BBRv3 in the public Internet: a boon or a bane?

Danesh Zeynali
Max-Planck-Institut für Informatik

Emilia N. Weyulu
Max-Planck-Institut für Informatik

Seifeddine Fathalli
Max-Planck-Institut für Informatik

Balakrishnan Chandrasekaran
Vrije Universiteit Amsterdam

Anja Feldmann
Max-Planck-Institut für Informatik

Abstract

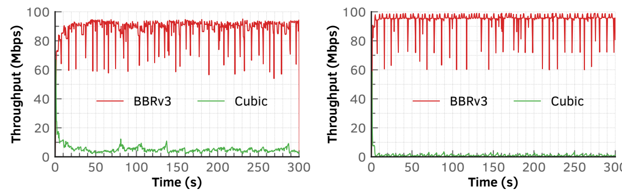
The third version of the Bottleneck-Bandwidth and Round-trip (BBR) algorithm, BBRv3, is now the default CCA for all of the public Internet traffic from google.com and YouTube. In this work, we built upon our prior work [23] and examine BBRv3's ability to coexist with Cubic flows by taking loss, in the form of explicit congestion notification (ECN) signals, into account. Our evaluations reveal that, when ECN is enabled, a single BBRv3 flow can acquire more than ~99% of the bandwidth even when competing with five Cubic flows. Our findings have crucial implications for using BBRv3 in the public Internet.

ACM Reference Format:
Danesh Zeynali, Emilia N. Weyulu, Seifeddine Fathalli, Balakrishnan Chandrasekaran, and Anja Feldmann. 2024. BBRv3 in the public Internet: a boon or a bane?. In *Applied Networking Research Workshop (ANRW 24), July 23, 2024, Vancouver, AA, Canada.* ACM, New York, NY, USA, 3 pages. https://doi.org/10.1145/3673422.3674889

1 BBR in the Public Internet

The Bottleneck-Bandwidth and Round-trip (BBR) congestion control algorithm (CCA) models the network path between a sender and receiver by periodically estimating the bottleneck bandwidth and round-trip time (RTT) [4]. It then uses that model to maximize the sender's throughput—by matching its rate to the measured bandwidth—and minimize delay as well as loss—by matching the in-flight volume to the bandwidth-delay product (BDP). In practice, the CCA essentially converges with a high probability to Kleinrock's optimal operating point [12]. Today, BBR is the CCA for all public Internet traffic served from google.com and YouTube [3].

The first version of BBR eschewed the use of loss as a signal to rate control the sender, regardless of whether it was in the startup phase or steady state. Several studies revealed that this design was aggressive in competition with loss-based CCAs and highly unfair to them in shallow buffer scenarios [1, 8, 10, 18, 22]. In 2019, the second version, BBRv2, took loss into account to share bandwidth equitably with widely used CCAs such as Cubic and NewReno [5, 6]. Independent evaluations showed, however, that BBRv2 suffered from low link utilization and was overwhelmed by loss-based CCAs in deep buffer scenarios [11, 15, 19]. BBRv3, the most recent version, shipped in July 2023 with optimizations to improve its bandwidth convergence and fairness concerns [2]. We simply ask *whether these optimizations indeed facilitate an equitable sharing of bandwidth with the widely used CCAs in the Internet such as Cubic.*



(a) ***ECN disabled*** **(b)** ***ECN enabled***

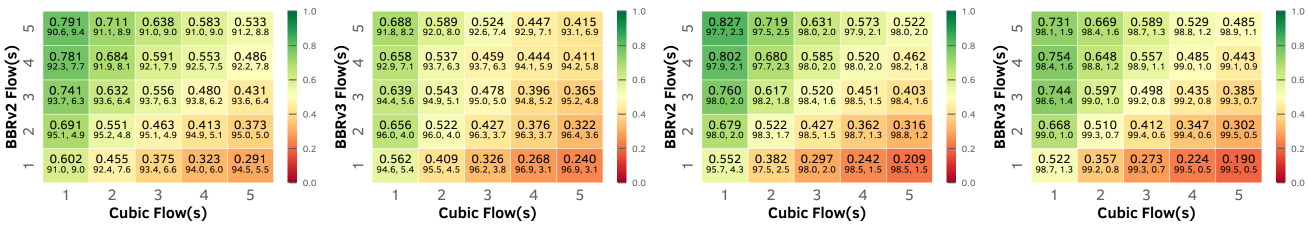
Figure 1: *BBRv3 is quite unfair to Cubic, and, with ECN enabled, the BBRv3 flow quickly chokes the Cubic flow.*

One of the fundamental changes in BBRv2, which has remained intact in BBRv3, is the use of explicit congestion notification (ECN) signals for adapting the sender's delivery rate. ECN allows routers along the network path between a sender and receiver to *mark* a packet to signal impending congestion. To support ECN a router must use an active queue management (AQM) policy. An AQM policy enables the router to mark packets in its buffer—by setting the ECN field in their IP headers to the *Congestion Experienced* (CE) codepoint—for signaling an imminent packet drop and reduce the likelihood of that buffer becoming full. For this CE codepoint to help the sender, the receiver must echo the signal to the sender, via the *ECN-Echo* (ECE) field in the TCP header. An ECN-aware sender, hence, tests ECN support at the receiver during TCP connection establishment, and the transport is deemed ECN-capable only when the endpoints can successfully negotiate support. Per Google, BBRv3's support for ECN, along with a few performance optimizations, enables it to be fair towards loss-based CCAs [2, 3].

BBR's ability to coexist fairly alongside widely used CCAs such as Cubic has huge implications for its deployment in

ANRW 24, July 23, 2024, Vancouver, AA, Canada

ACM ISBN 979-8-4007-0723-0/24/07
https://doi.org/10.1145/3673422.3674889



(a) ***BBRv2 w/o ECN*** **(b)** ***BBRv3 w/o ECN*** **(c)** ***BBRv2 w/ ECN*** **(d)** ***BBRv3 w/ ECN***

Figure 2: *JFI and average bandwidth values for BBRv2 and BBRv3 w/ and w/o ECN. Neither versions share bandwidth fairly with Cubic, albeit Cubic flows get a higher bandwidth w/o ECN than w/ ECN.*

the public Internet. Our preliminary work[1] showed, however, that BBRv3 is quite unfair to Cubic in various network scenarios, albeit we did not test with ECN support [23]. With widely used OSes supporting ECN, several measurement studies show widespread ECN adoption in the Internet [13, 14, 20, 21]. In this paper, we, hence, investigate whether BBRv3 can share bandwidth equitably with Cubic (the default CCA in the Linux Kernel, and one of the most widely used loss-based CCAs in the Internet) when the endpoints as well as routers along the path use ECN for mitigating congestion. This study presents a first cut towards an extensive evaluation of BBRv3 with ECN as an examination of its suitability for deployment in the public Internet.

2 Approach

We set up a testbed with a dumbbell topology [23]. We used two senders (BBRv2 and BBRv3 on Linux kernels v5.13.12 and v6.4.0, resp.) and one receiver; each host had 4 cores and 8 GiB of RAM and they were connected via 25 Gbps links. We used `tc` [9] to (a) configure a bottleneck bandwidth of 100 Mbps on the path, (b) introduce a round-trip delay of 100 ms between the endpoints, and (c) implement RED [7] or CoDel AQMs [16], as required, at the bottleneck. For CoDel, we used a queue of size 1-BDP and marked the packets after 24.2 ms of sojourn time. These settings are based on those used by Google [6], but scaled appropriately for our testbed. In Google's experiments, for instance, the marking threshold corresponds to the serialization delay for 20 packets on a 1 Gbps link with 1 ms RTT; it is essentially equal to 25% of their queue size [6]. Unless otherwise mentioned, we used DCTCP-style ECN marking recommended by BBR. We enabled TCP segmentation offloading (TSO) and large receive offload (LRO) at the end hosts, but disabled them on the bottleneck to limit the queue size to fixed packet sizes.

3 Is BBR fair to Cubic?

We start with a simple throughput test where we pit a BBRv3 flow against a Cubic flow. Per Fig. 1, the BBR flow does not leave any bandwidth for the Cubic flow, and enabling ECN, surprisingly, only exacerbates this unfairness. Instead of DCTCP-style ECN marking, if we opt for RFC-3168 [17] style (using RED queues), the BBR (v3 and v2) sender stalls indefinitely. We omit this result due to space constraints.

To determine BBRv3's fairness, we vary the number of BBR and Cubic flows (between 1 and 5) contending for bandwidth (for 300 s) and compute the Jain's fairness index (JFI). We repeat this trial thrice and compute the average of the JFI values (Fig. 2); the figure also shows the aggregate share of BBR and Cubic flows, below the averages. BBR does not share bandwidth equitably with Cubic regardless of whether ECN is used. Disabling ECN alleviates the unfairness, but only marginally. Even when five Cubic flows compete with a single BBR flow, the Cubic flows hardly receive any bandwidth from the BBR flow; the Cubic flows, unfortunately, only end up competing between themselves for the bandwidth leftover by the BBR flow. BBR's unfairness towards Cubic remains consistent when we scale up the experiment by increasing the counts of both BBR and Cubic flows in lockstep. If we repeat the experiments using BBRv2 instead of BBRv3, BBRv2 fares slightly better than BBRv3 in coexisting with Cubic; BBRv3's performance optimizations, unfortunately, seem to have eroded the progress made by BBRv2 on improving the fairness towards loss-based CCAs.

4 Concluding remarks

The BBR CCA has undergone three revisions in the past seven years to improve its performance and its ability to coexist with widely used CCAs, such as Cubic. Our evaluations show, however, that the most recent version is still highly unfair to Cubic, raising concerns about its use in the public Internet. Redesigning BBR to be fair to loss-based CCAs while maintaining high throughput and low delay presents a significant challenge; we leave it to future work. Our study, additionally, highlights an key challenge in evaluating CCAs: lack of a clear consensus on CCA evaluation and/or benchmarking. We believe that it is a key source of the discrepancies between our findings and those of Google concerning BBR's behavior. We hope that this work informs the discussion on standardizing CCA evaluations.

[1] https://inet-bbrv3eval.mpi-inf.mpg.de/

References

[1] Yi Cao, Arpit Jain, Kriti Sharma, Aruna Balasubramanian, and Anshul Gandhi. 2019. When to Use and When Not to Use BBR: An Empirical Analysis and Evaluation Study. In *ACM Internet Measurement Conference (IMC)*.

[2] Neal Cardwell, Yuchung Cheng, C. Stephen Gunn, Steven Yang, David Morley, Soheil Hassas Yeganeh, Ian Swett, Bin Wu, Victor Vasiliev, Priyaranjan Jha, Yousuk Seung, and Van Jacobson. 2023. BBRv3: Algorithm Bug Fixes and Public Internet Deployment. https://datatracker.ietf.org/meeting/117/materials/slides-117-ccwg-bbrv3-algorithm-bug-fixes-and-public-internet-deployment-00. [IETF 117; CCWG].

[3] Neal Cardwell, Yuchung Cheng, C. Stephen Gunn, Steven Yang, David Morley, Soheil Hassas Yeganeh, Ian Swett, Bin Wu, Victor Vasiliev, Priyaranjan Jha, Yousuk Seung, and Van Jacobson. 2024. BBRv3: Algorithm Overview and Google's Public Internet Deployment. https://datatracker.ietf.org/meeting/119/materials/slides-119-ccwg-bbrv3-overview-and-google-deployment-00. [IETF 119; CCWG].

[4] Neal Cardwell, Yuchung Cheng, C. Stephen Gunn, Soheil Hassas Yeganeh, and Van Jacobson. 2016. BBR: Congestion-Based Congestion Control. *Queue* 14, 5 (Oct. 2016).

[5] Neal Cardwell, Yuchung Cheng, Soheil Hassas Yeganeh, Ian Swett, and Van Jacobson. 2022. *BBR Congestion Control*. Internet-Draft draft-cardwell-iccrg-bbr-congestion-control-02. Internet Engineering Task Force (IETF). https://datatracker.ietf.org/doc/draft-cardwell-iccrg-bbr-congestion-control/02/

[6] Neal Cardwell, Yuchung Cheng, Soheil Hassas Yeganeh, Ian Swett, Victor Vasiliev, Matt Mathis Bin Wu, Priyaranjan Jha, Yousuk Seung, and Van Jacobson. 2019. *BBR v2: A Model-based Congestion Control IETF 105 Update*. Technical Report. [IETF 105; ICCRG].

[7] Sally Floyd and Van Jacobson. 1993. Random Early Detection Gateways for Congestion Avoidance. *IEEE/ACM Trans. Netw.* 1, 4 (Aug. 1993).

[8] Mario Hock, Roland Bless, and Martina Zitterbart. 2017. Experimental evaluation of BBR congestion control. In *IEEE International Conference on Network Protocols (ICNP)*.

[9] Bert Hubert. 2021. tc(8) — Linux manual page. https://man7.org/linux/man-pages/man8/tc.8.html. [Last accessed: April 20, 2024].

[10] Geoff Huston. 2017. BBR TCP. https://labs.ripe.net/author/gih/bbr-tcp/. [Last accessed: April 19, 2024].

[11] Alexey Ivanov. 2020. Evaluating BBRv2 on the Dropbox Edge Network. In *Netdev, The Technical Conference on Linux Networking*.

[12] Leonard Kleinrock. 1979. Power and deterministic rules of thumb for probabilistic problems in computer communications. In *IEEE International Conference on Communications*.

[13] Mirja Kühlewind, Michael Walter, Iain R. Learmonth, and Brian Trammell. 2018. Tracing Internet Path Transparency. In *Network Traffic Measurement and Analysis Conference (TMA)*.

[14] Hyoyoung Lim, Seonwoo Kim, Jackson Sippe, Junseon Kim, Greg White, Chul-Ho Lee, Eric Wustrow, Kyunghan Lee, Dirk Grunwald, and Sangtae Ha. 2022. A Fresh Look at ECN Traversal in the Wild. *ArXiv* (2022).

[15] Aarti Nandagiri, Mohit P. Tahiliani, Vishal Misra, and K. K. Ramakrishnan. 2020. BBRvl vs BBRv2: Examining Performance Differences through Experimental Evaluation. *IEEE International Symposium on Local and Metropolitan Area Networks (LANMAN)* (2020).

[16] Kathleen Nichols and Van Jacobson. 2012. Controlling queue delay. *Commun. ACM* 55, 7 (jul 2012).

[17] K. Ramakrishnan, S. Floyd, and D. Black. 2001. The Addition of Explicit Congestion Notification (ECN) to IP. *RFC 3168, Internet Request for Comments* (2001).

[18] D. Scholz, B. Jaeger, L. Schwaighofer, D. Raumer, F. Geyer, and G. Carle. 2018. Towards a Deeper Understanding of TCP BBR Congestion Control. In *2018 IFIP Networking Conference (IFIP Networking) and Workshops*.

[19] Yeong-Jun Song, Geon-Hwan Kim, Imtiaz Mahmud, Won-Kyeong Seo, and You-Ze Cho. 2021. Understanding of BBRv2: Evaluation and Comparison With BBRv1 Congestion Control Algorithm. *IEEE Access* 9 (2021).

[20] Brian Trammell, Mirja Kühlewind, Damiano Boppart, Iain Learmonth, Gorry Fairhurst, and Richard Scheffenegger. 2015. Enabling Internet-Wide Deployment of Explicit Congestion Notification. In *Passive and Active Measurement Conference (PAM)*.

[21] Brian Trammell, Mirja Kühlewind, Piet De Vaere, Iain R. Learmonth, and Gorry Fairhurst. 2017. Tracking transport-layer evolution with PATHspider. In *Applied Networking Research Workshop (ANRW)*.

[22] Ranysha Ware, Matthew K. Mukerjee, Srinivasan Seshan, and Justine Sherry. 2019. Modeling BBR's Interactions with Loss-Based Congestion Control. In *ACM Internet Measurement Conference (IMC)*.

[23] Danesh Zeynali, Emilia N. Weyulu, Seifeddine Fathalli, Balakrishnan Chandrasekaran, and Anja Feldmann. 2024. Promises and Potential of BBRv3. In *Passive and Active Measurement Conference (PAM)*.