Improvement of Bandwidth Fairness between TCP BBRv2 and CUBIC

Hyo-Seop Cho
School of Electronic and Electrical Engineering
Kyungpook National University
Daegu, Korea
yg011175@knu.ac.kr

Geon-Hwan Kim
School of Electronic and Electrical Engineering
Kyungpook National University
Daegu, Korea
kgh76@ee.knu.ac.kr

You-Ze Cho*
School of Electronic and Electrical Engineering
Kyungpook National University
Daegu, Korea
yzcho@ee.knu.ac.kr

***Abstract*— Google proposed Bottleneck Bandwidth Round-trip propagation time version2 (BBRv2) congestion control algorithm to solve the bandwidth fairness problem of the BBRv1. The bandwidth fairness of BBRv2 depends on the Bandwidth Delay Product (BDP). When the bottleneck buffer is less than 2BDP, BBRv2 improves bandwidth unfairness in bandwidth sharing with loss-based congestion control algorithm such as CUBIC over BBRv1. However, when the bottleneck buffer exceeds 2BDP, BBRv2 has bandwidth fairness issues. In this paper, we proposed F-BBRv2 to solve the bandwidth fairness problem of BBRv2 by using the min_RTT value and adaptive pacing_gain method.**

Keywords*— *TCP congestion control algorithm, BBRv2, CUBIC, Bandwidth Fairness.

I. INTRODUCTION

In 2016, Google proposed a new concept for congestion control algorithms called Bottleneck Bandwidth Round-trip propagation time (BBRv1) [1]. The BBRv1 congestion control algorithm measures maximum bottleneck bandwidth and minimum delay. BBRv1 aims at maximum transmission speed and low queue delay through the measured value. When BBRv1 and loss-based congestion control algorithm share the same bottleneck link, the two algorithms do not share the bottleneck bandwidth fairly. To address the problem of BBRv1, Google proposed BBRv2 with packet loss and Explicit Congestion Notification (ECN) rate applied in 2019.

BBRv2 determines the transmission amount by measuring the bottleneck link bandwidth and the minimum Round-Trip Time (min_RTT) in the same way as BBRv1. Additionally, BBRv2 considers packet loss and ECN rate to share bandwidth fairly with other congestion control algorithms. If the packet loss is more than 2% or the ECN is more than 50%, BBRv2 does not increase the transmission rate anymore. Due to algorithm improvement, BBRv2 uses the bottleneck link bandwidth more fairly than BBRv1.

BBRv2 bandwidth fairness depends on the size of the bottleneck buffer Bandwidth Delay Product (BDP). In a bottleneck buffer environment of 2BDP or less, BBRv2 flow and loss-based congestion control e.g., CUBIC [2] flow use the bottleneck link bandwidth fairly. However, when the bottleneck buffer is more than 2BDP, BBRv2 flow and CUBIC flow unfairly use the bottleneck bandwidth. In this paper, we propose F-BBRv2 to address the unfairness of BBRv2 bandwidth.

II. ADAPTIVE PACING GAIN METHOD

When the BBRv2 and CUBIC flow share the same bottleneck bandwidth and the bottleneck buffer is 2BDP or more, the bottleneck bandwidth fairness problem occurs due to the algorithm`s different operating characteristics. BBRv2 fills the bottleneck buffer up to 2BDP in an environment where packet loss is absent. However, CUBIC fills the bottleneck buffer until a loss occurs. These different operating characteristics cause the CUBIC flow to more aggressively occupy bottleneck bandwidth than BBRv2 flow.

Additionally, if only the BBRv2 flow uses bottleneck bandwidth, the min_RTT estimated value in BBRv2 will not exceed 2*link delay, because it will only fill the bottleneck buffer up to 2BDP. However, when the bottleneck bandwidth is shared with the CUBIC flow and the bottleneck buffer size is 2BDP or more, long queue delay is created in the bottleneck buffer by CUBIC flow. The long queue delay accumulated in the bottleneck buffer causes the min_RTT of BBRv2 to be measured high.

In this paper, we propose Fair-BBRv2 (F-BBRv2) to address the issue that the bottleneck bandwidth of BBRv2 flow and CUBIC flow cannot be used fairly.

F-BBRv2 adds two steps to the existing BBRv2 to improve bandwidth fairness. First, F-BBRv2 detects that it shares a bottleneck bandwidth with the CUBIC flow when min_RTT exceeds 2*link delay. The existing BBRv2 left 15% headroom to increase fairness with other flows [3]. However, headroom reduces the transmission rate and exacerbates the problem of fairness in the CUBIC flow. Second, F-BBRv2 improves bandwidth unfairness with the CUBIC flow by increasing the transmission rate of the probe bandwidth phase by 15%.

III. EXPERIMENT RESULTS

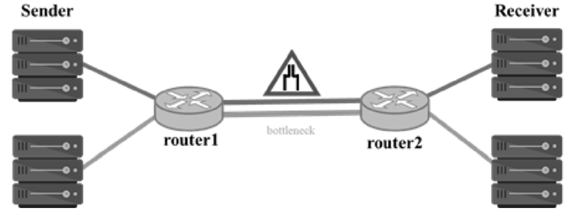
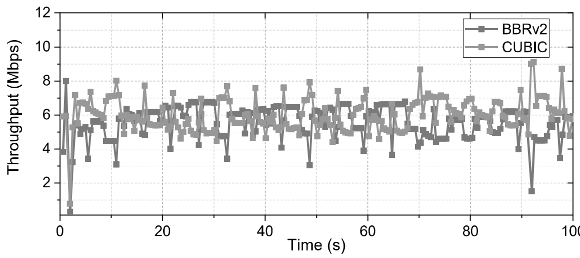
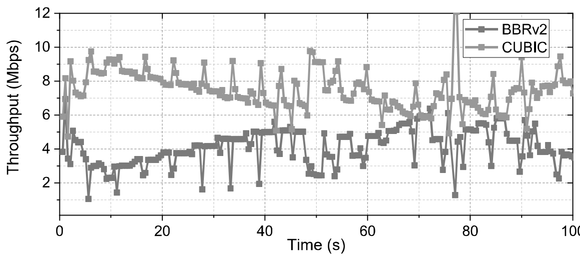


Fig. 1. Dumbbell topology

Figure 1 shows we used Network Simulator-3 (NS-3) to construct a dumbbell topology where multiple flows share the same bottleneck link. Access link bandwidth and bottleneck bandwidth are 12 Mbps. Each link delay is 15 ms and queue management used tail-drop. The experiment time is 100 s.



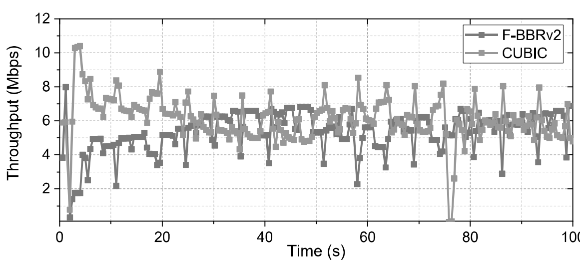
(a) buffer size < 2BDP



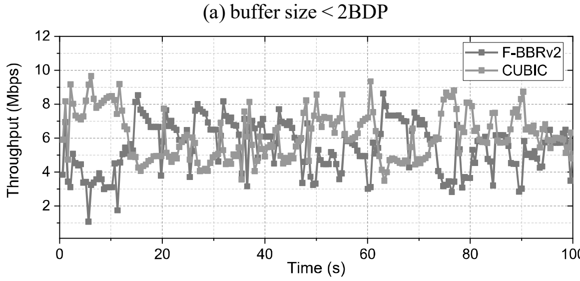
(b) buffer size > 2BDP

Fig. 2. Throughput of BBRv2 flow and CUBIC flow according to the bottleneck buffer size.

Figure 2 shows the throughput over time according to the bottleneck buffer size of BBRv2 flow and CUBIC flow. In Figure 2(a), the average throughput of the BBRv2 flow is 5.4 Mbps and the average throughput of the CUBIC flow is 6.1 Mbps. As a result, when the bottleneck buffer is less than 2BDP, the two flows share the bottleneck bandwidth fairly. However, the average throughput of the two flows is different in environments where the bottleneck buffer is 2BDP or more. Figure 2(b) indicates that the CUBIC flow has 3 Mbps higher average throughput than the BBRv2 flow.

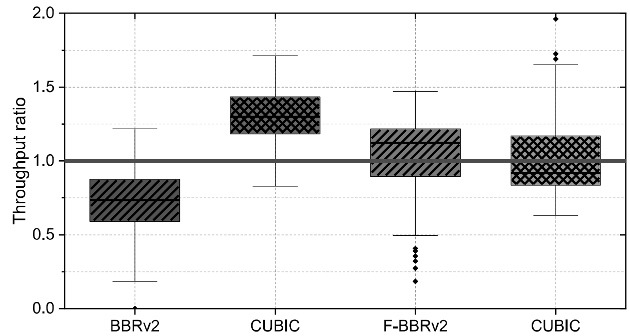


(a) buffer size < 2BDP



(b) buffer size > 2BDP

Fig.3. Throughput of F-BBRv2 flow and CUBIC flow according to the bottleneck buffer size.



(a) (b)

Fig. 4. Throughput ratio

Figure 3 shows the change in throughput for F-BBRv2 and CUBIC flows over time. The F-BBRv2 flow and the CUBIC flow share the bottleneck bandwidth fairly in Figure 3(a). Additionally, Figure 2(b) and 3(b) show that the F-BBRv2 flow share the bottleneck bandwidth more fairly than the BBRv2 flow.

Figure 4 indicates that throughput ratio of each flows. The throughput ratio is calculated as follows. Throughput ratio = each flow throughput / (bottleneck link bandwidth / number of flows). As the box chart approaches 1.0, it can be seen that the bottleneck bandwidth is being used fairly.

The two box charts in Figure 4(a) are located far from 1.0. In other words, the two flows do not share bottleneck bandwidth fairly. However, Figure 4(b) shows that the two box charts are close to 1.0 and the two flows use bandwidth fairly. Therefore, the F-BBRv2 improved and outperformed the bottleneck bandwidth fairness issue more than the BBRv2.

IV. Conclusion

There is a bandwidth fairness problem when BBRv2 flow shares bandwidth with CUBIC flow and the bottleneck buffer size is 2BDP or larger. In this paper, we propose F-BBRv2 as a solution to the bandwidth fairness problem of BBRv2. F-BBRv2 is 5.8 Mbps for both flows, regardless of the bottleneck buffer size. Therefore, F-BBRv2 has improved bandwidth fairness than BBRv2. In the future, we plan to experiment with various congestion control algorithms.

Acknowledgment

This research was supported in part by Basic Science Research Program through the National Research Foundation of Korea (NRF) funded by Ministry of Education (No. NRF-2018R1A6A1A03025109) and by National Research Foundation of Korea (NRF) grant funded by the Korea government (MSIT) (No. NRF-2019R1A2C1006249).

References

[1] N. Cardwell, Y. Cheng, C. S. Gunn, S. H. Yeganeh, and V. Jacobson, “BBR: Congestion-based congestion control,” ACM Queue, vol. 14, 2016.

[2] S. Ha, I. Rhee, and L. Xu, “CUBIC: A New TCP-Friendly High-Speed TCP variant,” ACM SIGOPS Operating Systems Review, vol. 42, 2008.

[3] N. Cardwell, Y. Cheng, S. H. Yeganeh, I. Swett, V. Vasiliev, P. Jha, Y. Seung, M. Mathis, and V. Jacobson, “BBRv2: A Model-based Congestion Control,” in Proc. IETF 104 meeting, 2019.