**REPORT: Lab 3 - Part B : BBR implementation**

1. **If the submitted code doesn't work, please use the commit -**
   **git reset --hard c7ec752da85d929baf381f9b3a5c74666c5adacc**

2. **Implementation**

The BBR state machine has been implemented partially.
The state transitions are simple -
STARTUP -> DRAIN -> PROBE_BW <-> PROBE_RTT

So the states move from STARTUP to DRAIN and then oscillate between PROBE_BW and PROBE_RTT

The CTCP implementation modifications include -

- Pacing based on Max Bandwidth
- Calculation of BW and RTT on receiving an ACK

The plots generated by the testing code are not consistent all the time but seem somewhat correct 50% of the time. I am not sure if the implementation hits the bottleneck bandwidth or the receiver-window because my implementation crashes if I do not honour the receiver window and I have not been able to fix it to throttle the sending side.

**Note the BDP plots below are in Bytes/millisecond and not Bits/millisecond (because I recorded the pictures before converting them to Bits/millisecond)**
It is easy to visualise that the graph will just be scaled and the general trend of the graph will be preserved as converting to bits/millisecond is the same as multiplying the y-axis by 8 bits/byte.

Hence the graphs can be considered -

The Plots can also be found in the **lab3/plots/** folder.

Since then, I have modified the ctpc code to now output bits/millisecond, so the new graphs generated might be 8 times scaled. The last 2 figures ( Figure 4 and 5) is an example of the bits/milliseconds plot.

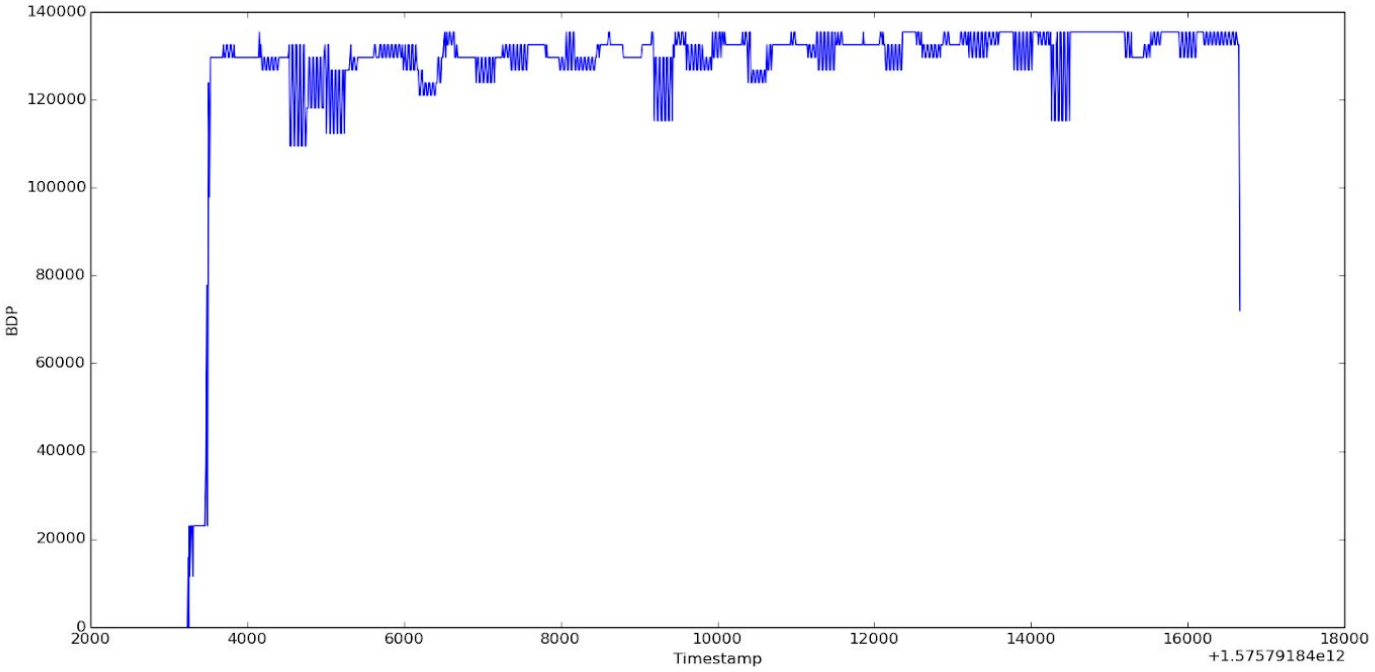


**FIGURE 1**

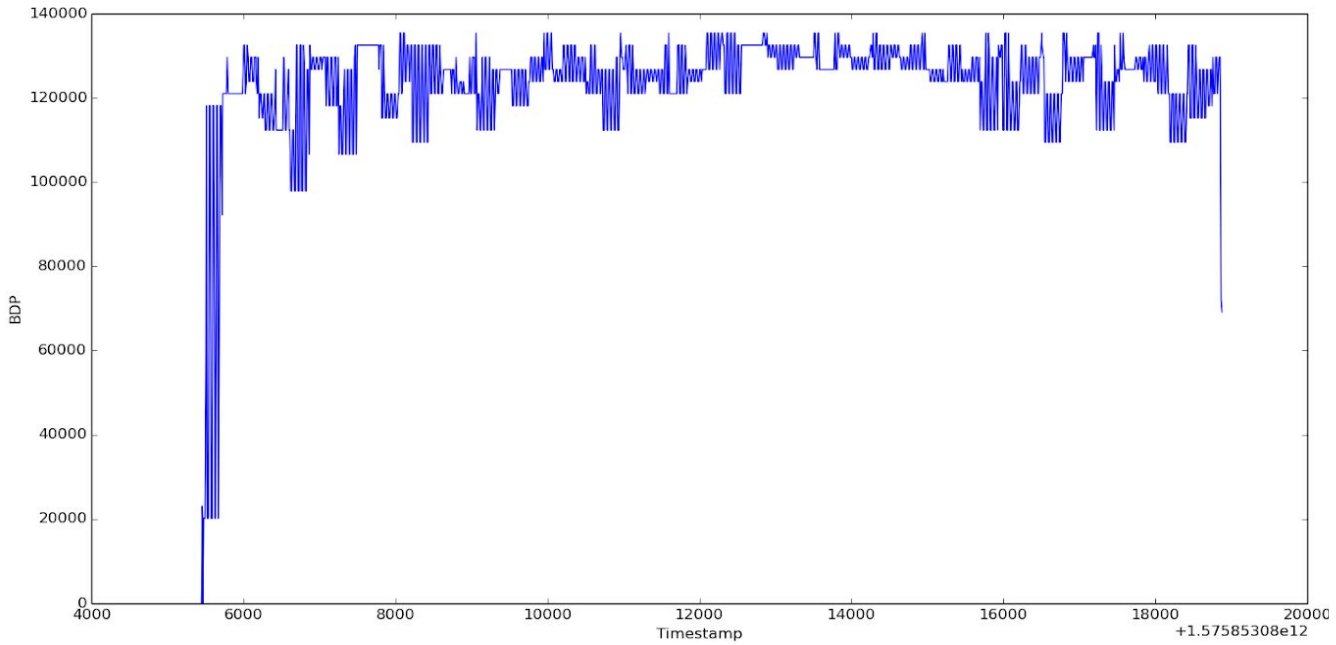


**FIGURE 2**

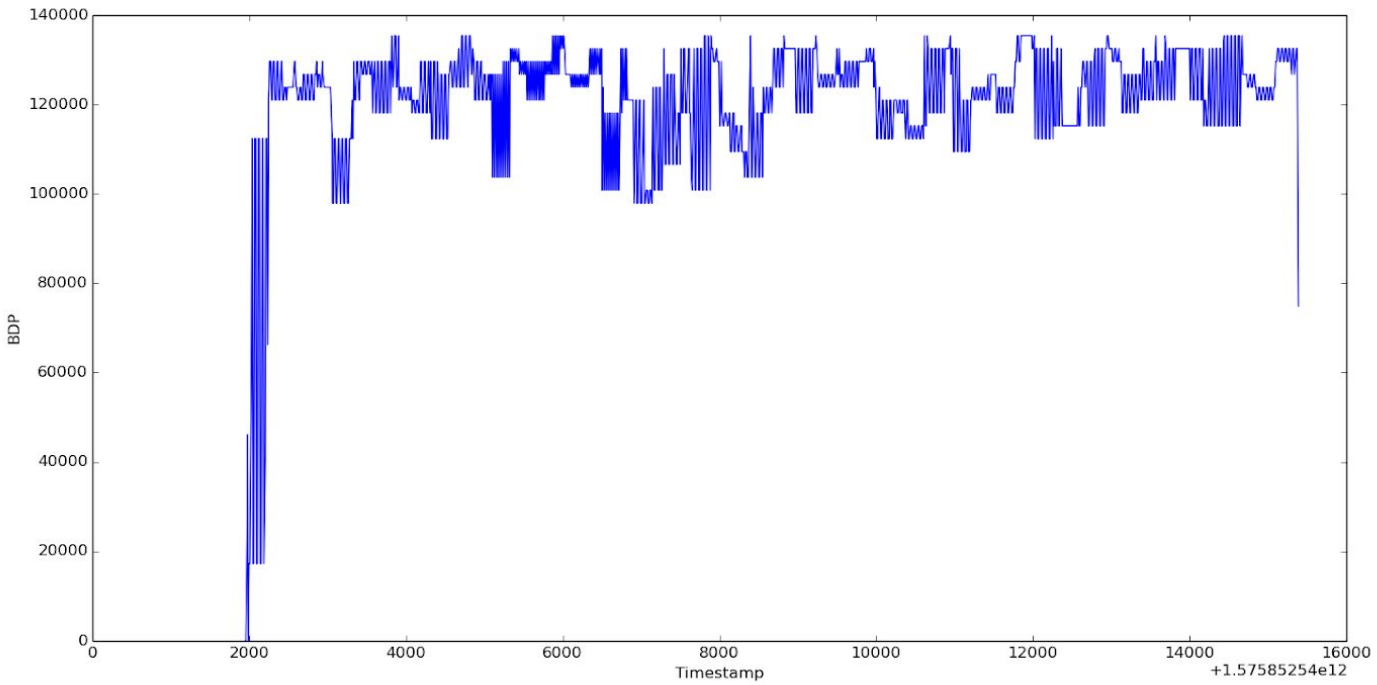


**FIGURE 3**

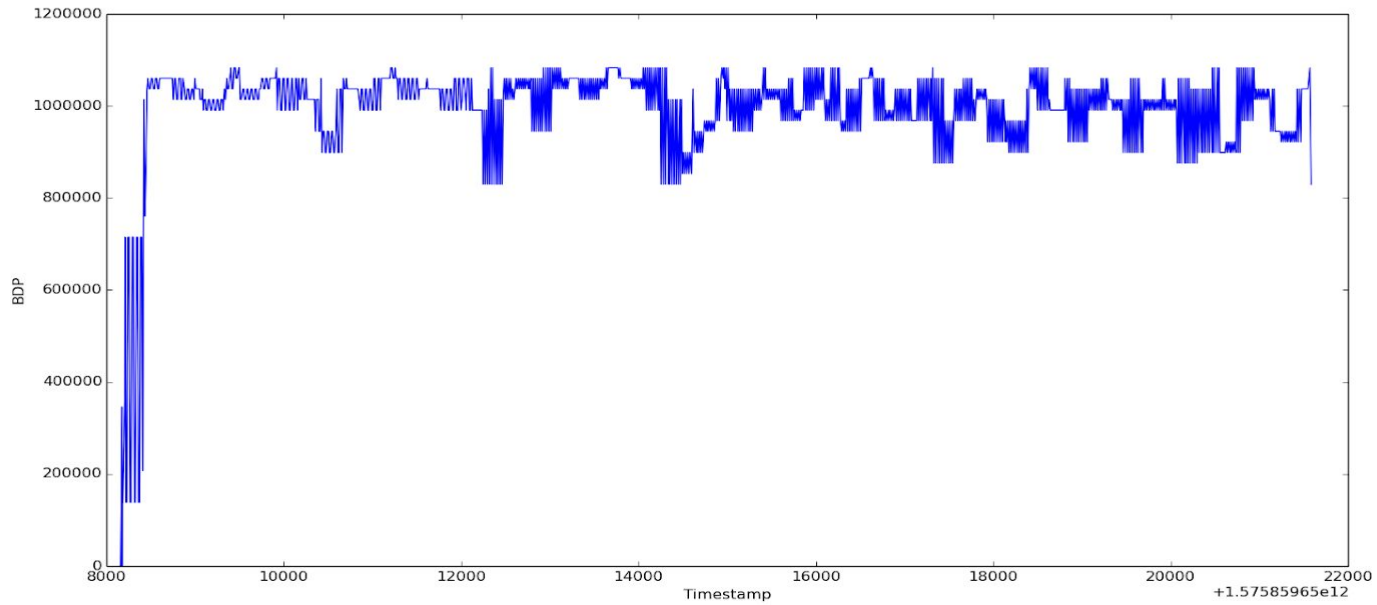


**FIGURE 4**

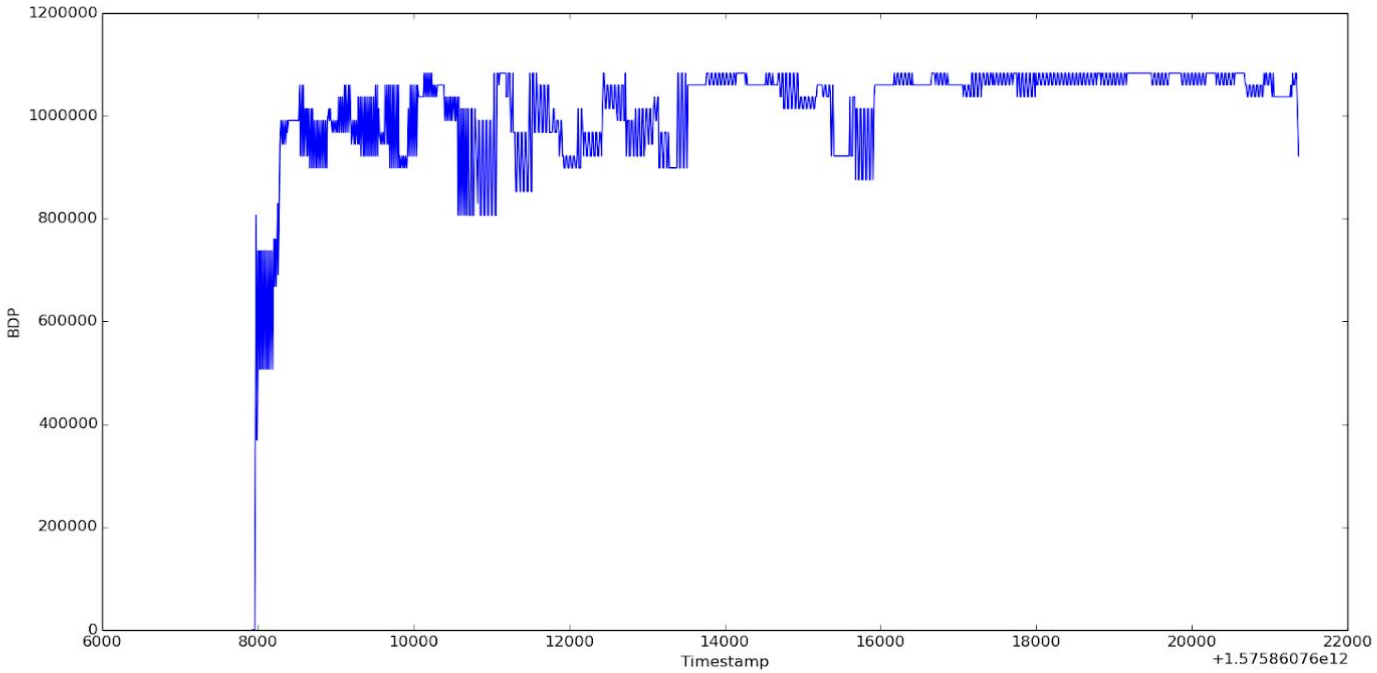


**FIGURE 5**

## DIFFICULTIES FACED WHILE IMPLEMENTATION

While designing TCP, I had not taken care much about extending the code for BBR hence it was a lot of pain rewriting and making fixes and making patch work to the already poorly written tcp code.

This teaches us to always write modular code and make them as much general and decoupled. However this is also how code in the real world evolves. So I got to know how difficult it might be to evolve the already written implementations of network or operating system code bases.

I believe if I had started fresh, without using the already written TCP code, I could have done much better, of course at the cost of more time to develop the code.

The parameters in the BBR are not as discussed in the paper but are arbitrary values which I felt were okay. Also the BBR_PROBE_RTT needs us to reduce the packets sent to 4 packets but I have used a really low PACING value of 0.25 to achieve a similar low sending rate.