BBR versus Cubic TCP: Experimentation and Analysis

Praveen Kumar Yelleswarapu(111986420)
Susritha Reddy YaramReddy(111870572)

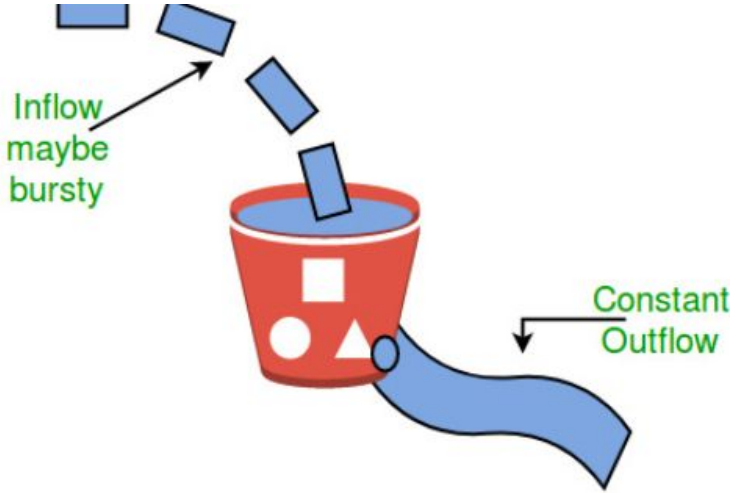
Contents:

1. Introduction
2. Experiments performed
3. Observations and Results

Introduction

Goal:

- Understand and Compare the working of BBR and Cubic Algorithms
- Perform the experiments using servers at various locations



Congestion Control:

- Too many sources sending too much data too fast for network to handle.

Cubic TCP Algorithm:

- Enhanced version of BIC with increased throughput and RTT fairness
- Window growth function is a Cubic function of time
- Does not rely on receipt of ACKs to increase window size; depends on last congestion event instead.

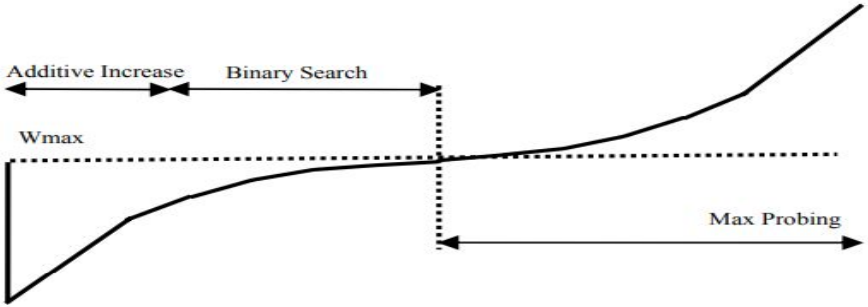


Fig. 1: The Window Growth Function of BIC

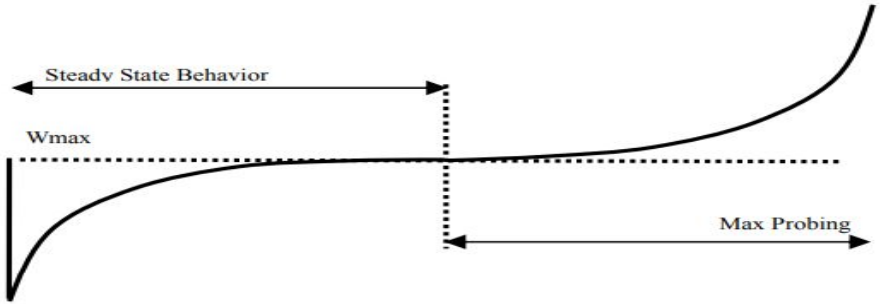


Fig. 2: The Window Growth Function of CUBIC

BBR Congestion Control

- Loss based congestion control - misinterprets loss as a signal of congestion.
- BBR - BottleNeck Bandwidth Round-trip Propagation time
- BBR uses Delivery rate and RTT instead of packet loss as signal of congestion

TCP before BBR

Today's Internet is not moving data as well as it should. TCP sends data at lower bandwidth because the 1980s-era algorithm assumes that packet loss means network congestion.

TCP BBR

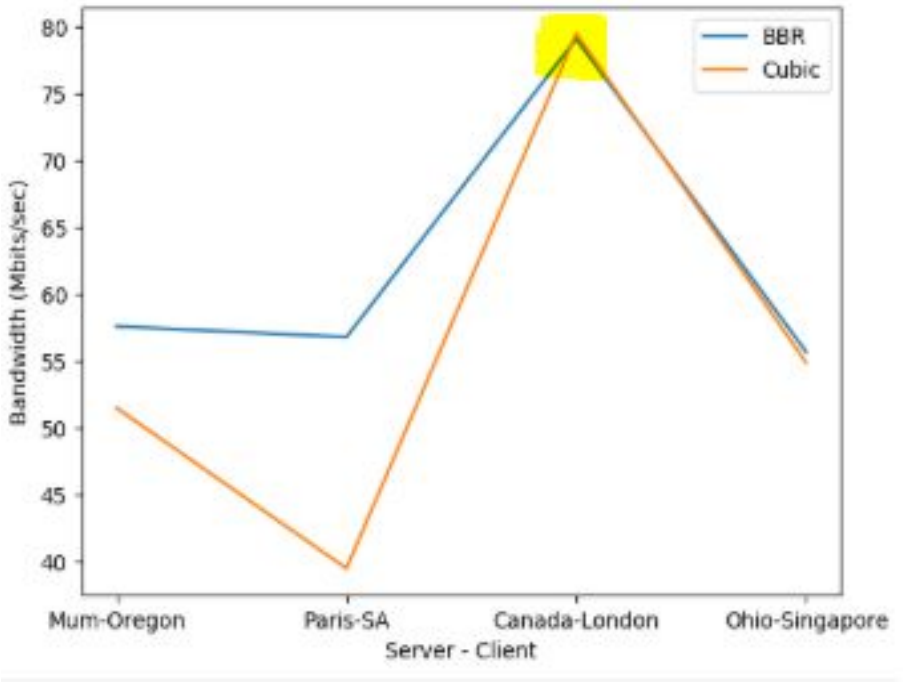
BBR models the network to send as fast as the available bandwidth and is 2700x faster than previous TCPs on a 10Gb, 100ms link with 1% loss. BBR powers google.com, youtube.com, and apps using Google Cloud Platform services.

Experiments Performed : Real Environment

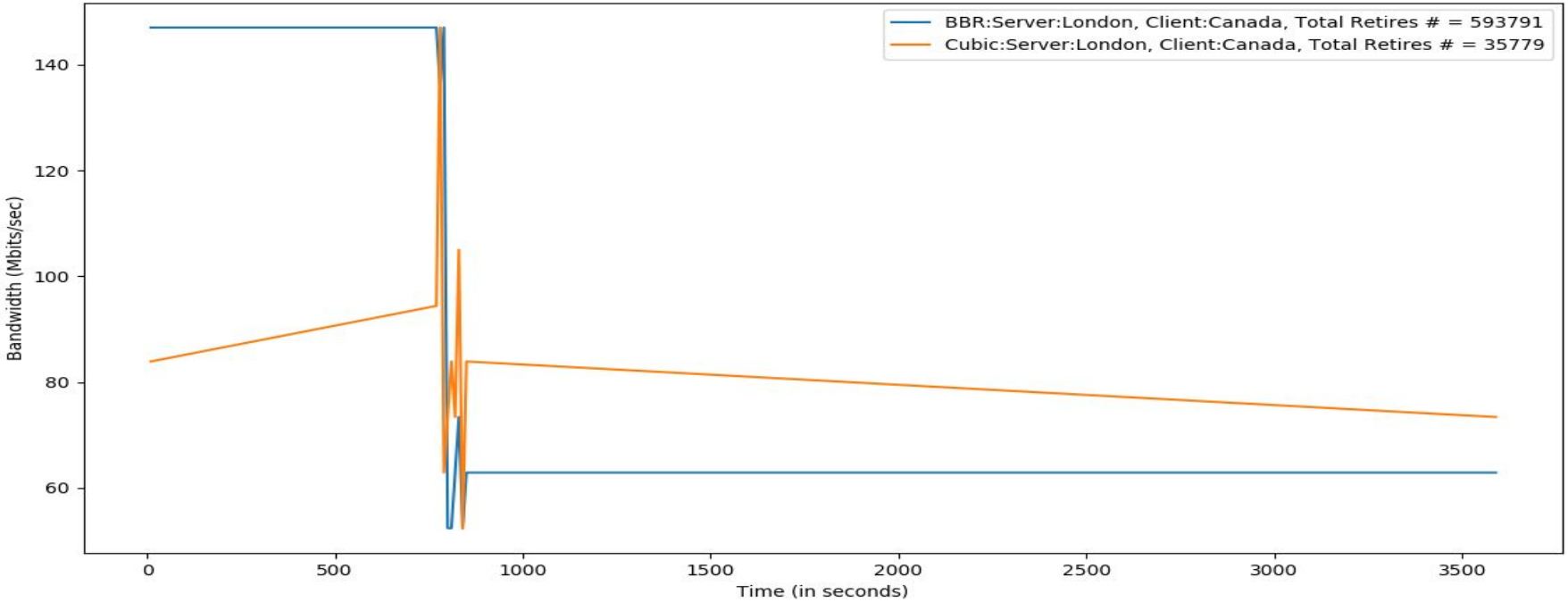
- Created servers and clients in 8 different regions over the world using Amazon EC2 instances.
- Used Iperf3 tool to measure the network speed and performance by sending data from one instance to another in different location.
- In all the tests, servers sent the data and the client received.

Client	**Server**
Mumbai	Oregon
Paris	South America
Canada	London
Ohio	Singapore

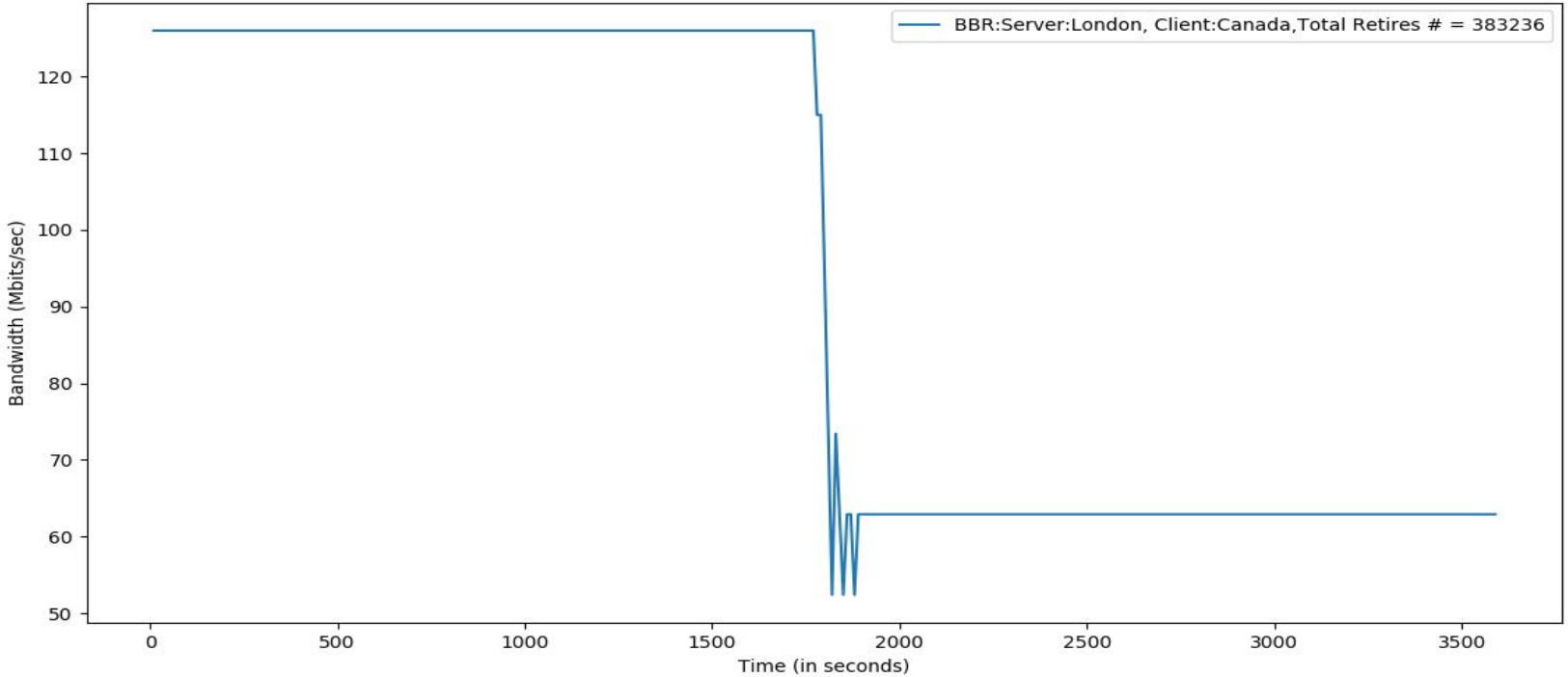
On average, BBR performed approximately 10% better



Observations



Observations Contd..

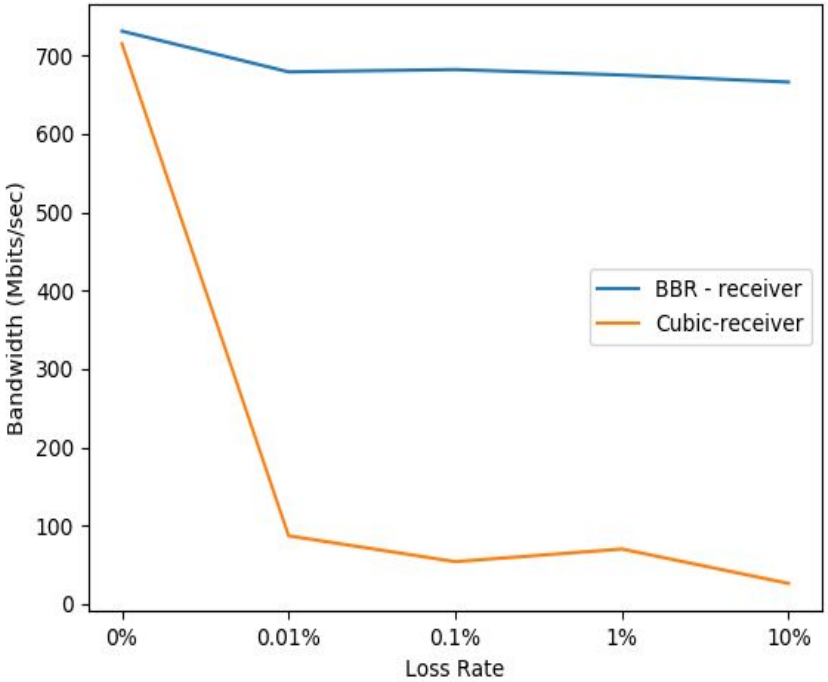


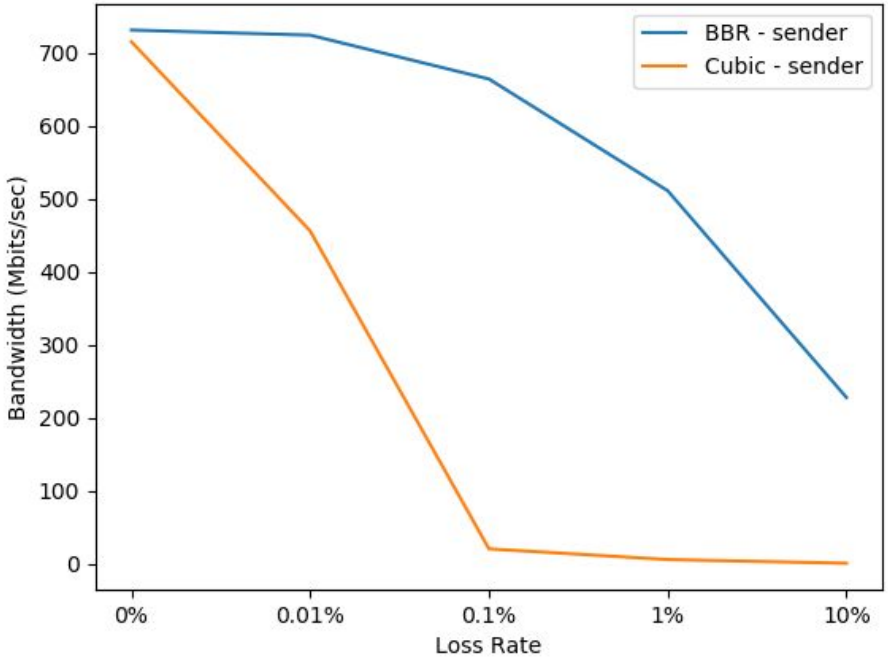
Observations and Hypothesis

Observation - In BBR, Once when the retries are too many, the bandwidth was almost halved, too many retries occur sequentially and this low bandwidth gets stabilized till the end of test. This test was repeated one more time to make sure that it is not a one time issue. This was observed between server as london
and client as canada over AWS

Hypothesis - Due to BBR's indifference to packet loss, when the router's buffers get full they still keep sending data at a high rate due to which the buffer's will not get a chance to get emptied there by it got stuck in this loop.

Experiments Performed : Simulated





Observations

TC is used to simulate a steady constant loss rate which is equivalent to a lossy network, as the loss rate increases, bbr performs much better when compared to cubic. This happens due to 2 reasons.

1) Due to constant loss, cubic gets stuck in congestion avoidance phase and never gets a chance to improve its congestion window size
2) Due to BBR's indifference to packet loss, even when packet loss happens, BBR doesnt decrease it's window size.

Fairness

Setup

1) Tested if an application which runs using one congestion algorithm gets effected, if changes are made to the config file (where we set the congestion algorithm) - Observation: it doesn't effect the running application
2) Tested if we can run two different congestion algorithms at the same time on iperf using different ports - Observation : we can run
3) Tested a combination of the above two after setting a constant rate (so as to differentiate BBR and cubic through bandwidth)

Fairness- Experiment and observation

Experiment - Performed experiments by running Cubic and BBR simultaneously on same machine to observe fairness of both algorithms.

Observation

When BBR and cubic were ran independently their bandwidth was 731 and 715 Mbits/sec, when they were run simultaneously, BBR's throughput came out to be 454 Mbits/sec whereas Cubic's was 462 Mbits/second. Running this second time gave throughput of 479 Mbits/sec to cubic and for BBR it was 438 Mbits/second.

In both experiments we didn't observe huge retries, yet surprisingly BBR's bandwidth came out to be low when compared to cubic.

THANK YOU!!