Impact of Background Traffic on the BBR and CUBIC Variants of the TCP Protocol

Daniela M. Casas-Velasco, Fabrizio Granelli, *Senior Member, IEEE*, and Nelson L. S. da Fonseca, *Senior Member, IEEE*

***Abstract*—This letter analyzes the performance of the TCP CUBIC and TCP BBR protocols in the presence of background traffic. The analysis is performed via emulation using actual TCP implementations and considering high capacity end-to-end data connections and different connection durations (e.g., mouse and elephant flows). The results indicate that the BBR protocol produces higher throughput and fairness than the CUBIC protocol.**

***Index Terms*—Congestion control, TCP, BBR, CUBIC.**

I. Introduction

The TRANSMISSION Control Protocol (TCP) represents the most diffused transport protocol on the Internet and is currently used even for video-on-demand. For this reason, analyzing the performance of the different variants of congestion control introduced for current and future networks is of utmost importance. Two TCP variants have attracted the attention of scientists and practitioners in recent years: the TCP CUBIC and TCP BBR.

The TCP CUBIC protocol [1] employs a congestion control algorithm of currently widespread use on the Internet. Based on congestion resolution through response to packet losses, the CUBIC protocol is an alternative to those protocols which adopt the Additive-Increase / Multiplicative-Decrease (AIMD) congestion control mechanism such as the well-known New RENO version of TCP. Unlike its predecessors, the CUBIC protocol modifies the linear window growth function by using a cubic function to improve the scalability of the TCP over networks with large Bandwidth Delay Product (BDP).

Moreover, during steady-state, the CUBIC protocol aggressively increases the window size when it is far from the saturation point and more slowly when it is closer to that point. The key feature of the CUBIC protocol is that its window growth depends only on the real-time interval between two consecutive congestion events. Thus, the window growth becomes independent of Round Time Trip (RTT). This feature allows CUBIC-regulated flows competing in the same bottleneck to have approximately the same window size independent of their RTT, thus achieving good RTT fairness.

TCP Bottleneck Bandwidth and Round-trip propagation time (BBR) [2], [3] is a congestion control mechanism proposed by Google and disclosed as a plug-and-play solution for increasing throughput in unstable connections. Instead of responding primarily to packet loss, the algorithm includes real-time measurements of delivery rates and RTTs in its decisions. Based on all the parameters considered, a new measure called pacing gain is calculated; it is derived to define the values of the intervals between sending packets and the congestion window. Due to its approach, the BBR protocol does not instantly respond to packet loss since it is dependent on the pacing rate variation. In this way, the BBR protocol controls the number of packets in transit rather than directly controlling the window size. This approach brings benefits such as increased flow rates in high-capacity networks and reduced sensitivity to random packet losses over wireless networks.

Furthermore, the BBR protocol differs from other algorithms in that it actively avoids network congestion due controlling the number of packets in transit. On the other hand, algorithms such as that of CUBIC increase the number of packets in transit until the bottleneck is reached, which can be a problem when too many packets end up queued in large router buffers, a state known as *bufferbloat*. A state machine is used to vary the BBR control parameters. It acts to maximize throughput, minimize latency and ensure fair bandwidth sharing.

Several congestion control mechanisms prior to that of BBR were primarily loss-based. In these cases, the data rate was only adjusted when the bottleneck buffer started overflowing and packets started being dropped. These algorithms made sense during the inception of TCP since the buffer size was slightly more significant than the links BDPs. However, the increased buffer sizes offered in current forwarding devices result in congestion control algorithms that report congestion scenarios after a prolonged period of queueing, which can result in high packet loss ratios. For facing such congestion scenarios, the BBR protocol looks at the bottleneck bandwidth of the paths and estimates the RTT to determine the existence congestion in a network; this results in accurate and immediate congestion detection. The BBR protocol, thus, provides better utilization and end-to-end network bandwidth.

This letter provides an analysis of the performance of the TCP CUBIC and TCP BBR protocols in the presence of background non-responsive UDP traffic, leading to a better

Manuscript received 22 June 2022; accepted 10 July 2022. Date of publication 18 July 2022; date of current version 26 August 2022. This work was supported in part by the Sao Paulo Research Foundation (FAPESP) under Grant 2015/24494-8. The associate editor coordinating the review of this article and approving it for publication was F. Y. Li. *(Corresponding author: Nelson L. S. da Fonseca.)*

Daniela M. Casas-Velasco and Nelson L. S. da Fonseca are with the Institute of Computing, University of Campinas, Campinas 13083-872, Brazil (e-mail: danielac@lrc.ic.unicamp.br; nfonseca@ic.unicamp.br).

Fabrizio Granelli is with the Department of Information Engineering and Computer Science, University of Trento, 38122 Trento, Italy (e-mail: fabrizio.granelli@unitn.it).

Digital Object Identifier 10.1109/LNET.2022.3191994

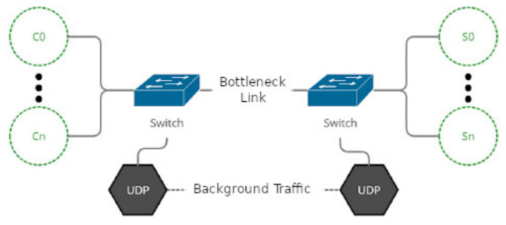


Fig. 1. Dumbbell topology.

TABLE I
SETTINGS FOR EXPERIMENTAL SCENARIOS

Parameter	Value
Bottleneck link capacity	100, 300 Mbps
Buffer size	1 BDP
RTT	50ms
Interflow gap	0.1 ms
mice flows	300 KB
elephant flows	30 MB
Ratio between flows	20% *elephant* 80% *mice*
UDP *background* traffic	10, 20, 30 % of the link capacity
TCP CUBIC/BBR flow distribution	100/0 0/100 25/75 50/50 75/25 % of the total flows

understanding of how such a scenario impacts on the two modern TCP variants. The analysis is performed using the actual implementation of both versions of TCP in the Linux Kernel within Virtual Machines. The Virtual Machines are connected by an emulated network using the widely known Mininet network emulator.

The achieved results provide a realistic picture of the actual performance figures to be expected. The main points of novelty of this letter include:

- An emulation-based in-depth performance evaluation of the recent versions of TCP in the presence of background unresponsive traffic. To the authors' knowledge, this is the first realistic evaluation in such a scenario.
- Specific analyses involving high end-to-end bottleneck bandwidth (100 Mbps and more).
- Fairness analysis considering mouse and elephant flows.

II. EVALUATION

A. Network Environment and Prototype

The environment used in the experiments is described below. We deployed the Dumbbell topology; it includes a bottleneck link between the sender and the receiver. For evaluation, we emulated the scenario represented in Figure 1 with TCP clients on the left side, generating traffic destined for their peers at the other end of the bottleneck link. There is also User Datagram Protocol (UDP) background traffic flow in the same direction as the TCP traffic. Tests were run on a VM with 8GB of RAM, 4 dedicated vCPUs, and Linux Ubuntu 18.04.1 with 64 bits kernel 4.15.0-38. We used the *iPerf3* tool to generate the TCP and UDP traffic with Mininet to emulate the network.

B. Network Parameters

The experiments performed employ the values of the parameters described in Table I. Experiments were carried out with both the BBR and CUBIC protocols for different percentages of connections in the bottleneck links of 100Mbps and 300Mbps. All combinations of parameter settings shown in the table were explored in the experiments. Thirty repetitions of each experiment were performed for each parameter set to generate a confidence interval for the mean values. We fixed some parameters' values, such as buffer size, interflow gap, and RTT, which configured the communication for all connections in all experiments. The parameters varied in the experiments were i) link bandwidth variation, which aimed to assess the impact of UDP background traffic with different bandwidth availability, ii) variations in the proportions of small and large flows (mice and elephants), iii) percentage of UDP background traffic, not subject to mechanisms of congestion control, and iv) the percentage of TCP connections using CUBIC and BBR. Specifically, five different percentage distributions were run. Tests with the TCP CUBIC protocol only; with TCP BBR protocol only, and three ratios of flow distributions: flows equally distributed between the two protocols (50% CUBIC, 50% BBR), predominantly BBR (75% BBR, 25% CUBIC), and predominantly CUBIC (25% BBR, 75% CUBIC).

C. Performance Metrics

The metrics used to evaluate the performance of the CUBIC and BBR protocols were the average connection throughput, link utilization, and fairness measured by Jain's fairness index. It accounts for the fairness of a set of values with n users. x_i is the throughput for the *ith* connection. The result ranges from $1/n$ (worst case) to 1 (best case), and it is maximum when all users receive the same allocation.

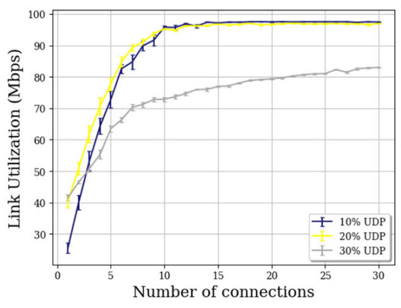
D. Performance Analysis

Results are presented for two link capacities (i.e., 100Mbps and 300Mbps). For every experiment result, confidence intervals were plotted, with a 90% confidence level. Each point on the graph represents the average of the experiments' results.

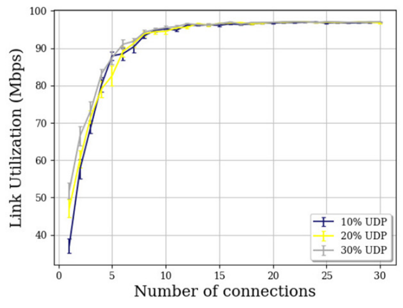
Utilization and Average Throughput for a 100Mpbs link

1) 100 Mbps - 100% CUBIC: Figures 2a and 2d show the link utilization and average throughput achieved employing only the CUBIC protocol. When the UDP background traffic is 10%, and 20% of the link capacity, the results indicate that the link reaches 95% utilization with only having ten connections and achieves a 97% utilization after 15 connections. In contrast, when the UDP background traffic is 30% of the link capacity, the link utilization increases slowly with an increase in the number of connections, with the link achieving 72% and 84% utilization for ten and 30 connections, respectively.

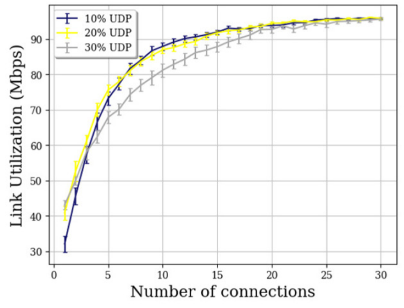
The average throughput decreases with the increase in the number of connections. When percentages of UDP background traffic of 10% and 20% of the link capacity, the average throughput is 17 Mbps, 6Mbps, and 3,8Mbps for one, ten, and 30 connections, respectively. When the UDP background traffic constitutes 30% of the link capacity, however, the average throughput for one and ten connections decreases to 9Mbps and 4,2Mbps, respectively, remaining the same for 30 connections.



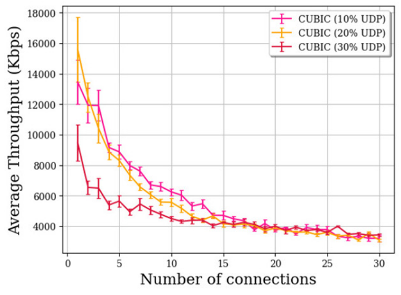
(a) 100% CUBIC



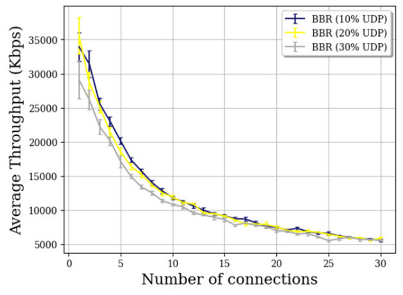
(b) 100% BBR



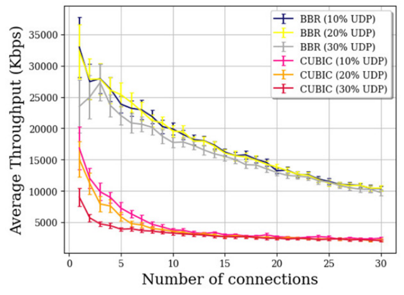
(c) 50% CUBIC and 50% BBR



(d) 100% CUBIC



(e) 100% BBR



(f) 50% CUBIC and 50% BBR

Fig. 2. Results for a 100 Mbps link. Figures (a), (b), and (c) show the link utilization when using CUBIC and BBR in different proportions. Likewise, Figures (d), (e), and (f) show the average throughput achieved by the connections using the CUBIC and BBR protocols.

2) 100 Mbps - 100% BBR: Figures 2b and 2e depict the link utilization and average throughput. The results show that link utilization increases with the number of connections regardless of the UDP background traffic. For instance, a single connection starts utilizing 36% of the link with 10% UDP background traffic but achieves at least 49% when the UDP traffic increases to 20% and 30%. When the link has ten connections, the observed utilization is approximately the same whatever the percentage of UDP background traffic (95.5%), reaching 97% after 15 connections. Results also show that the average throughput behaves similarly for all percentages of UDP background traffic. When the link has two to ten connections, the average throughput per connection drops from 18Mbps to 12Mbps. Furthermore, for more than ten connections, the link surpasses 95% of utilization, which causes the average throughput to decrease to 5Mbps.

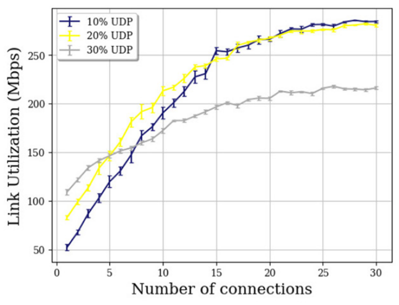
In contrast to when all connections use CUBIC, the utilization and average throughput were similar for the three different percentages of UDP traffic. This behavior reveals the ability of the BBR protocol to take advantage of the available link bandwidth and further exploit a significant amount of link capacity, even when sharing that link with traffic with no congestion control, such as UDP.

3) 100 Mbps - 50% BBR and 50% CUBIC: Figures 2c and 2f show the link utilization and average throughput achieved when the 50% of the connections employ the BBR protocol and 50% of them employ the CUBIC protocol. The results show that the utilization and throughput improve when compared to exclusively use of the CUBIC protocol. Indeed, the utilization observed with 30% of UDP background traffic increases rapidly and achieves at least 90% link utilization after 17 connections and 95% after 25 connections. The connections using the CUBIC protocol do not significantly differ from the previous case. BBR Connections achieve higher average throughput than those using CUBIC due to the greater ability of the BRR protocol to deal with congestion. For instance, the maximum throughput achieved by the BBR connections is approximately twice that using the CUBIC protocol. Instead of responding to packet loss, the BBR protocol uses real-time measurements of delivery rates and RTTs. Moreover, the CUBIC protocol increases the number of packets in transit until the bottleneck is reached, implying a lower average throughput of the connections.

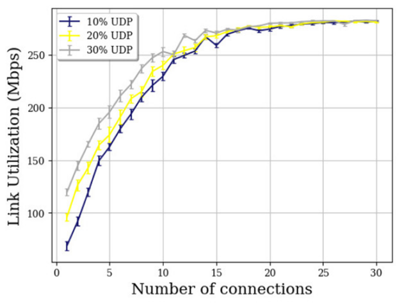
Utilization and Average Throughput for a 300Mpbs link

1) 300 Mbps - 100% CUBIC: Figures 3a and 3d show the link utilization and average throughput achieved when employing only the CUBIC protocol in a link with a capacity of 300Mbps, respectively. The results demonstrate that the CUBIC protocol has a similar growth in utilization for 10% and 20% of UDP background traffic, with a smaller growth for 30% of UDP traffic. Indeed, for 10% and 20% of UDP background traffic, the link reaches 95% utilization with 20 connections and 97% with 28 connections. With 30% of UDP traffic, the link utilization for 20 connections is 68.3% and for 28 connections it is 73%. The described behavior is similar to that observed for a link with a capacity of 100Mbps.

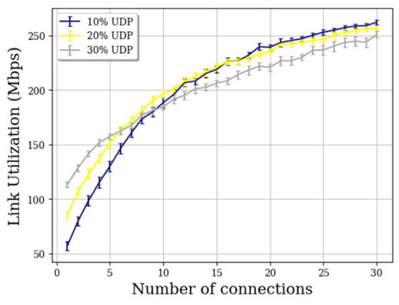
The average throughput decreases with the increase in the number of connections. There are differences in the achieved average throughput values depending on the percentage of UDP background traffic. For instance, the average throughput with ten connections is 13Mbps, 11Mbps, and 7 Mbps for 10%, 20%, and 30% UDP traffic. Moreover, when the utilization is 95% with 21 connections, the average throughput decreases to 8.5Mbps, 8Mbps, and 5Mbps for 10%, 20%, and 30% of UDP traffic, respectively.



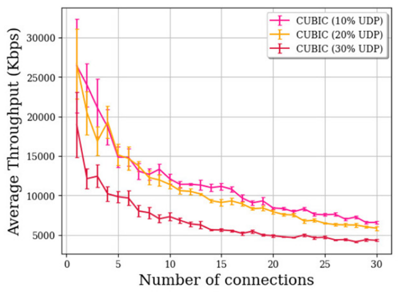
(a) 100% CUBIC



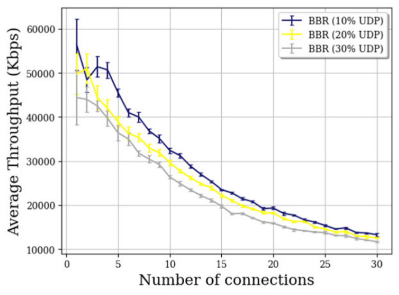
(b) 100% BBR



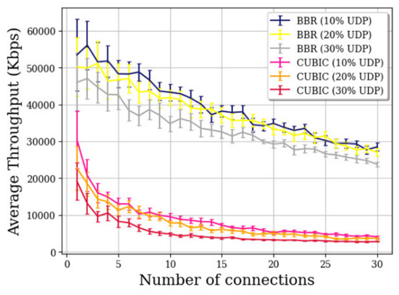
(c) 50% CUBIC and 50% BBR



(d) 100% CUBIC



(e) 100% BBR



(f) 50% CUBIC and 50% BBR

Fig. 3. Results for a 300 Mbps link. Figures (a), (b), and (c) show the link utilization when using CUBIC and BBR in different proportions. Likewise, Figures (d), (e), and (f) show the average throughput achieved by the connections in relation to the use of the CUBIC and BBR protocols.

The results observed in a link with a capacity of 100Mpbs and the CUBIC protocol used by all the connections show that the utilization grows at a slower pace. Indeed, with a greater link capacity, the maximum utilization ($\approx$ 97%) is achieved after 20 connections, while equivalent utilization is achieved after ten connections in a link of 100Mbps. Therefore, the CUBIC protocol is slow to take advantage of available resources since it only modifies its congestion mechanism in relation to packet loss and it waits until reaching packet loss to react.

2) 300 Mbps - 100% BBR: Figures 3b and 3e show the link utilization and average throughput achieved when employing only the BBR protocol in a link with a capacity of 300Mbps. The results show that the BBR connections present similar link utilization growth for all UDP background traffic percentages by maintaining a constant difference until reaching the maximum utilization with 20 connections. With 20 connections, the link reaches 95% utilization.

The results also show that the average throughput when the link reaches 95% utilization is 19Mbps for 10% and 20% of UDP background traffic. For 30% of UDP traffic, the throughput is 16Mbps. For up to 30 connections, the average throughput decreases to approximately 12Mbps for the three percentages of UDP traffic.

The average throughput observed after the maximum utilization is greater than that observed when using only the BBR protocol.

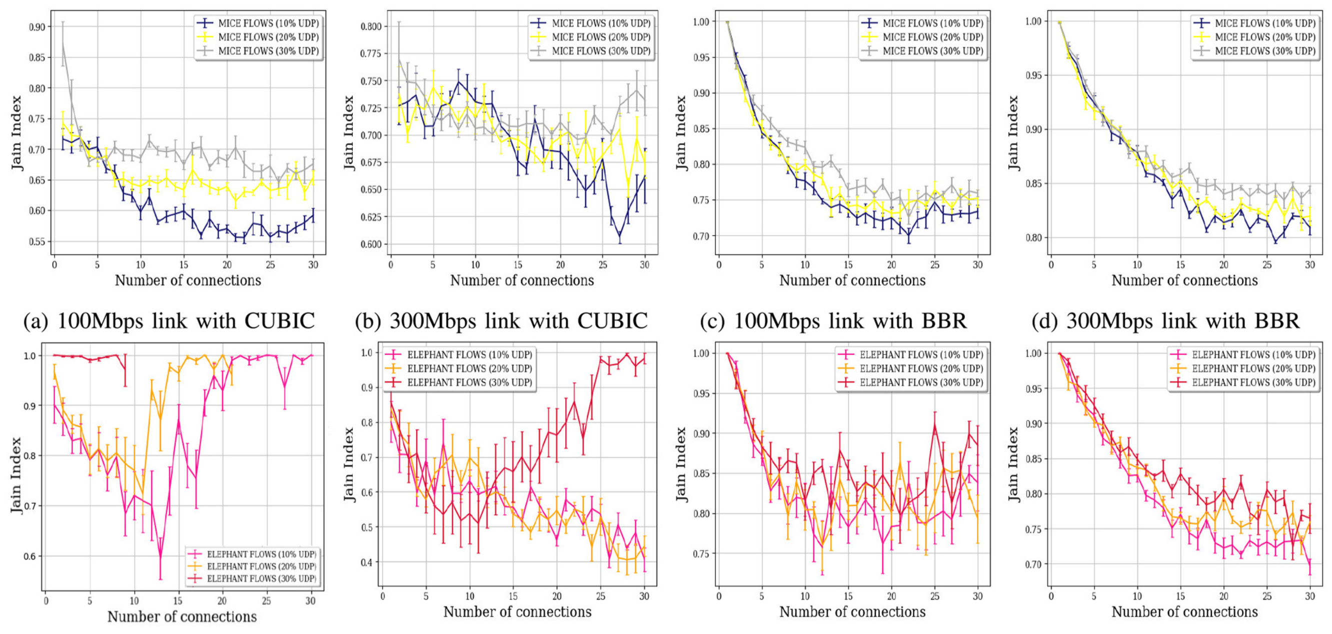
The connections using only the BBR protocol in a 300Mbps link lead to maximum utilization after ten connections, regardless of the percentage of UDP traffic, a result similar to that using a 100Mbps link. The BRR protocol controls the number of packets in transit instead of directly controlling the congestion window size; moreover, it uses multiple real-time measurements to define that number of packets. Therefore, it can rapidly take advantage of the available resources and achieve high utilization.

3) 300 Mbps - 50% BBR and 50% CUBIC: Figures 3c and 3f show the link utilization and average throughput achieved when 50% of the connections employ the BBR protocol and the other 50% use the CUBIC protocol over a link with a capacity equal to 300Mbps. The results show that the utilization curves maintain the format previously observed for a link of 100Mbps. The utilization surpasses 80% in cases with 10% and 20% UDP traffic and with 25 connections; when the number of connections increases to 30, the utilization reaches 80% for 30% of UDP traffic. The shapes of the average throughput curves are also maintained equivalent to those for a link of 100Mbps. The absolute values observed for the three percentages of UDP traffic in relation to the connections employing the BBR protocol vary. However, the average throughput drops to 25Mbps when the number of connections reaches 30 in all the cases.

Fairness in relation to flow type

The following results show the Jain's index measured for different percentages of UDP background traffic.

4) 100% CUBIC: Figures 4e and 4f show Jain's index values as measured in relation to mouse and elephant flows when for CUBIC connections. In a link with a capacity of 100Mpbs, Jain's index values for mouse flows decrease rapidly as a function of the percentage of UDP traffic. In contrast, when the link capacity is 300Mbps, Jain's index's behavior is different. From 1 to 25 connections, the mouse flows lead to a decrease in Jain's index value for all the percentages of UDP traffic. On the other hand, the behavior observed for the elephant flows for 100 and 300 Mbps link capacities varies in relation to the UDP traffic. For example, there is an increase in Jain's index once



(a) 100Mbps link with CUBIC (b) 300Mbps link with CUBIC (c) 100Mbps link with BBR (d) 300Mbps link with BBR

(e) 100Mbps link with CUBIC (f) 300Mbps link with CUBIC (g) 100Mbps link with BBR (h) 300Mbps link with BBR

Fig. 4. Fairness results regarding the type of flow using only CUBIC or BBR.

the number of connections in the link surpasses ten connections for the 100Mbps link with 20% and 30% of UDP traffic. Similar behavior is observed for the 300Mbps link when it holds only 10% of UDP traffic. In contrast, for a link capacity of 300Mbps and the percentage of UDP traffic of 20% or 30%, Jain's index value decreases after 15 connections.

These results depict that the CUBIC protocol prioritizes elephant flows in the scenario that the link holds a significant percentage of UDP traffic (30%); this indicates the effort to provide fairness for bigger packets in the network. Although Jain's index values observed for mouse flows tend to decrease, its minimum observed values are still greater than 0.7, which is still a good fairness value.

5) 100% BBR: Figures 4g and 4h show Jain's index measured in relation to mouse and elephant flows when all connections use only the BBR protocol. Results show that Jain's index value tends to decrease regardless of the UDP traffic and flow type, whichever the link capacity. The Jain's index regularly decreases for the mouse flows, starting with a value of 1.0 for one connection and reaching around 0.75 and 0.85 for the 100 and 300Mbps links after 15 connections. For the case of elephant flows and the 100Mbps link, the decrease is irregular, presenting ups and downs after five connections, with an initial Jain's index value of 1.0, yet barely reaching around 0.85 with 30 connections. In contrast, for a link of 300Mbps, the decrease is regular, with around 0.75 for 30 connections. These results show that the BBR protocol leads to a fair share of link resources; therefore, its congestion control is advantageous whatever the percentages of UDP traffic and regardless the flow type.

III. Conclusion

The results showed that the BBR protocol achieves higher average throughput than the CUBIC protocol regardless of the load of the UDP background traffic. Moreover, the higher the background traffic load, the lower is the increase in link utilization achieved by the CUBIC protocol. Conversely, the BBR protocol packet loss does not trigger any reaction since it uses real-time measurements of delivery rates and RTTs. Moreover, BBR offers greater fairness than CUBIC regardless of the background traffic and flow type.

Acknowledgment

The authors would like to thank G. L. H. Rincão and W. G. da Cruz for the support in the emulation experiments.

References

[1] S. Ha, I. Rhee, and L. Xu, "CUBIC: A new TCP-friendly high-speed TCP variant," *ACM SIGOPS Oper. Syst. Rev.*, vol. 42, no. 5, pp. 64–74, 2008.

[2] N. Cardwell, Y. Cheng, C. S. Gunn, S. H. Yeganeh, and V. Jacobson, "BBR: Congestion-based congestion control," *Commun. ACM*, vol. 60, no. 2, pp. 58–66, 2017.

[3] N. Cardwell *et al.*, "BBR congestion control work at Google: IETF 101 update," in *Proc. ICCRG IETF 102nd Meeting*, 2018, pp. 1–44.