### Lab 4 Test TCP Performance

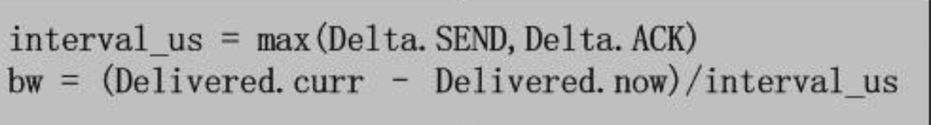
Name: Eduardo Wang Zheng

E-mail: eduardo@sjtu.edu.cn

1. Experiments
   1. BBR algorithm

In fact, BBR algorithm is very simple and consists of four parts:

1. Calculation of instantaneous bandwidth **bw:**



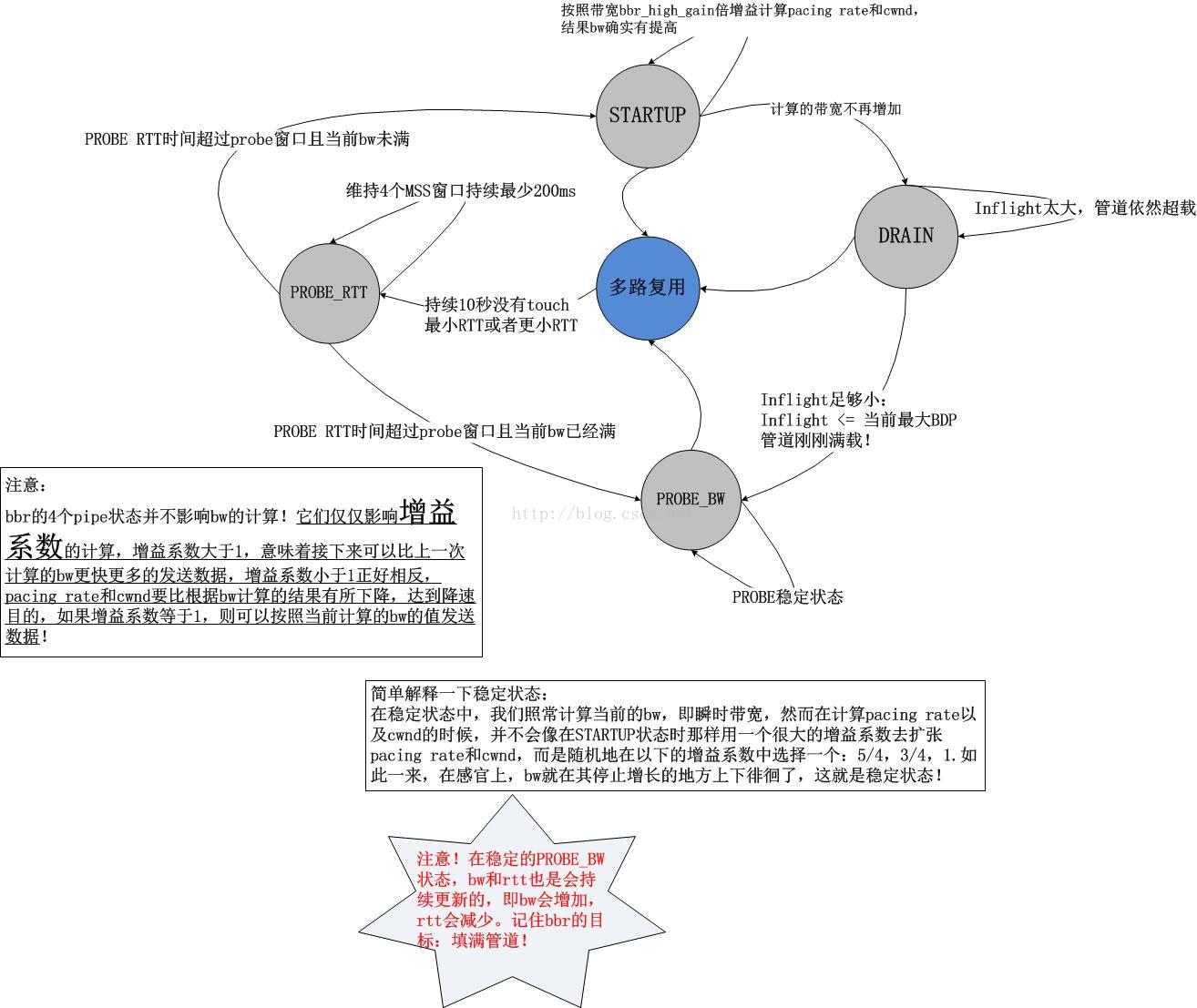
Delivered.curr is the current largest packet N.O. in the tracking list. Delivered.now is the previous largest packet N.O. in the tracking list.

Delivered.curr - Delivered.now is the total ACK packets during the specific time interval.

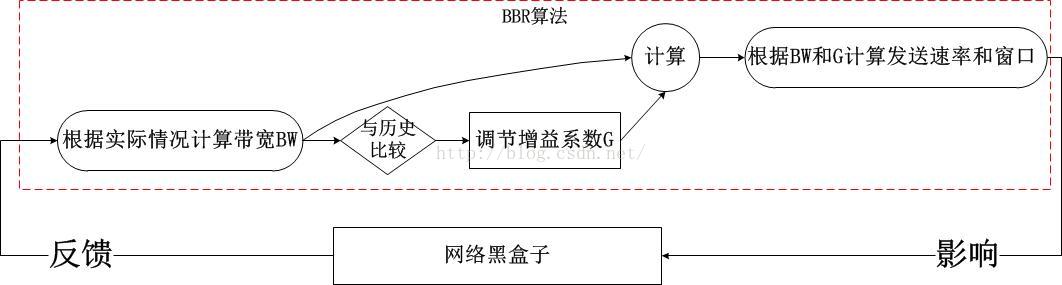
Interval\_us is the time interval, which is the maximum between Delta.SEND and Delta.ACK.

Delta.SEND is the time interval between two sending packets, Delta.ACK is the time interval between two ACK packets.

1. RTT tracking
2. Maintenance of BBR pipe state machine. According to the congestion behavior of the Internet, BBR defines four states, namely STARTUP，DRAIN，PROBE\_BW，PROBE\_RTT. BBR switches freely between the four states by observing the calculated instantaneous bandwidth BW and RTT



Through the above state machine and the bandwidth calculation method in the previous section, we know the working mode of BBR: calculate the **pacing rate** and **cwnd** continuously based on the current bandwidth and the current gain coefficient. During the duration of TCP connection, the instant bandwidth will be calculated every time an ACK is received, and then the result will be fed back to the pipe state machine of BBR to continuously adjust the gain coefficient. This is the whole of BBR. We find that it is a typical closed feedback system. The diagram is as follows:



So we have, **pacing rate=BW\*G**. **cwnd** actually describes a network pipeline, so **cwnd** is actually the capacity of this pipeline, that is, BDP!

The minimum RTT is used to calculate the BDP, and BDP \* G' is used to calculate the **cwnd**. Here G' is the gain coefficient of **cwnd**, which is the same as the bandwidth gain coefficient. It is also obtained according to the BBR state machine.

1. Results (**pacing rate**, **cwnd** ) output. When BBR calculates **cwnd**, it also calculates a corresponding **pacing rate**, which specifies the time interval between packets of one window size indicated by **cwnd.**
   1. Test TCP Reno and TCP BBR in Mininet

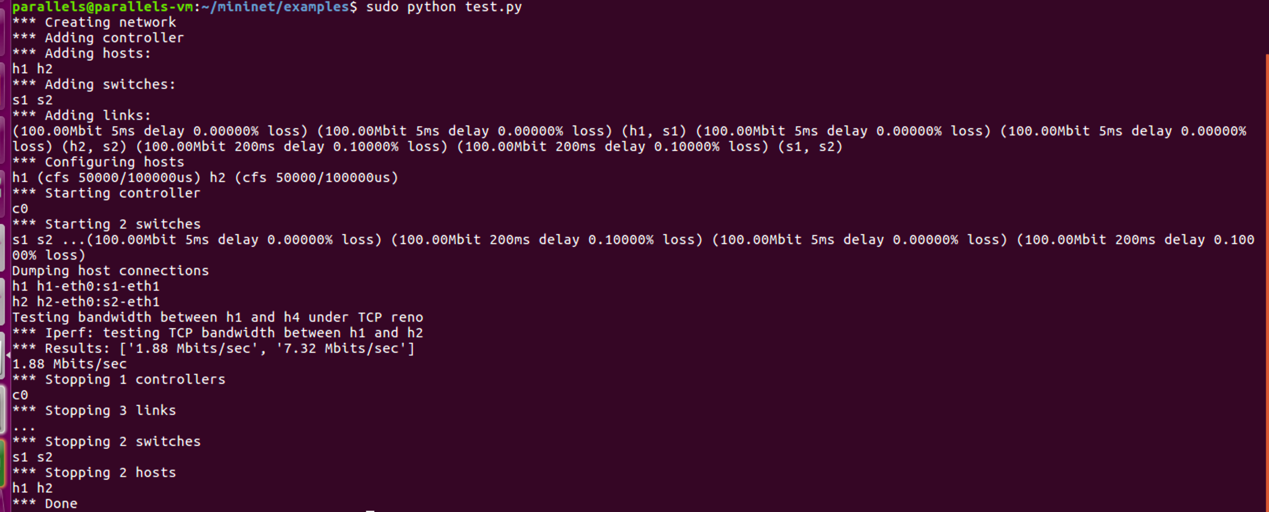


Figure1 Test TCP Reno

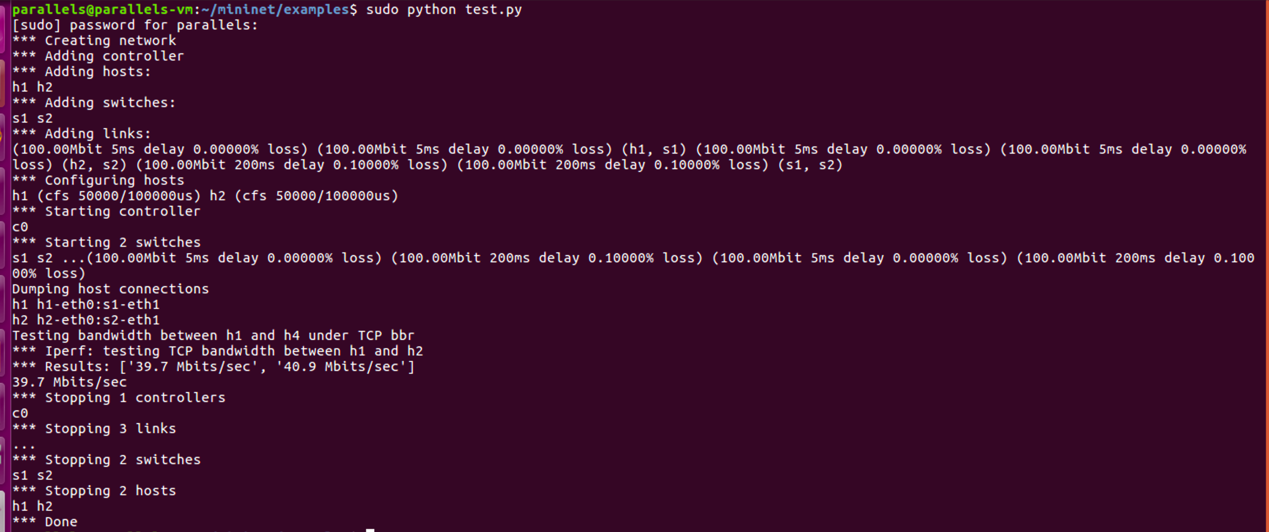


Figure2 Test TCP BBR

* 1. Study how TCP throughput varies with respect to link bandwidth/link delay/loss rate for the above two TCP versions
     1. Evaluation

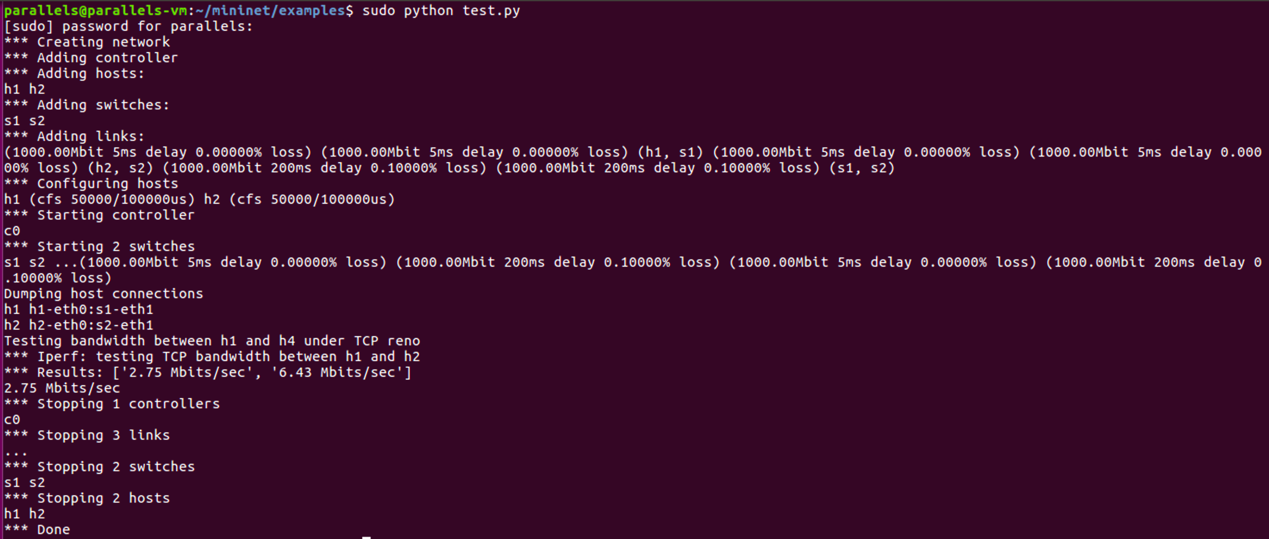


Figure3 Test TCP Reno

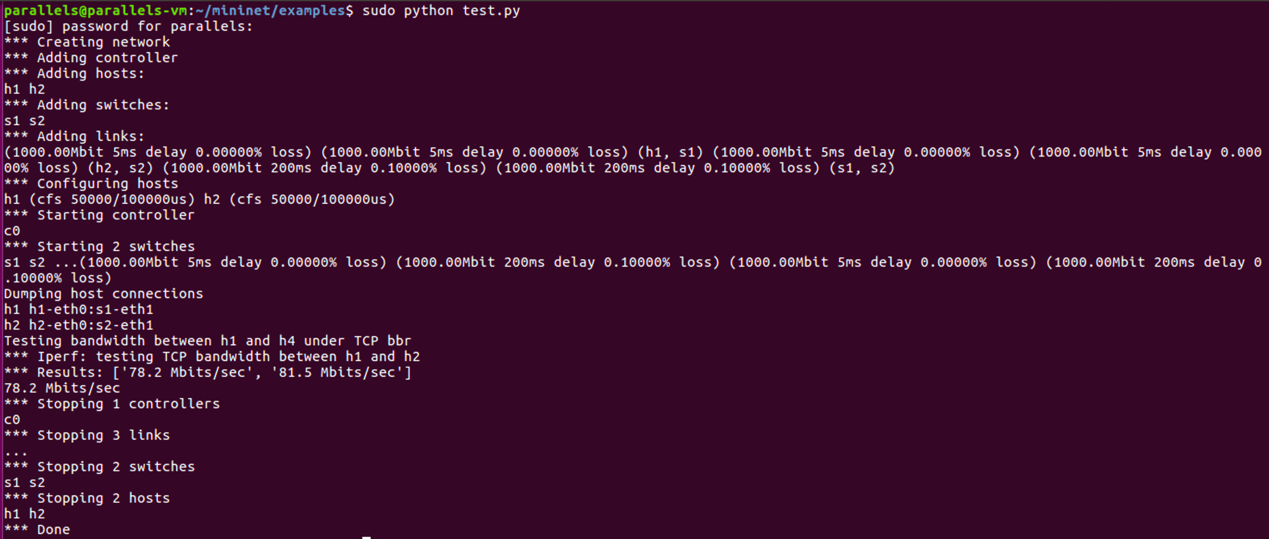


Figure4 Test TCP BBR

From figure 1, 2, 3 and 4, we can know that the TCP throughput for both two TCP versions increase as the **link bandwidth** increases. But TCP BBR seems to increase faster.

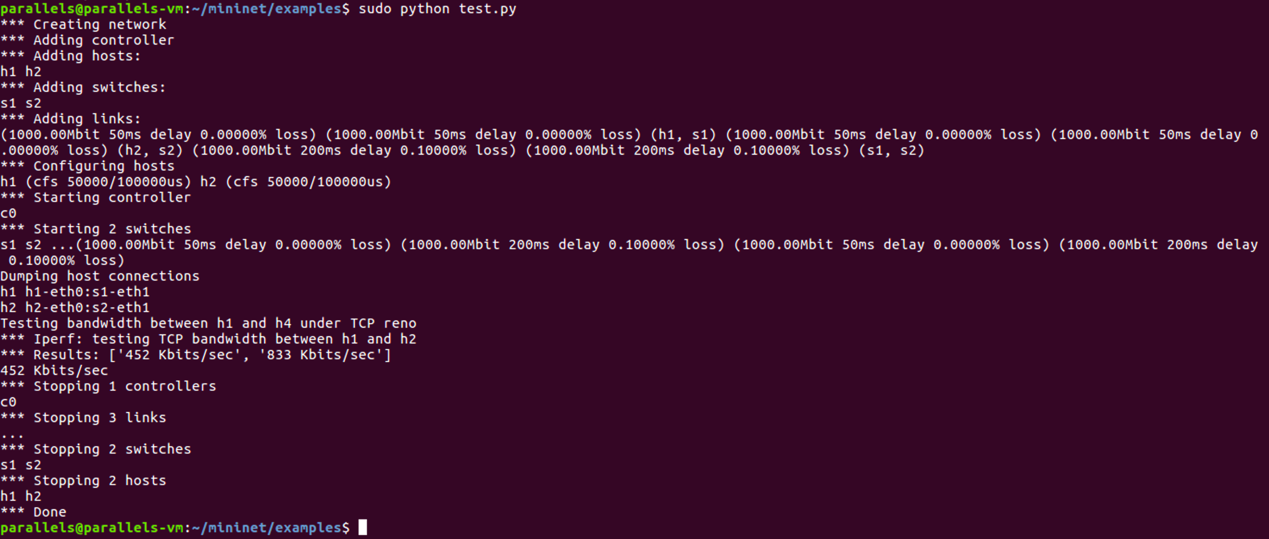


Figure5 Test TCP Reno

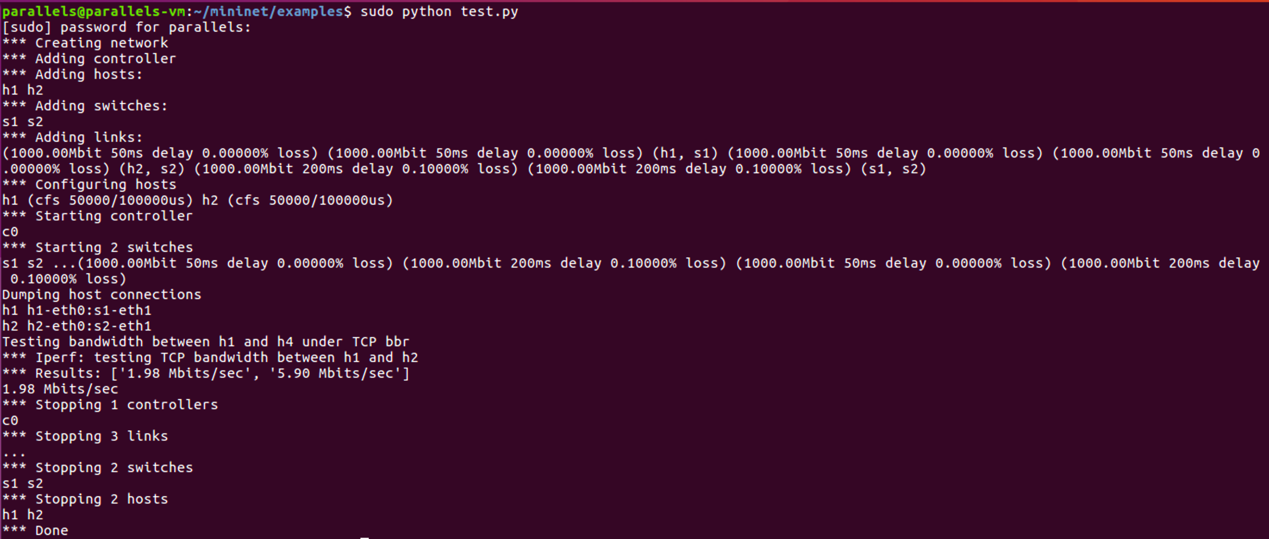


Figure6 Test TCP BBR

From figure 3, 4, 5 and 6, we can know that the TCP throughput for both two TCP versions decrease dramatically as the **link delay** increases.

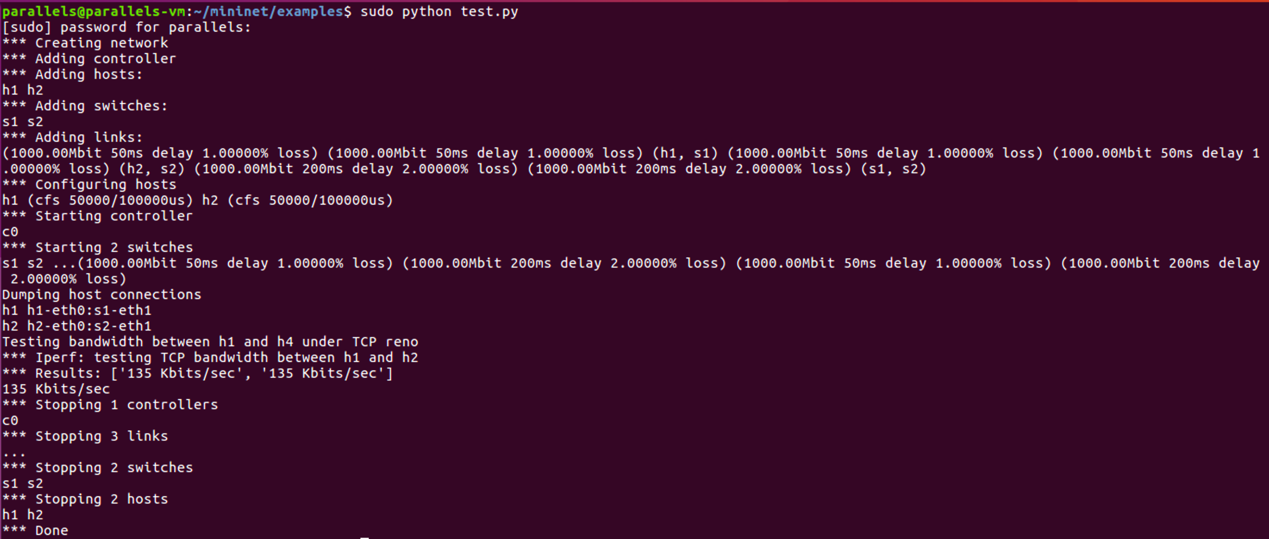


Figure7 Test TCP Reno

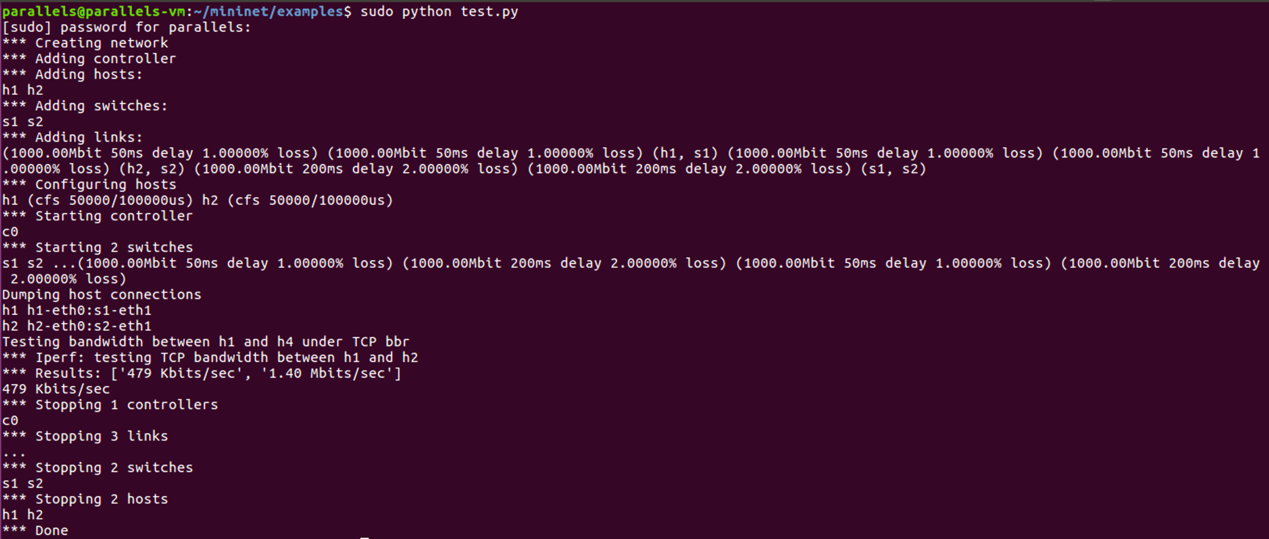
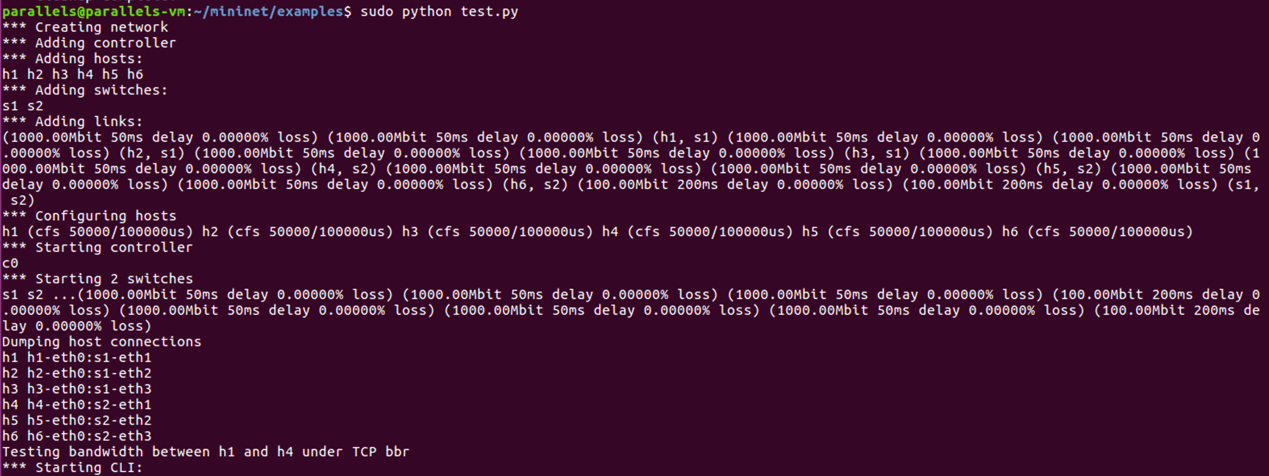


Figure8 Test TCP BBR

From figure 5, 6, 7 and 8, we can know that the TCP throughput for both two TCP versions decrease as the **loss rate** increases.

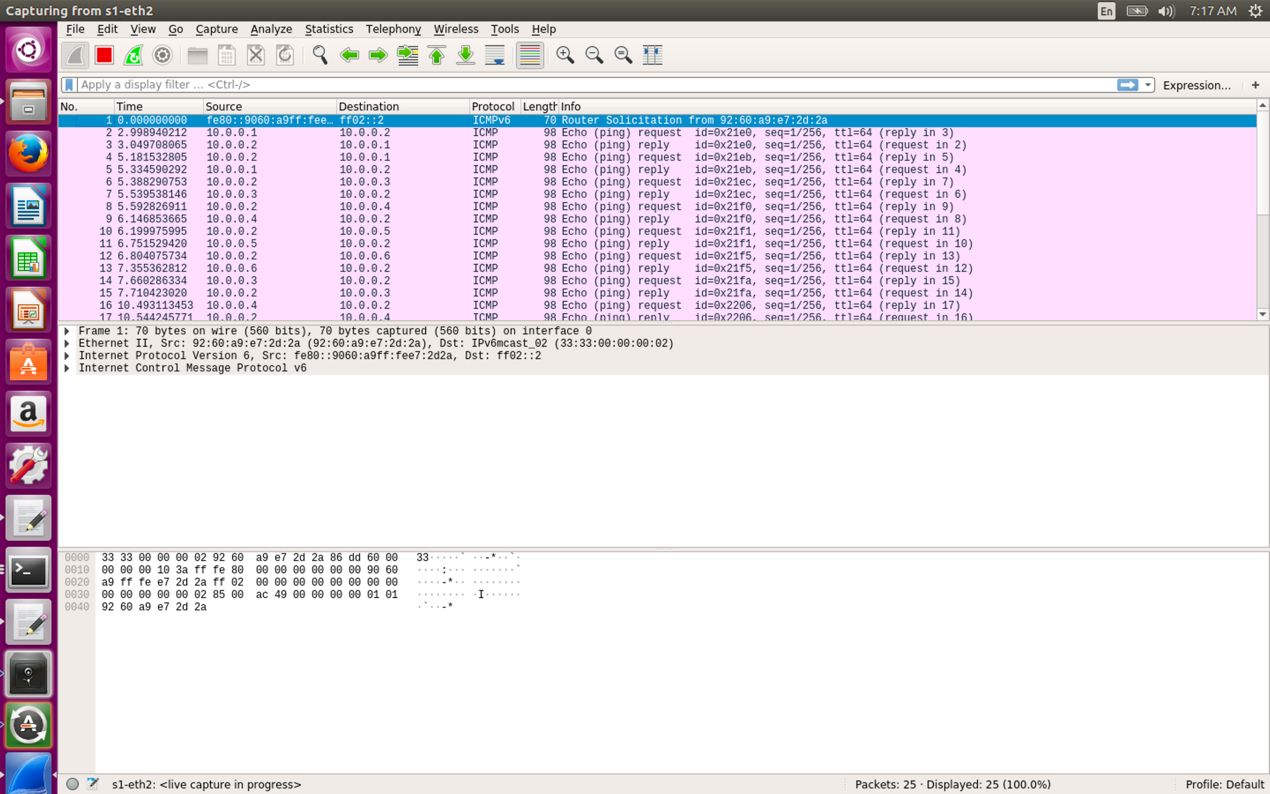
* 1. Construct a network with a bottleneck link shared by multiple pairs of senders and receivers

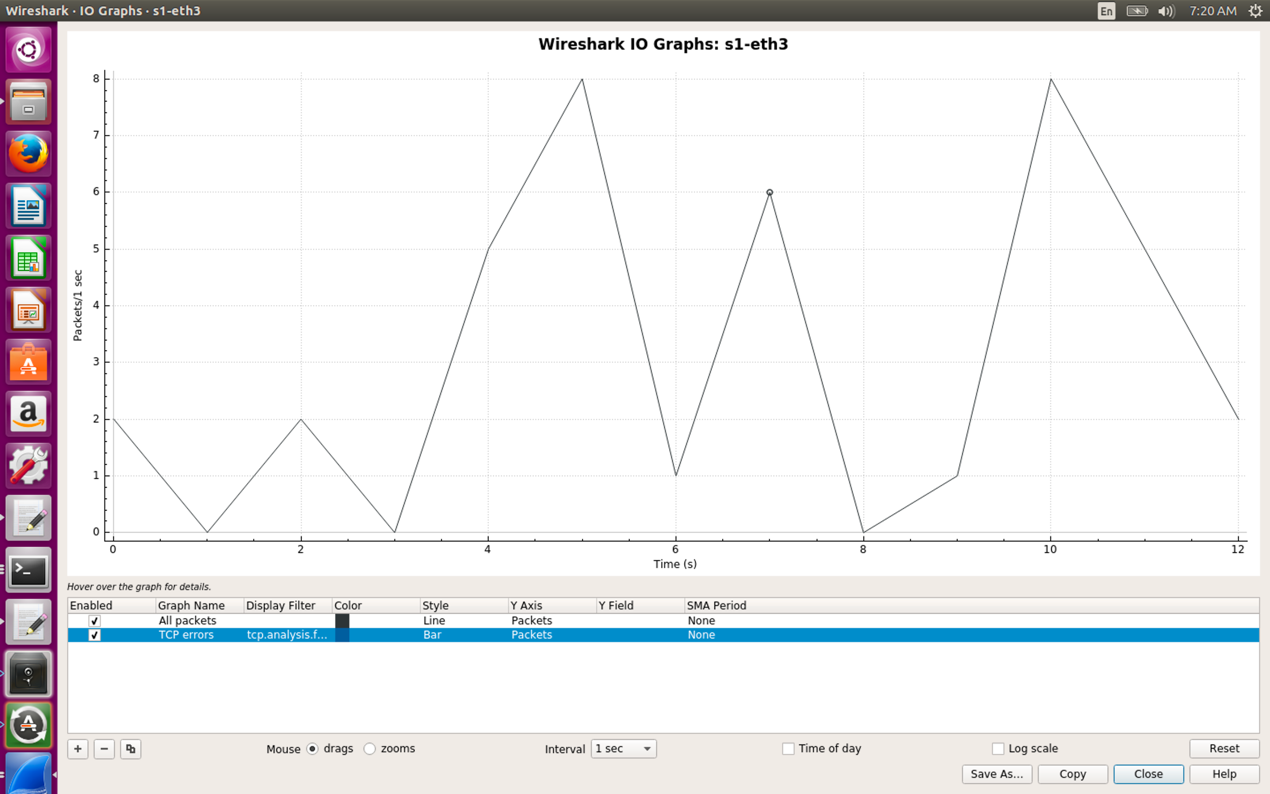


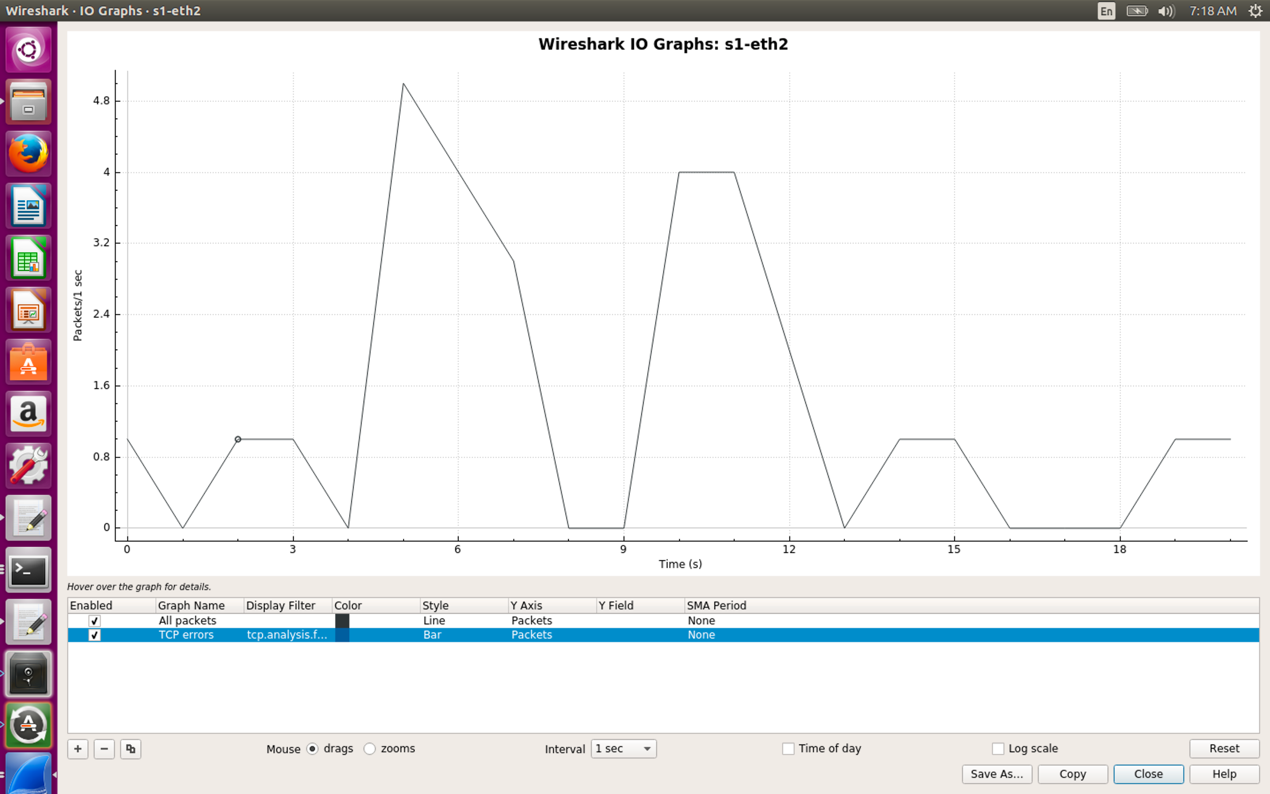
I created a network where 3 pairs share a bottleneck link.

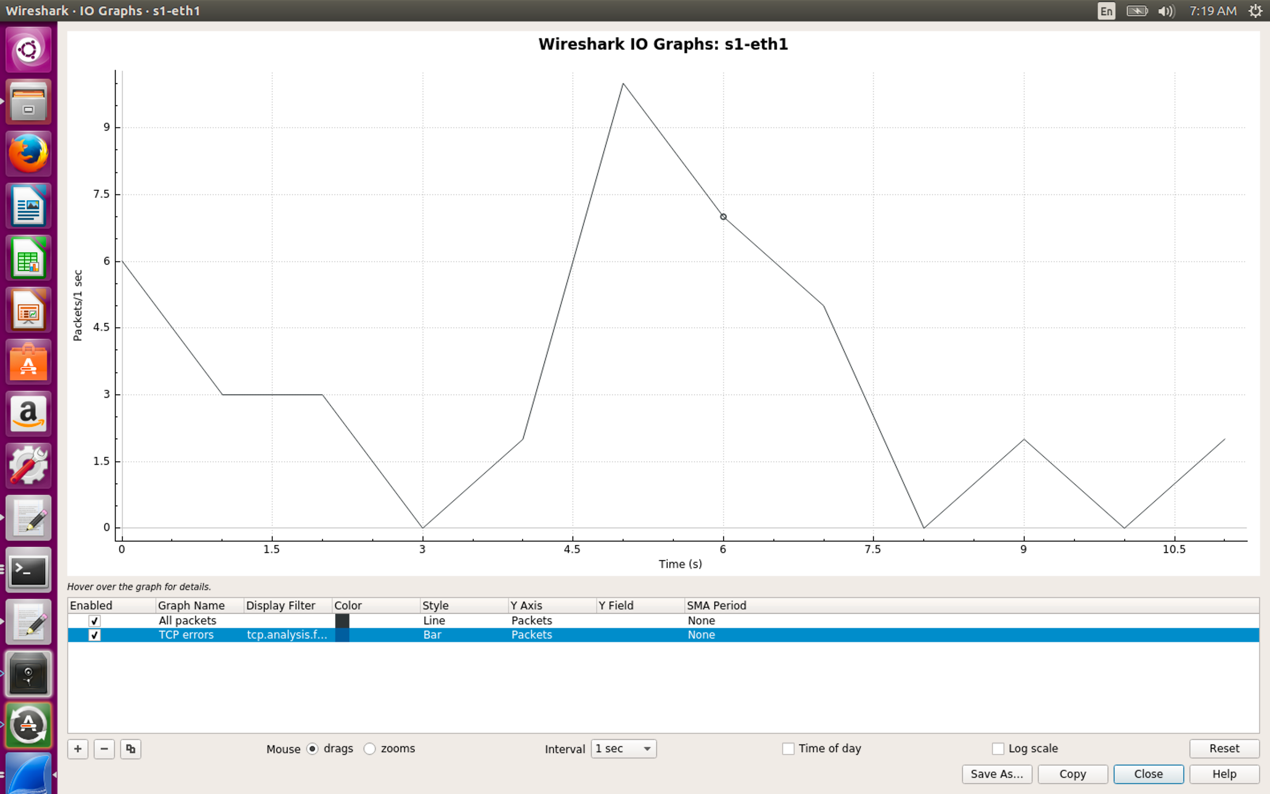
* 1. Study how these sender-receiver pairs share the bottleneck link.











I used wireshark and execute a single instruction to study how these sender-receiver pairs share the bottleneck link.

It seems a total mess. However, we can find that in a specific time period, there is only one pair occupies most of the bottleneck link. I think this indicates that the relationship between the 3 pairs who share a bottleneck link is actually similar to a kind of competition mode.