International Journal of Scientific Research in Computer Science, Engineering and Information Technology
ISSN : 2456-3307 OPEN ACCESS
Available Online at : www.ijsrcseit.com
doi : https://doi.org/10.32628/CSEIT2390652

Through the RTT Lens: A Comparative study of CUBIC and BBR Congestion Control

Runjeeth Nikam, Kunal Babbar, Gaurav
NJIT, Newark, New Jersey, USA

ARTICLEINFO

Article History:

Accepted: 02 Dec 2023
Published: 20 Dec 2023

Publication Issue
Volume 9, Issue 6
November-December-2023
Page Number
291-296

ABSTRACT

As the demand for high-performance internet applications continues to surge, the efficiency of network congestion control algorithms becomes a critical factor in ensuring a seamless user experience. This research paper delves into the comparative analysis of two prominent congestion control mechanisms: BBR (Bottleneck Bandwidth and Round- trip propagation time) and Cubic[1]. Both algorithms play pivotal roles in regulating data flow within networks, but their approaches differ significantly. The comparison section highlights the adaptive behavior of each algorithm, emphasizing real-world implications for diverse network scenarios. The discussion interprets the findings, offering a nuanced understanding of the strengths and weaknesses of both BBR and Cubic, thereby contributing to the broader discourse on congestion control strategies.

Keywords: ALU, Adders, Subtractors, Borrow

I. INTRODUCTION

As the Internet continues exponential growth in terms of traffic volumes, user connectivity, and link speeds, effectively managing network congestion is paramount for maintaining quality of service and low latency. When buffers and queues become excessively filled due to overly aggressive data flow, it results in a significant rise in round-trip times (RTTs)[2], leading to issues such as lag, jitter, and subpar user experiences in interactive applications. Real-time communications, gaming, stock trading, and cloud computing workloads all suffer when inflated queues result in high delays and bounded throughput.

Two predominant congestion control algorithms[3] employed today take differing approaches to detect congestion and constrain queue growth – BBR (Bottleneck Bandwidth and RTT) pioneered by Google and CUBIC adopted widely across Linux, Windows, and MacOS operating systems. Google now handles a significant portion of Internet traffic and quantifies through its vantage point the pressing need for model-based algorithms like BBR rather than purely loss-driven methods to address the buffer-bloat problem.
This paper presents an experimental comparison focused specifically on RTT performance between BBR and CUBIC algorithms based on emulations of congested bottleneck router links. The merits and

shortcomings of each method are analyzed with regards to standing queue lengths and induced latency. Results showcase BBR's ability to maintain minimal queues while fully utilizing the link capacity, contrasted with CUBIC's sawtooth latency from filled router buffer queues.

II. NETWORK CONGESTION CONTROL OVERVIEW

Congestion control refers to techniques used in communication networks to regulate traffic and avoid overloading links[4]. As more and more nodes transmit data over routers with finite packet processing and queueing capacity, queues can build up, leading to longer delays, packet losses, and choking connectivity. Analogous to highway congestion spilling over into feeder roads, network congestion results in major gridlock for all traffic flows.

Managing this congestion is critical to maintaining network efficiency and quality of service for diverse applications. Key areas impacted include:

- Latency – Interactive apps delay when queues inflate round-trip times. Real-time communications require consistent latency under 150ms for usability.
- Throughput – When queues fully overwhelm buffers, routers start dropping packets, requiring costly retransmissions. This throttles effective throughput.
- Scalability – As links speed up from Mbps to Gbps ranges, congestion control must efficiently scale up using higher bandwidth- delay products.
- Bottleneck – It refers to a point in the network where the flow of data is constrained or limited, causing a decrease in overall performance [5]. It can occur when the available bandwidth in the network is insufficient to handle the volume of data being transmitted. It is typically where the congestion in the network hits the hardest. Therefore, they set the maximum delivery rate for data across the network.

To address these pressing challenges, congestion control continues to evolve via new algorithms like BBR that incorporate model-driven methods beyond just responding to packet loss or transient delays. As the global Internet accelerates, managing congestion remains pivotal.

III. BBR CONGESTION CONTROL

The BBR congestion control algorithm aims to explicitly manage bottleneck queue lengths by matching sending rate to measured delivery rate and limiting inflight data to 1-2 bandwidth-delay products (BDP). As described in [1], it relies on continuous estimates of both bottleneck bandwidth (BtlBw) and round-trip propagation time (RTprop) to set the pacing rate and amount of data in flight.

A. Uncertainty Principle:

The uncertainty principle[6] explains that that we cannot know both the RTT and the delivery rate with perfect accuracy at the same time. If we try to measure the RTT with high accuracy, we will inevitably disturb the delivery rate. This is because the act of measuring the RTT involves sending packets back and forth, which can congest the network and reduce the delivery rate. Similarly, if we try to measure the delivery rate with high accuracy, we will inevitably disturb the RTT. This is because the act of measuring the delivery rate involves sending and receiving many packets, which can also congest the network.Therefore, even BBR cannot eliminate the uncertainty between RTT and delivery rate.

Fig. 1 Shows the effect the amount of data inflight has on delivery rate and round-trip time.

BDP is the Optimal Point[7] because it maximizes delivered bandwidth while minimizing delay and loss, both for individual connections and for the network as a whole. This is because at the BDP, the pipe is full and the inflight data excess, creates a queue at the bottleneck, which results in the linear dependence of RTT on inflight data shown in the upper graph. Packets are dropped when the excess exceeds the buffer capacity. Congestion is just sustained operation to the right of the BDP line, and congestion control is some scheme to bound how far to the right a connection operates on average.

Loss-based methods operate to the right of BDP as shown in Fig.1 because they try to achieve full bottleneck bandwidth at the cost of high delay and frequent packet loss. This is because loss-based methods rely on packet loss as a signal of congestion. When a packet is lost, the sender assumes that the network is congested and slows down its transmission rate[8]. However, packet loss can also be caused by other factors, such as network errors or congestion at other points in the network. As a result, loss-based methods can overreact to congestion and cause unnecessary packet loss and delay.

Fig.2 shows BDP as the optimal point for data transfer.

In Fig.2 the optimal point represents the maximum bandwidth that can be achieved without causing congestion, while maintaining a minimum round-trip time (RTT). BBR's approach to estimating the optimal point involves a dynamic process of probing the network and adjusting its sending rate based on observed feedback[9]. This process involves three main phases:

- Startup: During the startup phase, BBR exponentially increases its sending rate until it encounters packet loss, indicating congestion.
- Drain: Once congestion is detected, BBR enters the drain phase, where it reduces its sending rate to allow the network queues to drain.
- Probe: In the probe phase, BBR alternates between probing for the maximum bandwidth (max BW) and probing for the minimum RTT (min RTT).

BBR probes for more available bandwidth using ‘gaining cycle’ – temporarily increasing the pacing rate by 25% for one RTT before compensating with a slower rate. These cycles test whether throughput increases or RTT grows to indicate congestion. As a result, BBR converges rapidly on the optimal point that maximizes utilization with the minimum standing queue. Adaptivity remains robust across cellular connections with dynamic capacity.

IV. CUBIC CONGESTION CONTROL

CUBIC [10] and other loss-based algorithms grow the congestion window (cwnd) based on a cubic function, relying on packet loss detected through timeouts to indicate congestion requiring cwnd reduction. Initially CUBIC sets cwnd aggressively like standard slow start before switching to the cubic profile. This profile allows faster utilization of increased capacity between losses, enabling high throughput. However once loss occurs, CUBIC reverts to a linear cwnd growth.

This loss-driven approach inherently induces increased queueing delays since the network must drop packets before senders will throttle rates. As a result, CUBIC tends to operate with high buffer occupancy and latencies to fill the link, only easing off once overflow losses trigger the backoff. This sawtooth behavior repeats cyclically.

V. METHODOLOGY

To compare RTT behavior between BBR and CUBIC congestion algorithms, an experiment testbed is constructed using the Google Cloud Platform infrastructure. As shown in Figure 3, two Ubuntu virtual machines communicate over a congested 10

Mbps bottleneck link with 40ms base propagation latency. The sender VM hosts iPerf [11] to generate TCP traffic loads using either BBR or CUBIC variants.

The receiver captures packet traces on the virtual interface using TCPDump [12], storing capsules for each test. Timestamps within packet headers enable deriving RTT for each ack on the sender side by comparing send/receive time. Then this file is further filtered out by the Tshark tool and it filters the traces based on their source and destination IP addresses. Now this useful data is stored on the client machine to be visualized by using matplotlib library in python. This cycle is repeated for both the congestion control algorithms i.e. BBR and CUBIC.

Figure 4 Shows the flow of data in the experiment

VI. PERFORMANCE METRICS/ EXPERIMENTAL RESULTS

Figure 5 shows the behavior of RTT for the first 21 seconds for both CUBIC and BBR. As you can see the RTT for CUBIC is considerably high with spikes as it is a loss-based congestion algorithm. CUBIC keeps on increasing the sending rate until it identifies congestion by packet loss, hence higher round trip times. In the case of BBR, the amount of inflight data that can be sent is calculated at periodic intervals to ensure that there is no congestion. The spikes in the BBR depict its probing to see for higher Bandwidth.

This inflight data is calculated depending on the round-trip time and bottleneck bandwidth for the most recent packets. Hence, BBR ensures better throughput with fewer round-trip times.

BBR observes the increased RTT and throttles its pacing rate to stabilize at precisely the onset of congestion around 50ms RTT – the optimum point for lowest latency and full 10 Mbps utilization. In contrast, CUBIC continues expanding its congestion window, driving up the standing queue. Significantly higher latencies spike above 400ms whenever reaching capacity, inducing buffer bloat. Periodic packet loss from overflow finally compels CUBIC to drain its queue below 100ms before repeating this cyclical sawtooth behavior.

Figure 5

While BBR maintains negligible loss near zero, CUBIC suffers up to 2% loss due to its reliance on drops for congestion signals. No binomial congestion events occur with BBR since it operates with minimal queues. BBR is highly sensitive to round-trip time (RTT) variations[13]. It dynamically adjusts its behavior based on the observed RTT, adapting to changing network conditions. CUBIC is less sensitive to RTT changes. It tends to maintain a more stable congestion window size over a wide range of RTT values.

VII. CONCLUSION

Managing standing queues and round-trip times is critical to providing low latency network services. Applications from VoIP calls to gaming rely on congestion control algorithms to curb buffer bloat. As evidenced in our emulated test cases, BBR’s model-based approach explicitly targets an optimum operating point with the minimum delay. RTT is held stable independent of configured buffer size or competing traffic levels. In contrast, CUBIC’s loss-triggered window adjustments incur repeated buffer growth and much higher RTTs peaking near capacity. While both algorithms do converge on fair bandwidth allocation when contending on a bottleneck link, CUBIC pays a heavy price in terms of elevated latency and periodic packet loss from drops. This induces jitter and quality degradation for real-time applications. BBR's ability to maintain loss-free operation with minimized queues demonstrates significant advantages for the next-generation of delay-sensitive networked services. Varying BBR’s key parameters would also gauge adaptivity to more heterogeneous paths[14]. Evaluating benefits for streaming video and gaming use cases could better illustrate the quality perceived by end users when using advanced congestion control.

VIII. FUTURE WORK

Dynamic ProbeRTT Adjustment: One of the notable distinctions between BBRv1 and BBRv2 is the constant probeRTT value used (10ms and 5ms, respectively). A promising avenue for future work involves exploring dynamic mechanisms to adjust the probeRTT value adaptively based on network conditions. This dynamic adjustment could enhance BBR's responsiveness to diverse network scenarios, optimizing its performance further.

Machine Learning Integration: Leveraging machine learning techniques to predict and optimize probeRTT values based on real-time network characteristics could be a valuable research endeavor. This approach could potentially lead to more efficient congestion control strategies that adapt to changing network conditions and traffic patterns. Security and Robustness: Evaluating the resilience of BBR and Cubic to various network attacks and anomalies is another avenue for future research. Ensuring that these algorithms maintain stable and efficient performance even in adverse conditions is essential for network reliability.

Energy Efficiency: With the growing importance of energy-efficient networking, investigating the energy consumption implications of BBR and Cubic algorithms is an area of increasing relevance. Researchers can explore ways to minimize energy usage while maintaining high network performance.

IX. REFERENCES

[1]. N. Cardwell, “BBR: Congestion-Based Congestion Control.” [Online]. Available: https://research.google/pubs/pub45646/

[2]. Larry L. Peterson, Bruce S. Davie, “Computer Networks (Fifth Edition), 2012.”

[3]. M. Geist, “Overview of TCP Congestion Control Algorithms.” [Online]. Available: https://www.net.in.tum.de/fileadmin/TUM/NET/NET-2019-06-1/NET-2019-06-1_03.pdf

[4]. S. H. Low, F. Paganini, and J. C. Doyle, “‘Internet congestion control,’ in IEEE Control Systems Magazine, vol. 22, no. 1, pp. 28-43, , doi: 10.1109/37.980245.,” 2002.

[5]. Locating Internet Bottlenecks: Algorithms, Measurements, and Implications. (2004). United States: Carnegie- mellon univ pittsburgh pa school of computer Science.

[6]. D. Scholz, B. Jaeger, L. Schwaighofer, D. Raumer, F. Geyer, and G. Carle, “‘Towards a Deeper Understanding of TCP BBR Congestion Control,’ 2018 IFIP Networking Conference (IFIP Networking) and Workshops, Zurich, Switzerland, 2018, pp. 1-9, doi: 10.23919/IFIPNetworking.2018.8696830.”

[7]. T. V. Lakshman and U. Madhow, “‘The performance of TCP/IP for networks with high bandwidth-delay products and random loss,’ in IEEE/ACM Transactions on Networking, vol. 5, no. 3, pp. 336-350, June 1997, doi: 10.1109/90.611099.”

[8]. C. Williamson and D. Cheriton, “Loss-Load Curves: Support for Rate-Based Congestion Control in High-Speed Datagram Networks”.

[9]. M. Hock, R. Bless, and M. Zitterbart, “M. Hock, R. Bless and M. Zitterbart, ‘Experimental evaluation of BBR congestion control,’ 2017 IEEE 25th International Conference on Network Protocols (ICNP), Toronto, ON, Canada, 2017, pp. 1-10, doi: 10.1109/ICNP.2017.8117540.”.

[10]. V. Sharma and S. Poojary, "'Analytical Model for Congestion Control and Throughput with TCP CUBIC Connections,' 2011 IEEE Global Telecommunications Conference - GLOBECOM 2011, Houston, TX, USA, 2011, pp. 1-6, doi: 10.1109/GLOCOM.2011.6134001.". [11] C.-H. Hsu and U. Kremer, "IPERF : A Framework for Automatic Construction of Performance Prediction Models".

[11]. "The Tcpdump Group. "tcpdump - a powerful tool for network monitoring and data acquisition." [Online]. Available: https://www.tcpdump.org/

[12]. N. Cardwell, Y. Cheng, and C. S. Gunn, "BBR congestion control: an update.," Mar. 2017. [14] A. Mishra, T. W. Han, and B. Leong, "Are we heading towards a BBR-dominant internet?," Oct. 2022.

Cite this article as :

Runjeeth Nikam, Kunal Babbar, Gaurav, "Through the RTT Lens : A Comparative study of CUBIC and BBR Congestion Control", International Journal of Scientific Research in Computer Science, Engineering and Information Technology (IJSRCSEIT), ISSN : 2456-3307, Volume 9, Issue 6, pp.291-296, November-December-2023. Available at doi : https://doi.org/10.32628/CSEIT2390652
Journal URL : https://ijsrcseit.com/CSEIT2390652