6D-04

BBR+: Improvement of Congestion-Based Congestion Control for Deep Buffer Link

Satoshi Utsumi† Shi Weizhi‡ Norio Suzuki† Salahuddin Muhammad Salim Zabir§

†Fukushima University ‡Chongqing University of Technology §National Institute of Technology, Tsuruoka College

1 Introduction

In September 2016, Google presented a new congestion-based congestion control, which they named as TCP BBR (Bottleneck Bandwidth and Round-trip propagation time [1]), and available in Linux kernel version 4.9 or later. TCP BBR is a high perfoamance congestion control, which maximizes utilization of the bottleneck link, avoiding bufferbloat problem [2]. However, TCP CUBIC is known as the default congestion control in Linux, in Android, and in iOS [3]. The paper [4] points out the performance degradation problem when TCP BBR flows co-exist with TCP CUBIC flows in large buffer links, in which bufferbloat is more probable. In this paper, we propose a new congestion-based congestion control TCP BBR+ which has high performance while co-existing with TCP CUBIC, and evaluate its performance.

2 Problems of BBR with CUBIC in Deep Buffer Link

2.1 Sequential Timeout Problem

TCP CUBIC continues to increase its congestion window size until some packets are lost. Then, if the buffer size in the bottleneck link is large, the queue size in the link grows largely by TCP CUBIC flows. When the queue size grows largely, queueing delay also becomes large. If $RTprop$, the Round-Trip propagation time (minimum RTT), is updated regularly, the estimated value of RTprop is large. Next, BDP, which is the bandwidth delay product, is updated as $BDP = BW * RTprop$. then, BDP is also estimated largely. In ProbeBW state, $cwnd$,the congestion window size, is calculated as $cwnd = 2 * BDP$. Hence the cwnd is calculated to grow quite big. As the result, many packets in the TCP CUBIC flow will be lost. Because of many packet losses, the sequential timeouts will be occur. The sequential timeouts decrease total throughput of the bottleneck link.

2.2 Pressured Throughput Problem by CUBIC

As described in section 2.1, TCP CUBIC continues increasing the congestion window size until some packets are lost. On the other hand, TCP BBR stabilizes its congestion window size, if $RTprop$ is constant. Then, TCP BBR flows may lose to TCP CUBIC flows, that is, the throughput of TCP BBR flows may decrease.

3 BBR+: Improvement of BBR for Deep Buffer Link

In this section, we propse a new congestion-based congestion control TCP BBR+, (i) which has large utilization on the bottleneck link avoiding bufferbloat when no TCP CUBIC flow is in the bottleneck link, (ii) which avoids decreasing throughput also when its flows are with some TCP CUBIC flows in the bottleneck link with large buffer.

TCP BBR+ introduces a new parameter $NewBDP$, which is $NewBDP = BW * RTT$, where RTT is not Round-trip propagation time but recent smoothed RTT. If we define $cwnd = NewBDP$, feedback loop may occur. So, we define $cwnd = min(2 * BDP, \alpha * NewBDP) < NewBDP$ in ProbeBW state, here , $\alpha = 0.98$. Because of this calculation, TCP BBR+ can avoid the sequential timeouts by too large $cwnd$ in section 2.

Also, when queue size in the bottleneck link is large, RTT also becomes large accordingly. $NewBDP$ varies linearly with respect to RTT. The $cwnd$ of TCP BBR+ is updated using $NewBDP$ as above. Because the $cwnd$ of TCP BBR+ also becomes large when TCP CUBIC flows increase the queue size in the bottlenck link, TCP BBR+ flows does not lose to TCP CUBIC flows.

4 Performance Evaluation

In this section, we evaluate the performance when TCP BBR+ co-exists with TCP CUBIC, using the network emulator Dummynet. Fig. 1 shows the net-

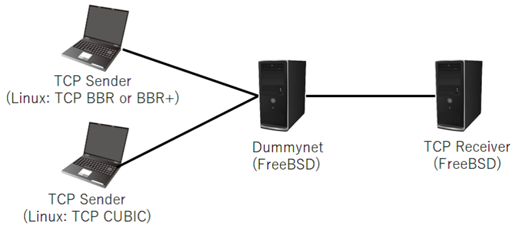

Figure 1: Network Topology

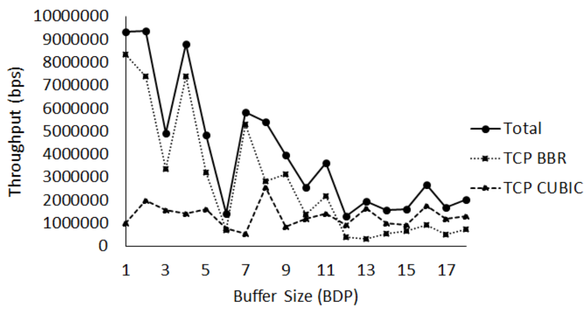

Figure 2: TCP BBR vs. TCP CUBIC [4]

work topology. Table 1 shows the parameters of TCP sender and Dummynet. The transmission time from TCP sender is 120 sec. We measure the throughput of TCP BBR+, TCP CUBIC, and the total taking the average over 10 runs for each buffer size at the bottleneck link on the above experiment.

Fig. 2 shows the results when TCP BBR co-exists with TCP CUBIC in the topology of Fig. 1 [4]. As shown in Fig. 2 and as described in section 2, when the buffer size of the bottleneck link is large, the total throughput is small. Also, when the buffer size is large, the throughput of TCP BBR is smaller then that of TCP CUBIC.

Fig. 3 shows the results when TCP BBR+ co-exists with TCP CUBIC in the topology of Fig. 1. Unlike Fig. 2, also when the buffer size in the bottleneck link is large, the total throughput is large. When the buffer size is small, like TCP BBR, the throughput of TCP BBR+ is quite larger than that of TCP CUBIC. When the buffer size is large, unlike TCP BBR, TCP BBR+ does not lose by TCP CUBIC, although the throughput of TCP BBR+ is close to that of TCP CUBIC.

5 Conclusion

TCP BBR is a high performance TCP congestion control, which maximizes the utilization of the bottleneck link and avoids the bufferbloat. In the bottleneck link with deep buffer probable to be bufferbloat, the throughput of TCP BBR co-existing with TCP

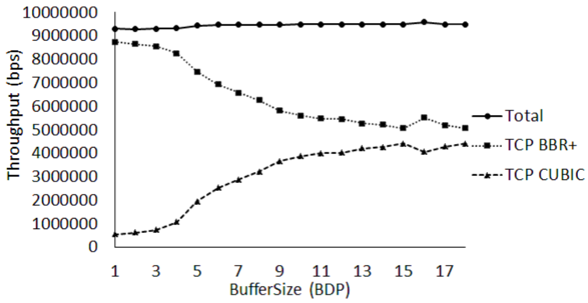

Figure 3: TCP BBR+ vs. TCP CUBIC

Table 1: Experimental Prameters

Capacity in Bottleneck Link	10 (Mbps)
RTprop (Roud-Trip propagation time)	40 (ms)
Buffer Size in Bottleneck Link	1 ∼ 18 (BDP)
Maximum Congestion Window Size	100 (MBytes)
Data Transmission Time	120 (sec)

CUBIC decrease because of the sequential timeouts. In this paper, we propose a new high perfromance congestion-based congestion control TCP BBR+, which can yield large throughput even when co-existing with TCP CUBIC in the bottleneck link with deep buffer. We found that TCP BBR+ can get high performance in the network with the deep buffer co-existing with TCP CUBIC, not decreasing the total throughput.

References

[1] N. Cardwell, Y. Cheng, C. S. Gunn, S. H. Yeganeh, and V. Jacobson, "BBR: Congestiono-based Congestion Control," ACM Queue, vol. 14, issue 5, Sep./Oct. 2016.

[2] V. Cerf, V. Jacobson, N. Weaver, and J. Gettys, "BufferBloat: What's Wrong with the Internet?," ACM Queue, vol. 9, issue 12, Dec. 2011.

[3] Y. Hozumi, S. Utsumi, and S. M. S. Zabir, "Performance Evaluation of TCP Congestion Control on iOS: Estimation of Algorithm and Parameters," Forum on Information Technology (FIT2017), Tokyo, Japan, Sept. 2017 (in Japanese).

[4] H. Yashiro, S. Weizhi, and S. Utsumi, "Co-existance Problem of BBR and CUBIC in Deep Buffer Link," The 80th National Convention of IPSJ, Tokyo, Japan, March 2018 (in Japanese).