2018 30th International Teletraffic Congress

Impact of TCP BBR on CUBIC Traffic: A Mixed Workload Evaluation

Habtegebreil Haile, Per Hurtig,
Karl-Johan Grinnemo, Anna Brunstrom
Karlstad University
Karlstad, Sweden

Eneko Atxutegi Narbona and Fidel Liberal
University of the Basque Country
Bilbao, Spain

Åke Arvidsson
Kristianstad University
Kristianstad, Sweden

***Abstract*—A recently proposed congestion control algorithm (CCA) called BBR (Bottleneck Bandwidth and Round-trip propagation time) has shown a lot of promise in avoiding some of the problems that have plagued loss-based CCAs. Nevertheless, deployment of a new alternative algorithm requires a thorough evaluation of the effect of the proposed alternative on established transport protocols like TCP CUBIC. Furthermore, evaluations that consider the heterogeneity of Internet traffic sizes would provide a useful insight into the deployability of an algorithm that introduces sweeping changes across multiple algorithm components. Yet, most evaluations of BBR's impact and competitive fairness have focused on the steady-state performance of large flows. This work expands on previous studies of BBR by evaluating BBR's impact when the traffic consists of flows of different sizes. Our experiments show that under certain circumstances BBR's startup phase can result in a significant reduction of the throughput of competing large CUBIC flows and the utilization of the bottleneck link. In addition, the steady-state operation of BBR can have negative impact on the performance of bursty flows using loss-based CCAs over bottlenecks with buffer sizes as high as two times the bandwidth-delay product.**

I. INTRODUCTION

Packet loss has long been used as the primary indicator of congestion along network paths on the Internet. Based on the seminal work on congestion control by Jacobson [1], a number of loss-based Congestion Control Algorithms (CCAs) have been proposed and undergone multiple modifications to offer better performance in terms of faster convergence, better loss-recovery, improved fairness, etc. However, the dominant loss-based CCAs, including the widely deployed CUBIC [2] algorithm, still have performance issues resulting from their built-in dependence on loss for detecting congestion.

BBR [3] (Bottleneck Bandwidth and Round-trip propagation time) is a recently proposed CCA from Google that takes a different approach to congestion control. Particularly, it does not rely on loss as the primary indicator of congestion. The manifestation of large persistent packet queues in network nodes with excessive buffer space, i.e. bufferbloat [4], is one of the by-products of using loss-based CCAs that motivated the development of BBR [5]. Without properly configured Active Queue Management (AQM), loss-based CCAs will keep a big portion of a large bottleneck buffer occupied for extended periods of time. This can result in performance degradation of latency sensitive flows sharing the bottleneck. Another problem with loss-based CCAs is the overreaction of the CCAs to pre-congestion losses. The overreaction of loss-based CCAs to pre-congestion losses is the reason behind the low throughput of large flows in a shallow-buffered bottleneck shared with bursty loss-based flows.

BBR's advantage in avoiding the problems mentioned above, along with Google's active involvement in improving and deploying the algorithm have made BBR a promising alternative CCA for wide scale deployment. However, introducing an alternative CCA like BBR on the Internet implies a gradual deployment of the new algorithm. The new algorithm will then have to co-exist with current loss-based algorithms for an extended period of time. This gives rise to a need for a thorough understanding of the interactions between the new CCA and the existing loss-based CCAs. Thus, this works aims to contribute to the understanding of the interaction between the new BBR CCA and the pre-existing loss-based CCAs. Since CUBIC is one of the most commonly used loss-based CCAs (being the long-standing default CCA in Linux, and recently the default in Windows [6] as well), we select CUBIC as a representative loss-based CCA for our evaluations.

In addition to the above considerations, evaluations of a new alternative CCA should consider the diversity in the size of the data transfers that take place over the Internet. One reason is that a big portion of network data transfers are relatively small and thus data transmissions might not progress to the steady-state phase of the CCA. Additionally, bandwidth values for bottlenecks on access links span a wide range, thus the same data transfer could be dominated by different phases of a CCA in different networks, locations in the network, or at different times of the day. Furthermore, the bottleneck can occur at any point in the communication path, and buffer sizes can differ along that path. However, so far, evaluations of BBR have focused on bulk transfers and largely ignored data transfers with a mix of large and small flow sizes. In these types of data transfers, the interaction does not last long enough to achieve a sustained throughput. Therefore, in addition to throughput for large data transfers, evaluations that gauge the performance of small data transfers by employing completion time as a metric are needed.

In this paper, we present results that show that a burst of traffic that consists of successively arriving medium-sized BBR flows can cause a significant reduction in the throughput large flows of loss-based CCAs in bottlenecks with small

978-0-9883045-5-0/18/$31.00 ©2018 ITC
DOI 10.1109/ITC30.2018.00040

buffers. The significant reduction in the throughput of the large CUBIC flows that are competing with bursty BBR traffic is shown to result in a much lower bottleneck utilization. Additionally, we also evaluate the impact of BBR's steady-state operation on small- and medium-sized flows of the loss-based CCA. The results of our experiments show that small- and medium-sized flows that use CUBIC CCA experience significantly longer completion times when competing with large BBR flows in BDP (Bandwidth-Delay-Product) buffer bottlenecks. We also show that when competing with large BBR flows, relatively small flows with loss-based CCAs can experience longer completion times in bottlenecks with buffer size as large as 2 BDP. On a side note, our experiments confirm that large BBR flows do manage to avoid bufferbloat, thus allowing faster completion time for competing bursty flows in bottlenecks with large buffer.

The remainder of the paper is organized as follows: Section II gives a brief overview of the CCAs used in our experiments. The characteristics of the network and traffic used in our experiments are presented in Section III. An overview of the observations and general patterns that hold true for most of the experiment scenarios is presented in Section IV. Section VI gives more detail about the impact of BBR's steady state while Section V focuses on the impact of BBR's startup. Section VII briefly discusses about the implications of the results and reflects on current activities that are related to the issues shown in this paper. Possible future undertakings to address issues that need to get more attention for seamless incremental deployment of BBR are also pointed out. Related works are presented in Section VIII before concluding in Section IX.

II. REVIEW OF EVALUATED CCAS

TCP CUBIC is a CCA optimized for connections with high bandwidth-delay products (BDP). The CUBIC congestion window (*cwnd*) increments are based on a cubic function of the time since the last congestion event. In the congestion avoidance phase that follows a fast recovery, the *cwnd* increases in a concave manner until it reaches a W_{max} value, which is the size of the congestion window when the last loss event occured. After plateauing around the W_{max} value for a while the *cwnd* increase continues in a concave manner until the next loss event. The Linux TCP CUBIC implementation also incorporates a hybrid slow-start mechanism [7], which is a proactive startup mechanism that aims to detect and avoid congestion before packet loss happens. The hybrid slow start mechanism adjusts the *ssthresh* based on congestion signals obtained from ACK train length and increases in packet delays.

BBR CCA avoids the use of packet loss as a signal of congestion, and instead opts for modeling the bottleneck where congestion is occurring. The underlying principle behind BBR is that an end-to-end connection is governed by the bandwidth capacity available on the bottleneck link and the round-trip propagation time (RTT). Therefore, BBR tries to manage congestion by using estimates of the minimum end-to-end RTT of a flow and the available bandwidth at a bottleneck to control its sending rate and amount of inflight packets. In steady state, BBR continuously probes for bandwidth and RTT. Bandwidth probing is performed through a periodic 25% increase in the sending rate, and minimum RTT probing is done by lowering the amount of inflight packets to a minimum value of four packets every 10 seconds, unless a lower value than the current minimum RTT is detected in the previous 10-second interval. In steady state, BBR sends data at the estimated bandwidth rate for the majority of a bandwidth probing cycle. It also limits the amount of inflight data to a value that is equal to the product of estimated bandwidth and propagation RTT. BBR has an exponential startup that exits when it fails to detect a significant increase (more than 25%) in the delivery rate over three RTTs. Upon exit from startup, BBR drains the queue built by the exponential bandwidth search and enters the steady-state phase.

TABLE I
SUMMARY OF THE NETWORK AND FLOW PARAMETERS USED IN THE EXPERIMENTS

Bottleneck bandwidths	5 Mbps, 10 Mbps, 20 Mbps, 30 Mbps, 40 Mbps, 50 Mbps, 60 Mbps, 70 Mbps
RTTs	45 ms
Buffer sizes	0.5 BDP, 1 BDP, 2 BDP, 4 BDP, 8 BDP
Small flow sizes	100 KB
Medium flow sizes	500 KB, 1 MB, 2 MB,
Large flow sizes	30 MB, 90 MB [1]
CCAs	BBR, CUBIC
Interflow gaps	0.0 s, 0.1 s, 0.2 s, 0.3 s, 0.4 s, 0.5 s 0.6 s, 0.7 s, 0.8 s, 0.9 s [2]

III. EXPERIMENTAL SETUP

Table I summarizes the network and flow settings that were used in the experiments.

A. Network Emulation

The topology shown in Figure 1 was used to emulate a bottleneck link and conduct experiments on different flow combinations. The link between each node is capable of carrying up to 10 Gbps of traffic. The traffic source node at one end of the setup generates different traffic combinations destined for the traffic sink node at the other end of the setup. The edge nodes run Linux 4.11.0 operating system on an Intel Xeon X5550 processor with four cores.

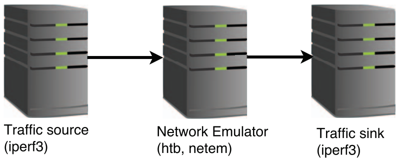


Fig. 1. Physical node connections for emulation experiments.

---

[1] 90 MB was used only for increasing interflow gap experiments on 40 Mbps bottleneck with 0.5 BDP buffer.

[2] Interflow gaps other than 0.1 s were used only in experiments where 90 MB large flows are competing with 1 MB medium-sized flows in a 40 Mbps bottleneck with 0.5 BDP buffer.

The middle node is used to create an emulated bottleneck and network. The node runs Linux 4.10. A bottleneck is emulated using the *htb* (hierarchical tocken bucket) *qdisc*, and buffer sizes are configured through a *netem qdisc*. A *netem qdisc* is also used to apply the delays that emulate the RTT.

The minimum end-to-end delay is configured to be the same (45 ms) for all flows to make sure results are not biased due to difference in RTT . In order to cover a wide range of current access speeds, we set the bandwidth values of the emulated bottleneck range between 5 Mbps and 70 Mbps. The buffer size for each bottleneck is set to have different values that are multiples of the bandwidth delay product (BDP) ranging between 0.5 BDP and 8BDP.

B. Test Traffic

Our experiments involve data flows of variable sizes whose data transfers are dominated by different phases of a CCA, e.g., the startup phase or the steady-state phase. For most of the tested bottlenecks, the data transfers of the large flows are dominated by the steady-state phase of the CCA, while medium- and small- sized flows complete most of their data transfers during the startup phase. If not explicitly said otherwise, we refer to 30 MB as a large flow, 1 MB as a medium-sized flow, and 100 KB as a small flow. To assess the role of the size of the medium-sized flows, flow sizes of 500 KB and 2 MB were also used as medium-sized flows for some experiments. We also used large flows of 90 MB for some experiments that require longer experiments for reliability. In general, the experiments involve a large flow competing with medium-sized or small flows and medium-sized flows competing with other medium-sized flows. The experiments were done with competing pairs of flow sizes with the same CCA as well as with pairs of different CCAs.

All flows are generated using iperf3. We wait 0.1 seconds before issuing the next iperf3 call to transmit the subsequent small or medium-sized flow after the completion of the previous small or medium-sized flow, i.e., interflow gap. Additionally, we also perform experiments with different interflow gaps to observe the impact. In all the experiments, the first of a succession of small- and medium-sized flows competing with a large flow is launched 0.5 seconds after the start of the large flow. This delay is introduced to ensure that small- and medium-sized flows avoid the startup phase of the large flow. Each experiment on a specific combination of flow sizes, CCA pairs, bottleneck bandwidth, and buffer size is repeated ten times.

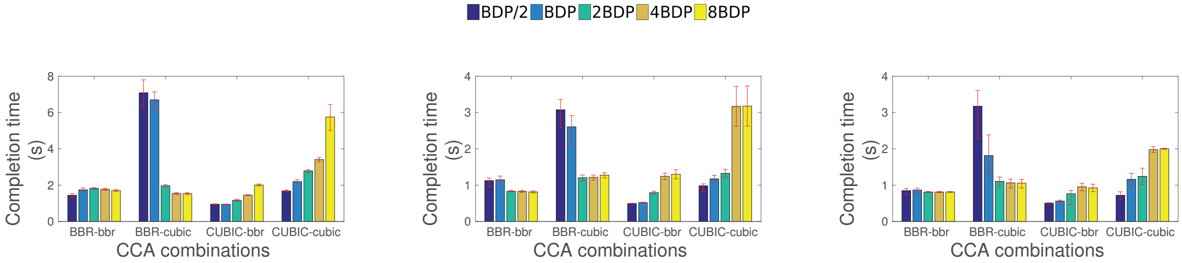
IV. OVERVIEW OF OBSERVATIONS

In this section we briefly discuss general observations for competing large and medium-sized flow experiments with different CCA combinations. The plots in Figures 2, 3 and 4 show the completion time of medium-sized flows, the throughput of large flows, and the bottleneck utilizations for different CCA and flow size combinations, respectively. We provide the confidence intervals for the completion time measurements, which were much more variable than the throughput and utilization measurements. The plots are given for four CCA combinations where the CCA for the large flow is written with all capital letters while the CCA of the small- or medium-sized flow is written in all small letters: large BBR flows competing with medium-sized BBR flows (*BBR-bbr*), large BBR flows competing with medium-sized CUBIC flows (*BBR-cubic*), large CUBIC flows competing with medium-sized BBR flows (*CUBIC-bbr*), and large CUBIC flows with medium-sized CUBIC flows (*CUBIC-cubic*). In the remaining part of this section each of the above traffic combinations are discussed one by one. We first discuss the homogeneous CCA cases in the next two paragraphs to serve as a baseline. We then proceed to discussing the mixed CCA observations by making comparisons with the baselines. Detailed observations from varying the experiment parameters for the mixed CCA cases are given in the next two sections.

When a large CUBIC flow is competing with medium-sized CUBIC flows the large flow tends to cause the medium-sized flows to experience longer completion time for larger bottleneck buffer sizes, as can be observed from the *CUBIC-cubic* results in Figure 2. This is primarily because the completion time of small- and medium-sized flows is dependent on the round trip time of packets and the bufferbloat effect of the large CUBIC flow will cause extended round trip times in large buffers. On the other hand, in small buffer bottlenecks large BBR flows are forced to a significantly lower throughput than the throughput they achieve in bottlenecks with large buffers. In most cases, a noticeable reduction in throughput of large CUBIC flows is observed when competing with medium-sized CUBIC flows over bottleneck links with 0.5 BDP buffers. This also seems to result in a reduction in bottleneck utilization as the arriving medium-sized CUBIC flows are not able to make up for the reduced throughput of the large flow.

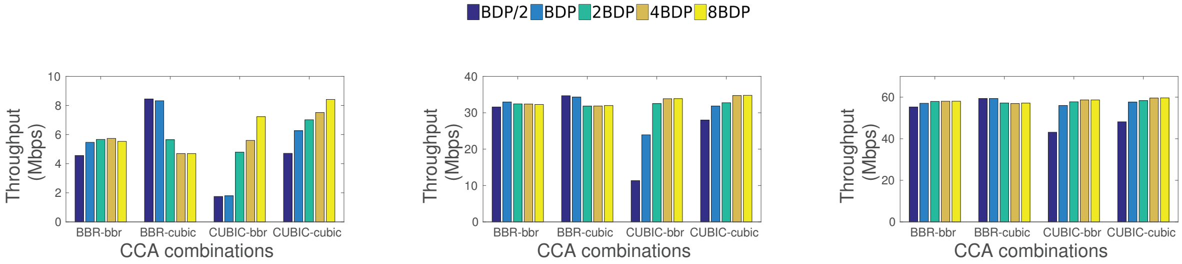
In the figures, flow combinations that are composed only of BBR flows, which are labeled *BBR-bbr* on the x-axis, show consistent performance, with respect to buffer size, in the throughput of large flows and completion time of medium-sized flows. This performance is achieved because large BBR flows are do not reduce the amount of in-flight packets by half if they experience early loss in small buffers. Moreover, medium-sized BBR flows do not experience increased delays as the buffer size shared with large BBR flows increases. This is because the amount of in-flight packets of large BBR flows is limited to keep the end-to-end RTT as close as possible to an estimated minimum RTT.

For flow combinations consisting of a large CUBIC flow and a medium-sized BBR flow, we observe that in large buffers the medium-sized BBR flows experience longer completion time when competing with large CUBIC flows. However, the effect is not as pronounced as the one experienced by the medium-sized CUBIC flows in similar conditions. In contrast, the reduction in throughput experienced by the large CUBIC flow in lower buffer sizes is much more significant with medium-sized BBR flows than with medium-sized CUBIC flows. We can see this by comparing *CUBIC-bbr* and *CUBIC-cubic* for 0.5 BDP buffer in Figure 3. Furthermore, for some bottleneck



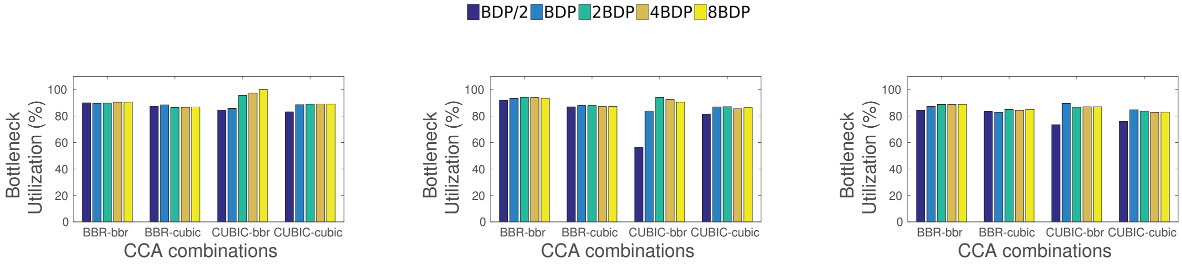
(a) 10 Mbps (b) 40 Mbps (c) 70 Mbps

Fig. 2. Completion time of medium-sized flows for different CUBIC and BBR traffic combinations over different bottlenecks with a range of buffer sizes.



(a) 10 Mbps (b) 40 Mbps (c) 70 Mbps

Fig. 3. Throughput of large flows for different CUBIC and BBR traffic combinations over different bottlenecks with a range of buffer sizes.



(a) 10 Mbps (b) 40 Mbps (c) 70 Mbps

Fig. 4. Bottleneck utilization for different CUBIC and BBR traffic combinations over different bottleneck with a range of buffer sizes.

bandwidths, the *CUBIC-bbr* flow combination is observed to results in noticeable reductions in the utilization of bottlenecks with buffer sizes less than or equal to BDP. The issue of reduced bottleneck utilization is discussed in detail in Section V.

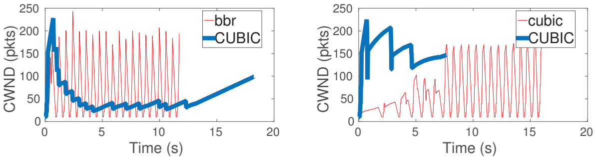
For a flow mix consisting of a large BBR flow competing with medium-sized CUBIC flows, the general trend is that the completion time of the CUBIC flows becomes longer as the bottleneck buffer becomes smaller. This can be observed from the *BBR-cubic* labeled results in Figure 2. In small buffer bottlenecks, the large BBR flows achieve improved throughput at the expense of the CUBIC flows. The extended completion time of the CUBIC flows competing with large BBR flows in small buffer (0.5 BDP and 1 BDP) bottlenecks is comparable to the completion time experienced by medium-sized CUBIC flows competing with large CUBIC flows in large buffer (4BDP and 8BDP buffers) bottlenecks. Figure 2 shows that, of all the tested buffer sizes, a 2 BDP buffer results in the most comparable completion time between medium-sized (1 MB) CUBIC flows competing with large BBR flows and medium-sized CUBIC flows competing with large CUBIC flows. This observation agrees with the previously identified 2 BDP buffer recommendation for fair operation between large BBR and large CUBIC flows [8]. However, detailed experiments presented later in Section VI show that CUBIC flows with size much lower than 1 MB might experience longer completion time when competing with large BBR flows in 2 BDP buffer bottlenecks.

V. IMPACT OF BBR STARTUP

This section provides detailed observations for the *CUBIC-bbr* flow combination presented in Section IV. Therefore it focuses on the effect of successive bursts of BBR flows on the throughput of large CUBIC flows, and the utilization of the bottleneck shared between a large CUBIC flow and the BBR bursts. The last subsection is a supplementary look into the impact of medium-sized BBR bursts on the performance of CUBIC bursts of the same size.

A. Effect of startup in bottlenecks with small buffer

In Section IV, it was shown that large CUBIC flows competing with medium-sized BBR flows in bottlenecks with a buffer size that is less than or equal to BDP suffer a substantial throughput reduction. The plots in Figure 5 show the congestion window of a large CUBIC flow along with the congestion window of successively arriving medium-sized BBR (Figure 5a) and CUBIC (Figure 5b) flows in a 40 Mbps bottleneck with 0.5 BDP of buffer. As follows from Figure 5a, as the BBR startup of the medium-sized flow increases exponentially without reacting to loss, it will quickly saturate the path and outcompete the large CUBIC flow. Thus, the successively arriving medium-sized BBR flows force the large CUBIC flow to be restricted to a very small share of the total capacity (BDP + buffer), which is about 240 packets for a 45 ms connection through a 40 Mbps bottleneck.



(a) *CUBIC-bbr* flow combination (b) *CUBIC-cubic* flow combination

Fig. 5. Congestion window of (a) a large CUBIC flow and competing medium-sized BBR flows (b) a large CUBIC flow and competing medium-sized CUBIC flows in a 40 Mbps bottleneck with 0.5 BDP buffer.

B. Effect of flow size on throughput and utilization

The observations presented in Section IV were for experiments with 1 MB-sized flows. In this subsection we present the variation observed when using medium-sized BBR flows other than 1 MB. Figures 6 and 7 show the throughput of the large flow and the utilization of the bottleneck when a large CUBIC flow is competing with medium-sized BBR and CUBIC flows of different sizes. The plots give the throughput and utilization for bottlenecks of different bandwidths. Each bottleneck has a buffer size of 0.5 BDP.

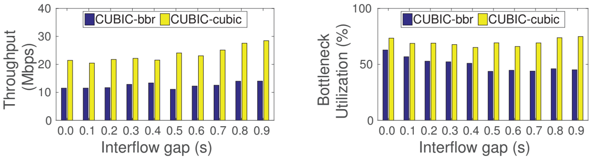
In the plots, it can be seen that as the size of the medium-sized BBR flows increases the effect of the BBR flows is experienced at higher and higher bottleneck bandwidths. This is because for higher bottleneck bandwidths the impact of the smaller medium-sized BBR flows (500 KB) is reduced. The bandwidth unused by the large CUBIC flow is large enough to allow the 500 KB BBR flows to complete their data transfer without interacting with the large CUBIC flow to incur loss that results in further throughput reduction.

An interesting observation is the significant reduction in the utilization of the bottleneck at specific bandwidths depending on the size of the medium-sized flow. In Figure 7, it can be seen that the utilization of a bottleneck is very low for particular combinations of medium file sizes and bandwidths: for 500 KB the dip occurs at 20 Mbps; for 1 MB at 40 Mbps and for 2 MB at 60 Mbps. For low bottleneck bandwidths, a high utilization is achieved because the given medium-sized BBR flows quickly reach steady-state operation and are able to send at a rate that is closer to the bottleneck capacity while leaving a very small capacity for the large CUBIC flow.

As the bottleneck bandwidth increases, the amount of time the BBR flows spend in the high-utilization, steady-state phase is reduced while the impact of the exponential probing phase of the continuously arriving BBR flows limits the large CUBIC flows to a very low throughput. For a BBR flow of a given size, a specific bandwidth exists for which the BBR flow spends the smallest amount of time in the steady-state phase while the impact of the startup phase on the throughput of a competing large CUBIC flow is the highest. This is the bandwidth value where the lowest bottleneck utilization is observed for the given BBR flow size.

C. Effect of interflow spacing on utilization

This subsection expands the observations in Section IV that were given for a 0.1 seconds gap between the completion of one medium-sized flow and the launch of another flow (interflow gap). In Figure 8, the throughput of large flows and the utilization of the bottleneck are given for increasing time gaps. The plots are for a bottleneck with 40 Mbps bandwidth and 0.5 BDP buffer size, which is the bottleneck where the lowest bottleneck utilization is observed when the bottleneck is shared between 1 MB-sized flows and a large CUBIC flow. Here a 90 MB large flow is used to allow a high number of medium-sized flows to overlap with the large flow when the gap is increased.



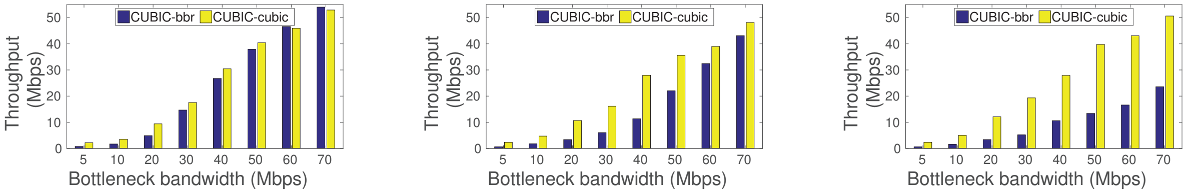
(a) Throughput of large CUBIC flows (b) Utilization of bottleneck

Fig. 8. Throughput of large CUBIC flows and bottleneck utilization for different interflow gaps between arriving medium-sized (1 MB) BBR and CUBIC flows in 40 Mbps bottleneck with a 0.5 BDP buffer.

From the figures, it can be seen that the biggest difference between the utilization of *CUBIC-cubic* traffic and *CUBIC-bbr* traffic occurs for the longest interflow gap. This is because despite the longer gap, in the interval between two BBR flows, the *cwnd* of the large CUBIC flow does not seem to grow by an amount that is much greater than in shorter gaps.

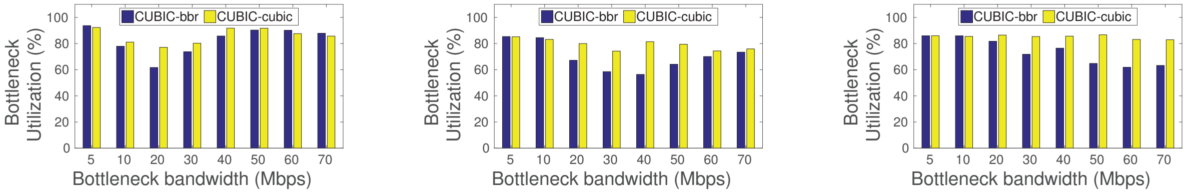
D. Impact on completion time of medium-sized CUBIC flows

Figure 9 shows the completion time for medium-sized CUBIC flows when they are sharing the bottleneck with other BBR and CUBIC flows of the same size. The bottleneck in the figure has a bandwidth of 20 Mbps, which is high enough to allow medium-sized (1 MB) CUBIC and BBR flows to complete most of their data transfer in the startup phase.



(a) 500 KB (b) 1 MB (c) 2 MB

Fig. 6. Throughput of large CUBIC flows competing with medium-sized BBR and medium-sized CUBIC flows over different bottlenecks with 0.5 BDP buffer.



(a) 500 KB (b) 1 MB (c) 2 MB

Fig. 7. Bottleneck utilization for different medium-sized BBR and CUBIC flows competing with large CUBIC flows in bottlenecks with 0.5 BDP buffer.

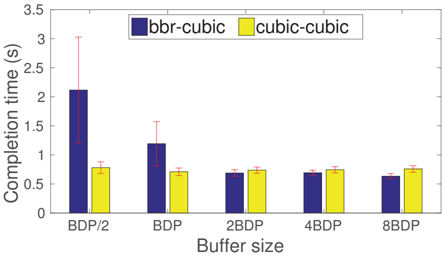


Fig. 9. Completion time of medium-sized (1 MB) CUBIC flows competing with BBR and with CUBIC flows of the same flow size in different buffer sizes of a 20 Mbps bottleneck.

It can be seen that when medium-sized CUBIC flows are competing against other CUBIC flows, the observed completion time does not vary with buffer size, and the variability of the completion times experienced for each buffer size is very limited. However, when medium-sized CUBIC flows are competing with BBR flows of the same size in small buffers, the CUBIC flows experience a very high average completion time that shows big variations depending on when the BBR flow arrives. If the BBR flow arrives after the CUBIC flow has grown its congestion window to a substantial value, then the CUBIC and BBR flows are able to achieve a fair completion time. However, if the CUBIC flow arrives after the BBR flow then it will experience loss and enter the congestion avoidance phase after reducing its congestion window. If the loss is experienced very early, the CUBIC flow will not be able to complete its transfer for a long time since successive BBR flows arrive and force repeated loss and reduction of the congestion window.

VI. IMPACT OF BBR STEADY-STATE

In this section, we present a detailed evaluation of the negative impact of large BBR flows competing with medium-sized CUBIC flows in small buffer bottlenecks. This section extends the observations for the *BBR-cubic* CCA combination made in Section IV. We evaluate the impact of the steady-state mechanism of large BBR flows on the completion time of medium-sized CUBIC flows. We also evaluate observations that relate to small CUBIC flows that slightly deviate from the general observation given in Section IV. We present medium-sized CUBIC flow results for a 20 Mbps bottleneck. This is because at a bandwidth of around 20 Mbps the medium-sized (1 MB) CUBIC flows experience the negative effects of competing with large BBR flows in small buffers but do not affect the large BBR flow in the same way as large CUBIC flows do in large buffers. Results from 5 Mbps bottlenecks are used for evaluating the impact on small (100 KB) CUBIC flows for the same reason.

A. *Effect of buffer size on the completion time of medium-sized flows*

In Figure 10, it can be seen that the biggest increase in completion time is observed for a buffer size of 0.5 BDP. This big difference is partly due to the fact that a single large flow that uses a loss-based CCA like CUBIC is very inefficient in utilizing a shallow-buffer bottleneck [9], thus allowing better completion times for the medium-sized flows sharing the bottleneck. However, the mean completion time of medium-sized CUBIC flows competing with large BBR flows is also longer over bottlenecks with a BDP-sized buffer, which is considered sufficient for an acceptable utilization by long-lived CUBIC flows. In bottlenecks with a BDP-sized buffer, medium-sized CUBIC flows experience a mean completion time that is significantly greater than the mean completion time experienced when competing with large CUBIC flows.

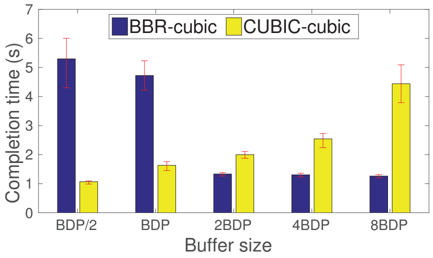
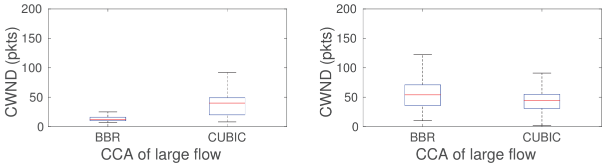


Fig. 10. Completion time of medium-sized (1 MB) CUBIC flows competing with large BBR and large CUBIC flows in different buffer sizes of a 20 Mbps bottleneck.



(a) Buffer size = BDP (b) Buffer size = 2 BDP

Fig. 11. Box plot of the *cwnd* of medium-sized CUBIC flows competing with large flows of different CCAs over a 20 Mbps bottleneck with (a) BDP-sized and (b) 2 BDP-sized buffers.

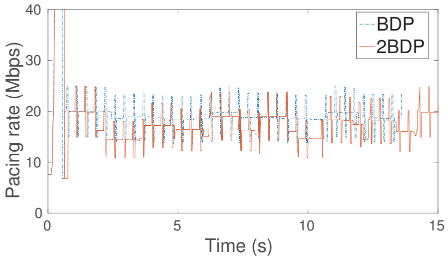


Fig. 12. Pacing rate of a large BBR flow competing with medium-sized CUBIC flows over a 20 Mbps bottleneck with BDP- and 2 BDP-sized buffers.

The box plots in Figure 11 show the congestion window values for medium-sized CUBIC flows that are competing with large flows of the two CCAs. It can be seen that when the buffer size is equal to 1 BDP, a large BBR flow limits the *cwnd* of medium-sized CUBIC flows to very low values (Figure 11a). On the other hand, when the buffer size is increased to 2 BDP, medium-sized CUBIC flows competing with a large BBR flow are able to obtain a higher *cwnd* value (Figure 11b). This value is much closer to the value achieved when the medium-sized flows are competing with large CUBIC flows. The reason is that a 2 BDP-sized buffer offers enough space for the *cwnd* of medium-sized flows to grow to a level that is significant enough for the large BBR flow to be able to detect lower delivery rate within a filtering window. Figure 12 shows that the pacing rate of a large BBR flow competing with medium-sized CUBIC flows is significantly lower for long durations over the 20-Mbps bottleneck with a 2 BDP-sized buffer than in a similar botteleneck with a 1 BDP buffer. In the 2 BDP case, the first flow is typically impacted the most until BBR detects its existence and lowers its sending rate. Subsequent medium-sized flows will take advantage of this reduction in BBR's sending rate to obtain a larger share of the bottleneck and achieve shorter completion times.

B. Observations for small CUBIC flows

Figure 13 shows that, in contrast to what was observed for medium-sized flows, small CUBIC flows can experience longer completion times when interacting with large BBR flows in 2 BDP buffer bottlenecks. As can be seen from the ECDFs in Figure 14, the completion times in 2 BDP buffers are much closer to the BBR maximum delivery rate filtering window of 10 RTTs. Therefore, the large BBR flow is not able to detect the small flows even in 2 BDP buffers. In addition, most of the small CUBIC flows were observed to experience between 11 and 15 packet retransmissions when competing with a large BBR flow. On the other hand, while competing with a large CUBIC flow, with the exception of the first two flows, none of the small CUBIC flows experience retransmissions in 2 BDP buffers. This implies that the sending rate (Figure 15) of the large BBR flow will remain high and the 2 BDP buffer will not be enough to avoid high number of retransmissions. As a result, the small CUBIC flows will not be able to achieve the same completion time performance as when they are competing with large CUBIC flows in the 5 Mbps and 2 BDP buffer bottleneck. For a bottleneck with BDP buffer, the behavior of the small CUBIC flows competing with large BBR flows is more or less guided by similar interactions as those observed between medium-sized CUBIC flows and large BBR flows.

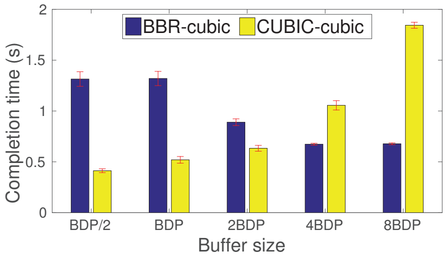
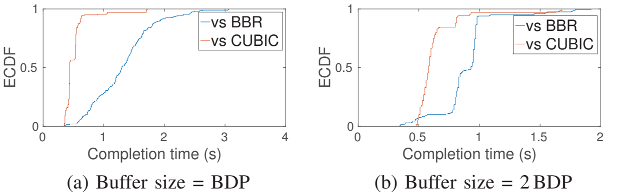


Fig. 13. Completion time of small (100 KB) CUBIC flows competing with large BBR and large CUBIC flows in different buffer sizes over a 5 Mbps bottleneck.

On the other hand, for bigger buffer sizes, small and medium-sized CUBIC flows were able to achieve better completion times when competing with large BBR flows than when competing with large flows using the loss-based CUBIC CCA. As can be seen in Figure 13, the biggest improvements were observed for the largest buffer (8BDP). This is because large



(a) Buffer size = BDP (b) Buffer size = 2 BDP

Fig. 14. Emprical CDF of the completion times of small CUBIC flows competing with large flows with different CCA over a 5 Mbps bottleneck with (a) BDP and (b) 2 BDP buffers.

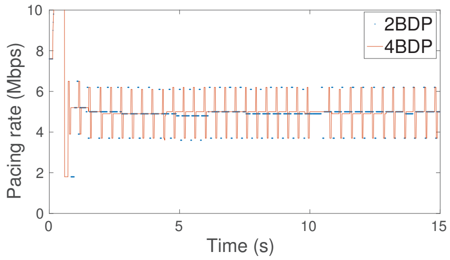


Fig. 15. Pacing rate of a large BBR flow competing with small CUBIC flows over a 5 Mbps bottleneck with BDP- and 2 BDP-sized buffers.

BBR flows do not create big bottleneck queues like large loss-based flows in bottlenecks with large buffers.

VII. DISCUSSION

According to [5], BBR aims to mitigate two main problems associated with loss-based CCAs; bufferbloat in large-buffers and low throughput in shallow buffers. As shown in Figure 3, large CUBIC flows competing with bursts of CUBIC flows (*CUBIC-cubic*) do not experience much problem with respect to reduced throughput and utilization over bottlenecks with a buffer size of 1 BDP or higher. However, the results of this paper show a significant increase in completion time of bursty CUBIC traffic competing with a large BBR flow over bottlenecks with buffer sizes as high as a BDP. Longer completion times were also observed for smaller CUBIC traffic bursts of 100 KB over a bottleneck with a buffer size of 2 BDP. In general, BBR strives to avoid inefficient low throughput when competing with bursty cross-traffic over shallow buffer bottlenecks that induce the low throughput inefficiencies of loss-based CCAs. An unintended consequence of this is the longer completion time experienced by the bursty CUBIC cross-traffic even in bottlenecks with the buffer approaching sizes that are not normally associated with the throughput and utilization inefficiencies of large CUBIC flows .

The BBR team has disclosed that it is working on a new version of the CCA (BBR v2.0). The new version aims to address the issues related to BBR's high loss and unfairness in shallow buffers [5]. The algorithm adopts a cautious approach to probing that limits the maximum inflight packets and the amount of probing packets. The new modifications appear to be targeted for better fairness with large loss-based CCA flows. However, improvements in the performance of small- and medium-sized loss-based flows are needed as well. In addition to the solutions targeted for large flow fairness, the available bandwidth estimation window can be shortened to improve BBR's response to allow better completion time to arriving CUBIC bursts. Allowing improved completion time for CUBIC cross-traffic bursts in a wide range of buffer sizes while achieving comparable throughput to large CUBIC flows would facilitate seamless deployment of BBR.

In [5], it was also indicated that the research for the new BBR version includes development of a less aggressive startup. Since the intended approach adopted to achieve this is not yet specified, we can not say much about this. However, the significant reduction in the throughput of a large CUBIC flow and the utilization of the bottleneck shown here for shallow buffer bottlenecks shared between bursty BBR flows and large CUBIC flows indicate that due attention should be given to BBR's startup mechanism. Introducing the hybrid slow-start mechanism [7] to exit the exponential window increase early, could reduce the impact of BBR's aggressive startup. Throughput prediction and estimation mechanisms can be employed to regulate the exponential growth to prevent the transient saturation caused by the startup.

The link utilization of CUBIC over a bottleneck with a small buffer improves as the number of flows increases. Future work will expand the experiments in this paper by increasing the number of simultaneous CUBIC and BBR flows. By increasing the number of large CUBIC flows, a better comparison can be made between the performance of large CUBIC flows competing with medium-sized BBR flows and large CUBIC flows competing with medium-sized CUBIC flows in bottlenecks with buffer sizes less than BDP. Hock et al. [10] has indicated that multiple BBR flows can result in the creation of a persistent queue in a bottleneck. Therefore, for multiple BBR flows, the performance of small- and medium-sized flows over bottlenecks with buffer sizes less than or equal to 2 BDP is likely to be worse than the results presented in this paper. Hence, by increasing the number of large BBR flows we could ascertain if the requisite 2 BDP buffer size for fair operation between large BBR and large CUBIC flows extends to small and medium-sized CUBIC (loss-based) flows as well.

VIII. RELATED WORK

Since BBR has only been publicly available recently, there have only been few independent evaluations of the algorithm. After the release of the original BBR paper [3], subsequent deployment updates were presented in [11], [12] and [5] by the Google BBR team. Cardwell et al. [11] presented performance improvements observed when deploying BBR for Google WAN traffic. In [12], the deployment of BBR for QUIC traffic is announced. In [5], the BBR development team introduces the development of the next version BBR (BBRv2) and roughly presents the mechanisms being considered for the new version.

Deploying BBR for general Internet traffic would require thorough testing and evaluation over a range of network and traffic settings. Hock et al. [10] examined large flows and found that multi-flow BBR can cause queuing delays in large buffers and massive losses in small buffers. The high loss caused by BBR in small buffers will then force competing CUBIC flows to have an unfairly small bandwidth share. Beshay et al. [13] performed evaluations of multiple CCAs in an emulated environment based on LTE link traces where the RTT is quite small. The results of the evaluations suggest that BBR is unable to properly estimate and utilize the available link capacity in the given conditions. Ma et al. [16] present results from experiments in a cluster of servers with end-to-end RTT of flows configured through *netem*. Their results show BBR's tendency to favor flows with long RTTs over flows with short RTTs.

Chung et al. [14] conducted experiments by downloading data using different CCAs while driving and at the same time being connected to an LTE network. The results indicate that BBR maintains a low RTT while achieving comparable throughput to CUBIC. The results presented in [15], from experiments performed on traces as well as real LTE network, show that BBR is able to achieve better throughput than most proposed cellular network oriented CCAs while maintaining comparable or lower RTT than CUBIC. Experiments on BBR's performance over live cellular network experiments and LTE emulators were conducted in [17]. The results of this paper show that, with a few exceptions, BBR is able to achieve better performance in most of the live network measurement scenarios.

The aforementioned independent evaluations of BBR involve studies on the performance of the steady-state operation of BBR flows. However, the steady-state and startup mechanisms of BBR are markedly different from the steady-state and startup of the well-established loss-based CCAs. Evaluation of the interactions between the startup operation of BBR with flows of other CCAs is therefore still missing. To this end, this paper adds new insight by evaluating the impact of BBR's startup mechanism on large CUBIC flows as well as on smaller CUBIC traffic bursts. It also extends previous works on the impact of BBR's steady-state operation by evaluating its impact on CUBIC traffic bursts.

IX. CONCLUSION

This paper presents and analyzes the impact of BBR's startup and steady-state mechanisms on variable-sized flows of a widely deployed loss-based CCA, CUBIC. We use flows of different sizes to evaluate the impact of BBR's startup mechanism over a range of bottlenecks with bandwidths ranging between 5 Mbps and 70 Mbps, and buffer sizes ranging between 0.5 BDP and 8 BDP. We also evaluate the response of BBR steady-state mechanism to bursts of CUBIC flows with different sizes.

The results of our experiments indicate that an all-BBR, mixed-size traffic exhibits exceptional consistency in performance over a range of bandwidths and buffer sizes. However, over a bottleneck with a small buffer shared by a large CUBIC flow and successive BBR bursts, our results show that, due to the aggressive BBR startup mechanism, a significant reduction in the throughput of the large flow and the utilization of the bottleneck is experienced. This observation suggests that BBR can worsen the low throughput and utilization of loss-based CCAs in shallow buffers by which its development was partly motivated. In addition, it was shown that a buffer size of 2 BDP is not sufficient for small CUBIC flows competing with large BBR flows to avoid a decrease in performance resulting from BBR's steady-state mechanism. We have observed that BBR's achievement in avoiding the extended completion times of small flows resulting from bloated buffers is commendable. Thus, we are hoping that the observations in this paper would provide valuable input towards improving BBR, an algorithm that is still under active development, and has the potential and the platform for wide-scale deployment. Future work will consider available capacity estimation mechanisms that can be used by the startup mechanism of BBR as well as by other (rate-based) CCAs.

REFERENCES

[1] V. Jacobson, "Congestion avoidance and control," *SIGCOMM Comput. Commun. Rev.*, vol. 18, no. 4, pp. 314–329, Aug. 1988.

[2] S. Ha *et al.*, "CUBIC: A new TCP-friendly high-speed TCP variant," *SIGOPS Oper. Syst. Rev.*, vol. 42, no. 5, pp. 64–74, Jul. 2008.

[3] N. Cardwell *et al.*, "BBR: Congestion-based congestion control," *Queue*, vol. 14, no. 5, pp. 50:20–50:53, Oct. 2016.

[4] J. Gettys and K. Nichols, "Bufferbloat: Dark buffers in the Internet," *Queue*, vol. 9, no. 11, pp. 40:40–40:54, Nov. 2011.

[5] N. Cardwell *et al.*, "BBR congestion control: Ietf 100 update: BBR in shallow buffers." Presentation in ICCRG at IETF 100th meeting, Nov 2017. [Online]. Available: https://datatracker.ietf.org/meeting/100/materials/slides-100-iccrg-a-quick-bbr-update-bbr-in-shallow-buffers/

[6] P. Balasubramanian, "Updates on Windows TCP." Presentation in ICCRG at IETF 100th meeting, Nov 2017. [Online]. Available: https://datatracker.ietf.org/meeting/100/materials/slides-100-tcpm-updates-on-windows-tcp-00

[7] S. Ha and I. Rhee, "Taming the elephants: New TCP slow start," *Comput. Netw.*, vol. 55, no. 9, pp. 2092–2110, Jun. 2011.

[8] N. Cardwell *et al.*, "BBR congestion control." Presentation in ICCRG at IETF 97th meeting, Nov 2016. [Online]. Available: https://www.ietf.org/proceedings/97/slides/slides-97-iccrg-bbr-congestion-control-02.pdf

[9] S. Jain and G. Raina, "An experimental evaluation of CUBIC TCP in a small buffer regime," in *NCC*, Jan 2011, pp. 1–5.

[10] M. Hock, R. Bless, and M. Zitterbart, "Experimental evaluation of BBR congestion control," in *ICNP*, Oct 2017, pp. 1–10.

[11] N. Cardwell *et al.*, "BBR congestion control: An update." Presentation in ICCRG at IETF 98th meeting, Mar 2017. [Online]. Available: https://www.ietf.org/proceedings/98/slides/slides-98-iccrg-an-update-on-bbr-congestion-control-00.pdf

[12] N. Cardwell, Y. Cheng *et al.*, "BBR congestion control: IETF 99 update." Presentation in ICCRG at IETF 99th meeting, Jul 2017. [Online]. Available: https://www.ietf.org/proceedings/99/slides/slides-99-iccrg-iccrg-presentation-2-00.pdf

[13] J. D. Beshay, A. T. Nasrabadi, R. Prakash, and A. Francini, "Link-coupled TCP for 5G networks," in *2017 IEEE/ACM 25th International Symposium on Quality of Service (IWQoS)*, June 2017, pp. 1–6.

[14] S. Ma, J. Jiang, W. Wang, and B. Li, "Towards RTT fairness of congestion-based congestion control," *CoRR*, vol. abs/1706.09115, 2017. [Online]. Available: http://arxiv.org/abs/1706.09115

[15] F. Li, *et al.*, "TCP CUBIC versus BBR on the highway," in *Passive and Active Measurement*, 2018.

[16] W. K. Leong, Z. Wang, and B. Leong, "TCP congestion control beyond bandwidth-delay product for mobile cellular networks."

[17] E. Atxutegi *et al.*, "On the use of TCP BBR in cellular networks," *IEEE Communications Magazine*, 03 2018.