to offload traffic from paths that are experiencing congestion towards path that are less congested.

The adoption of the aforementioned CCA has not been uneventful. The roll-outs of some CCA have been problematic [10.1145 3355369.3355604] than others. Specifically, the wide deployment of BBR(v1) attracted a fair amount of attention due to the (un)fairness issues that arise when BBR(v1) competes against legacy CCAs such as Cubic and New Reno . As it has been repeatedly reported, BBR(v1) does not react to packet losses, which results in large packet loss rate for itself and other competing flows using alternative CCAs. Since other CCAs (such as Cubic) do react to packet losses, this BBR(v1) behaviour resulted in BBR(v1) seizing more than its fair share of capacity when competing with CCAs that do react against packet losses. these fairness issues are now being corrected with the new version of BBR (BBRv2) and also triggered the community to re-think the fairness requirements imposed to novel CCAs in order to be deployed in the public Internet.

In this note, we focus in a different aspect of the interaction between different CCAs. Specifically, we posit that several of these CCAs implement similar functionalities in different ways which pose challenges to the correct interaction between these CCAs. The goal of this note is to initiate a line of research to identify potential invariants in CCAs, meaning, mechanisms that several CCAs implement and that would benefit from a common specification for all CCAs to improve their interoperability. Such standardised mechanisms could serve as building blocks for novel CCAs, so that when a new CCA needs to implement one of such functions, it re-uses the specified building block, rather than re-inventing it. To bootstrap the proposed work, we motivate and propose a first Congestion Control algorithm Invariant (CC), namely, periodic slow downs.

1.1. Periodic Slow Down Invariant

1.1.1. Motivation

Both BBR and LEDBAT++ estimate the base RTT as part of their operations. The base RTT is the RTT in the absence of queueing delay, which means it is the minimum RTT observable in a given path. LEDBAT++ uses the base RTT to determine the current queuing delay, which is computed as the difference between the current RTT and the base RTT. BBR uses the base RTT to determine the Bandwidth Delay Product (BDP) which affects the flight-size a flow is able to inject in the network.

In order to have visibility of the base RTT, both protocols perform periodic slow downs as an attempt to empty the queues and expose the base RTT. Because there may be multiple flows contributing to the queue, both protocols include some form of synchronisation logic, that allows multiple competing flows to slow down at the same time, increasing the chances to empty the queue and expose the base RTT. While both protocols implement the periodic slow down, the actual implementation details differ.

In the case of LEDBAT++, it performs a slow-start increase at the beginning of the connection. Then, LEDBAT++ executes periodic slow-downs to obtain more accurate measurements of the base RTT. Specifically LEDBAT++ sets the Congestion Window (CW) to 2 MSS during 2 RTTs and then performs a slow-start increase back to the value that it was using before the periodic decrease. An initial slow-down is performed 2 RTTs after exiting the initial slow-start. This process is performed periodically. If we call Tss the time that it takes for the slow-start to ramp back up, then LEDBAT++ performs the next periodic slow down after a period equal to 9Tss.

This mechanism effectively empties the queue when there is a single LEDBAT++ flow contributing to the queue (i.e. there is no other traffic, LEDBAT++ or otherwise). If there are other competing LEDBAT++ flows, this mechanism, albeit counter-intuitively, actually works. Where there is a single flow int he bottleneck and it is using LEDBAT++, it will correctly estimate the base RTT. If later on, another LEDBAT++ joins, the base RTT measured will include the added queueing delay T generated by the previous flow. This will trigger than the second flow will attempt to generate an additional queueing delay T on top of that, outcasting the first flow. This is called late-comer advantage and has been documented extensively [_10.1145_3355369.3355604]. At this point, only the second flow prevails. This is when the initial slow down of the second flow kicks in. Since the second flow has outcasted the first flow, when the second flow slows down, it exposes the base RTT.

In the base of BBRv1, if during the last 10s, a BBRv1 flow has not observed an RTT smaller than its current estimation of the base RTT (called RTprop), BBRv1 enters in the ProbeRTT state, reducing the inflight to only 4 packets during at least 200 ms and one RTT. RTprop is set to the minimum RTT observed during the last 10 s. This mechanism naturally embeds synchronisation of slow-downs across multiple flows. Suppose there are N uncoordinated BBRv1 flows competing in the bottleneck. When the first one of them performs a slow down, it is likely that the rest of the flows record a minimum value for the RTT, which would likely cause than the next slow down will occurs 10 s after this for all flows.

We have described how both LEDBAT++ and BBRv1 periodic slow down mechanism work when there are multiple LEDBAT++/BBRv1 flows respectively. We next consider how the slow down mechanism perform when there is a mix of BBRv1 and LEDBAT++ flows. Based on the logic of each of the mechanisms, we can easily conclude that will not synchronise their slow downs. The reason for this is that the period of the slowdowns does not match. In the case of BBR is a fixed period of 10 s, while in the LEDBAT++ case, the period depends both on the RTT and in the targeted CW. This lack of synchronisation has been verified experimentally in [COMNET].

1.1.2. Proposed invariant

Having two CCAs such as LEDBAT++ and BBR implementing two different slow down mechanisms is clearly counterproductive, since neither of them is able to perform concurrently and expose the base RTT when there is a mix of both types of flows competing in a bottleneck. Having a single slow down mechanism standardised that should be used as a building block by every CCA that requires a periodic slow down mechanism would naturally bring interoperability between the different CCAs, avoiding interference when they need to expose and measure the base RTT.

Regarding the specific mechanism, we believe that the one specified by BBR has merits over the one of LEDBAT++. Specifically, the one specified by BBR is able to synchronise the slowdowns of multiple flows, which seems challenging for the LEDBAT++ mechanism, especially when the different flows have different characteristics. for instance, if there are different LEDBAT++ flows with different RTTs competing in the same bottleneck, the periods of the slow downs of the different flows is likely to be different as the Tss for each flow will be different (because the RTTs are different).

1.2. Other potential invariants

As next steps, we propose to identify other potential invariants by identifying basic building blocks used in different CCAs and that if implemented in different ways would result in interference between the different flavours.

2. Security Considerations

3. IANA Considerations

4. Acknowledgements

This work was supported by the EU through the StandICT CCI project.

5. Informative References

[COMNET]
Bagnulo, M.B. and A.G. Garcia-Martinez, "When less is more: BBR versus LEDBAT++", , Computer Networks Volume 219, 2022.

[I-D.cardwell-iccrg-bbr-congestion-control]
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>.

[I-D.ietf-tcpm-dctcp] Bensley, S., Thaler, D., Balasubramanian, P., Eggert, L., and G. Judd, "Data Center TCP (DCTCP): TCP Congestion Control for Data Centers", Work in Progress, Internet-Draft, draft-ietf-tcpm-dctcp-10, 28 August 2017, <https://datatracker.ietf.org/doc/html/draft-ietf-tcpm-dctcp-10>.

[I-D.irtf-iccrg-ledbat-plus-plus] Balasubramanian, P., Ertugay, O., and D. Havey, "LEDBAT++: Congestion Control for Background Traffic", Work in Progress, Internet-Draft, draft-irtf-iccrg-ledbat-plus-plus-01, 25 August 2020, <https://datatracker.ietf.org/doc/html/draft-irtf-iccrg-ledbat-plus-plus-01>.

[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>.

[RFC6817] Shalunov, S., Hazel, G., Iyengar, J., and M. Kuehlewind, "Low Extra Delay Background Transport (LEDBAT)", RFC 6817, DOI 10.17487/RFC6817, December 2012, <https://www.rfc-editor.org/info/rfc6817>.

[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>.

[_10.1016_j.comnet.2013.02.020]
Carofiglio, G., Muscariello, L., Rossi, D., Testa, C., Valenti, S., and Elsevier BV, "Rethinking the Low Extra Delay Background Transport (LEDBAT) Protocol", Computer Networks, vol. 57, no. 8, pp. 1838-1852, DOI 10.1016/

j.comnet.2013.02.020, June 2013, <http://dx.doi.org/10.1016/j.comnet.2013.02.020>.

[_10.1145_3355369.3355604] Ware, R., Mukerjee, M. K., Seshan, S., Sherry, J., and ACM, "Modeling BBR's Interactions with Loss-Based Congestion Control", Proceedings of the Internet Measurement Conference, DOI 10.1145/3355369.3355604, 21 October 2019, <http://dx.doi.org/10.1145/3355369.3355604>.

Author's Address

Marcelo Bagnulo
UC3M

Email: marcelo@it.uc3m.es