

ReBBR: Reproducing BBR Evaluation Results

Luke Hsiao
Stanford University
lwhsiao@stanford.edu

Jervis Muindi
Stanford University
jmuindi@stanford.edu

ABSTRACT

This paper describes an attempt to reproduce one of the key findings from the BBR Paper by Cardwell et al. The primary result we are after is performance of BBR and CUBIC in networks link that have non-negligible packet loss. As reported in the original paper, we find that CUBIC has poor performance for loss rates above 0.1% whereas BBR is able to deal with loss much better.

1 INTRODUCTION

In this report, we attempt to validate the experimental results of “BBR: Congestion-based Congestion Control” [1]. We start by looking at the goals, motivations and results from the BBR [1] paper.

Goals What was the original paper trying to solve?

The original paper was trying to find a congestion control approach that would stay as close as possible to optimal network operating point in various network conditions. The network link is operating at the optimal point when bandwidth utilization is maximized and latency is minimized.

Motivation Why is the problem important/interesting?

This is important problem to tackle because traditional congestion control algorithms such as CUBIC have a tough time operating efficiently when there is non-negligible packet loss on the packet. This limitation comes from the CUBIC’s use of packet loss as a signal for congestion, which can unnecessarily hinder throughput leading to poor performance.

Results What did the original authors find?

The BBR paper describes a new form of congestion control that is based on actual congestion going in the network. The insight from the authors is that their approach estimates bottleneck bandwidth and then bottleneck latency in succession so as to be able to get as close as possible to the ideal operating point of maximum bandwidth and minimal latency. The paper had several findings.

The first finding is that BBR can quickly adopt to changes in bottleneck link bandwidth. During operating, BBR spends most of its time continuously probing if more bandwidth has become available. Figure 1 shows BBR reacting to a 2x increase and a 2x decrease to bottleneck bandwidth within about 2 seconds.

The second finding is that BBR is able to avoid bloating router buffers unnecessarily and thus it is able to keep RTT at a minimum value. This is contrast to CUBIC which continues to bloat a router’s buffer until it observes a packet loss event from overflowing the router buffer. Figure 2 shows CUBIC and BBR behave in this regard.

The third and perhaps most important finding is that BBR outperforms CUBIC for non-negligible loss rates. Figure 3 shows a graph showing the performance comparison between BBR and CUBIC for varying loss rates.

The authors report that Google has had a very good experience deploying BBR on B4. B4 is Google data center to data center high-speed Wide Area Network and it’s built primarily from commodity switches that have shallow/small memory buffers. The limited amount of buffering means that there is the occasional packet loss when there is a burst of incoming packets that arrive at the same time. After switching to BBR from CUBIC, Google saw throughput improvements ranging from 2x to 25x.

FIGURE 3: BANDWIDTH CHANGE

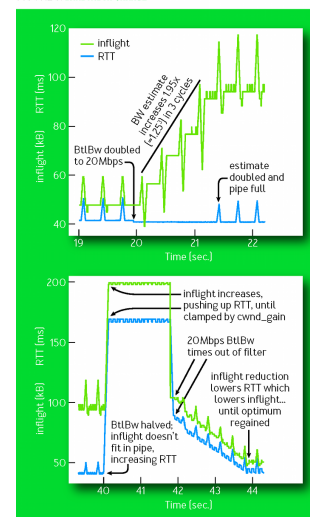


Figure 1: BBR converges to new bottleneck bandwidth at exponential rate.

2 REPRODUCING BBR EVALUATION

Goal What subset of results did you choose to reproduce?

Motivation Why that particular choice?

3 PROJECT STATUS

Progress so far.

Plan Plan for the next few weeks.

REFERENCES

- [1] Neal Cardwell, Yuchung Cheng, C Stephen Gunn, Soheil Hassas Yeganeh, and Van Jacobson. 2016. BBR: Congestion-Based Congestion Control. *Queue* 14, 5 (2016), 50.

FIGURE 5: FIRST 8 SECONDS OF 10-MBPS, 40-MS CUBIC AND BBR FLOWS

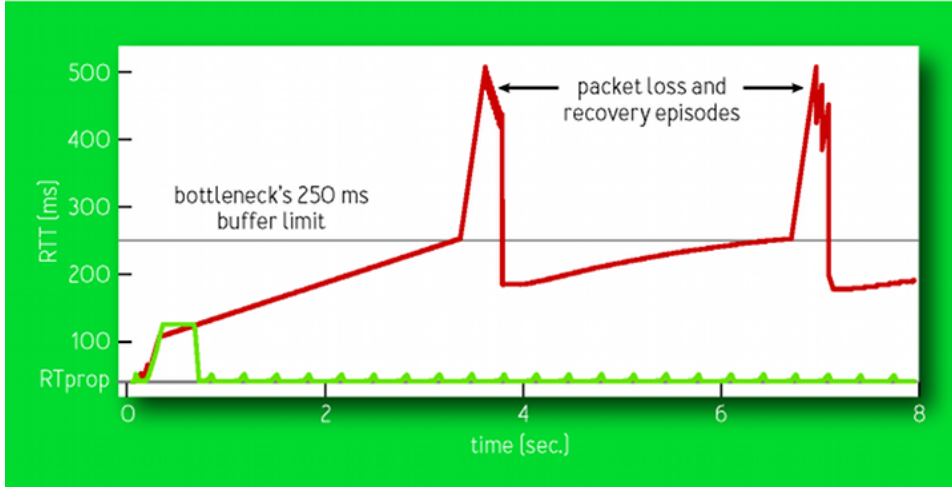


Figure 2: BBR avoids bloating router buffer unlike CUBIC.

FIGURE 8: BBR VS. CUBIC GOODPUT UNDER LOSS

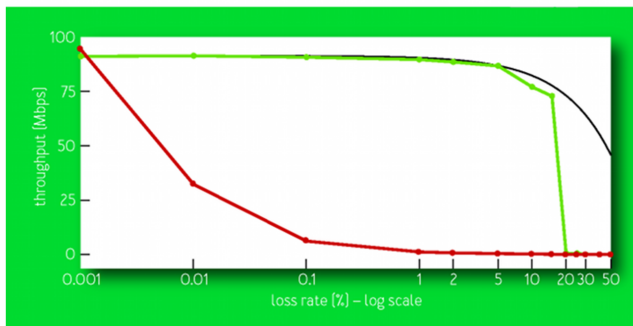


Figure 8 shows BBR vs. CUBIC goodput for 60-second flows on a 100-Mbps/100-ms link with 0.001 to 50 percent random loss. CUBIC's throughput decreases by 10 times at 0.1 percent loss and totally stalls above 1 percent. The maximum possible throughput is the link rate times fraction delivered ($= 1 - \text{lossRate}$). BBR meets this limit up to a 5 percent loss and is close up to 15 percent.

Figure 3: Throughput vs loss rate comparing CUBIC vs BBR.