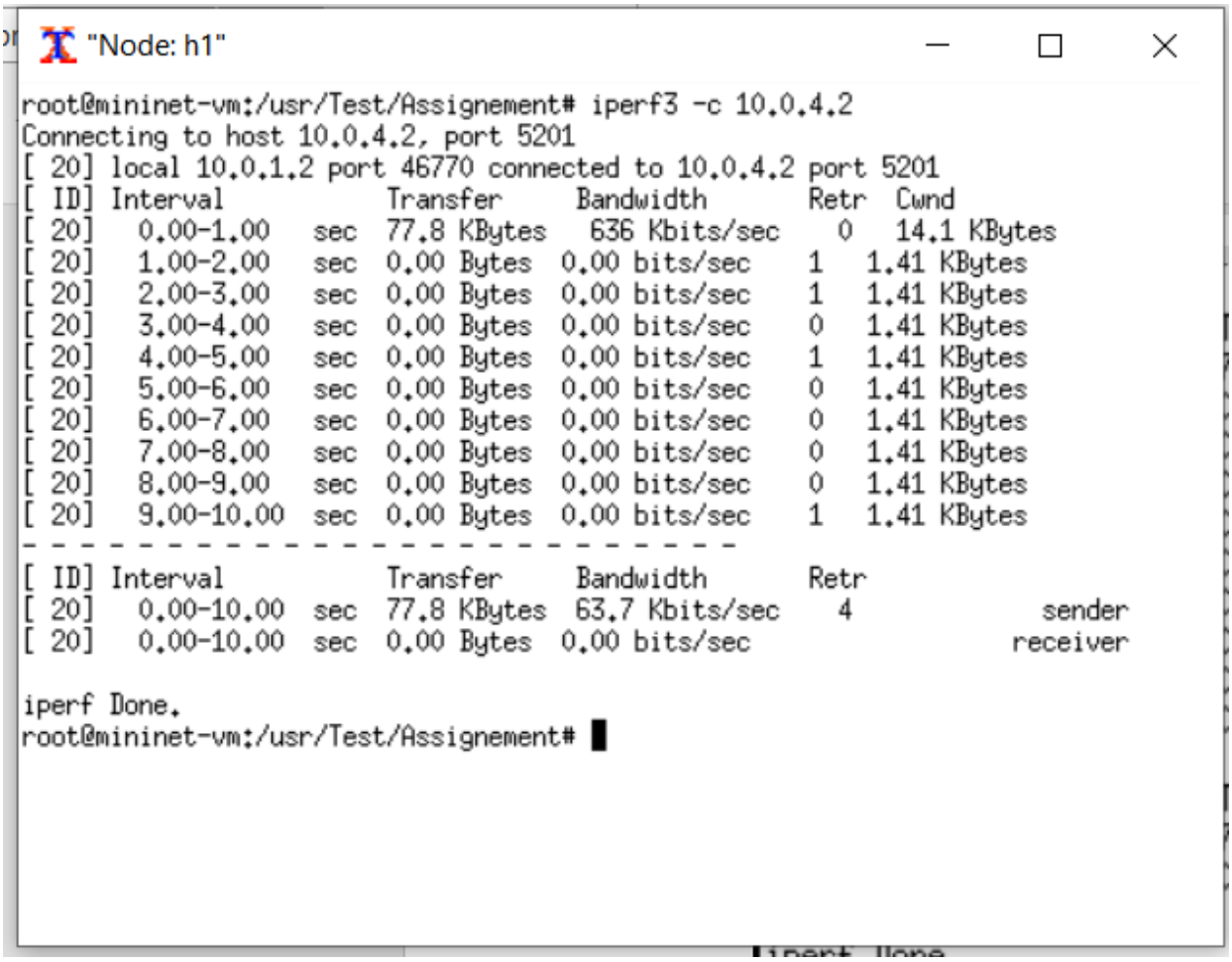
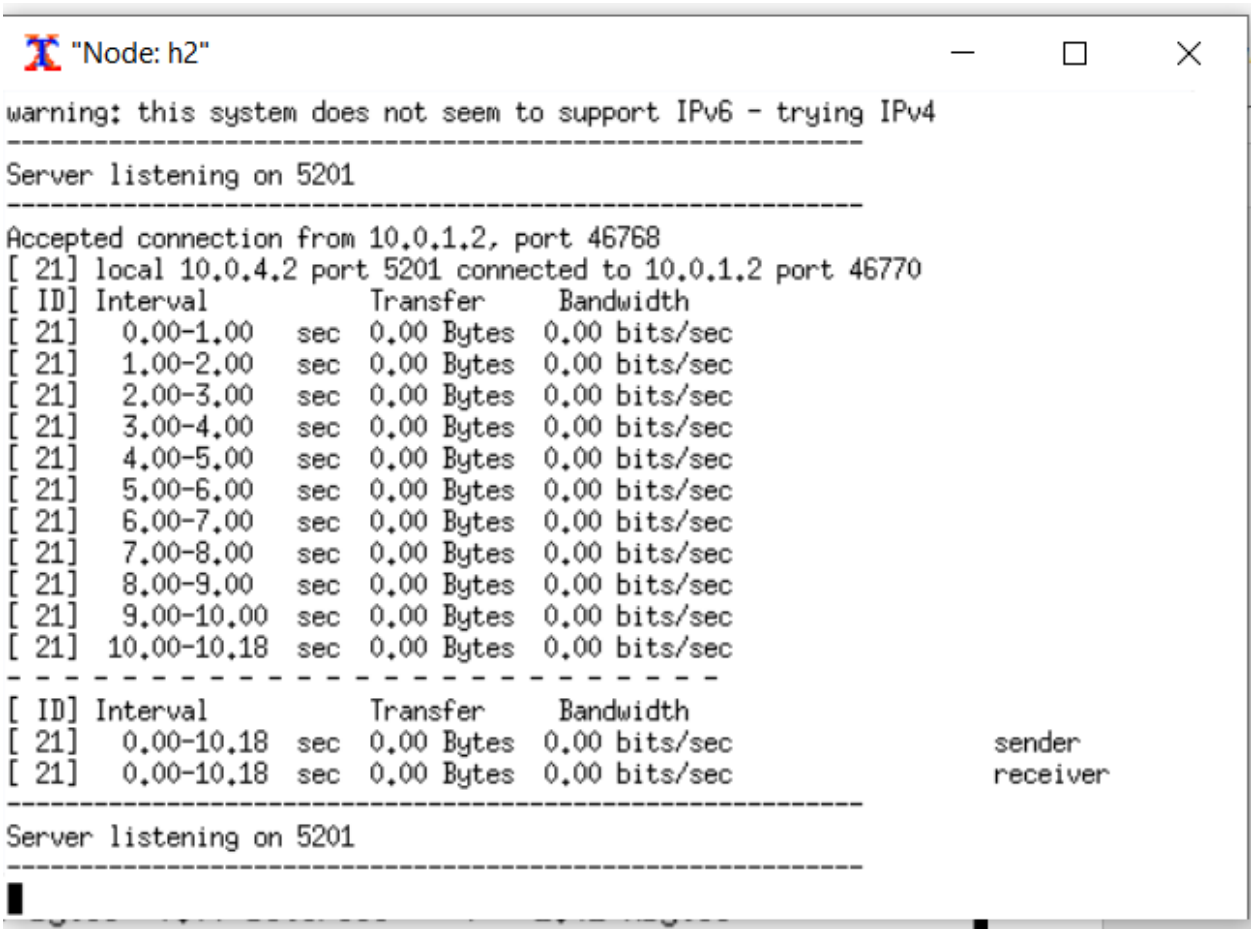
For Buffer size(limit) 10 kbits

1. Client-side node (h1)



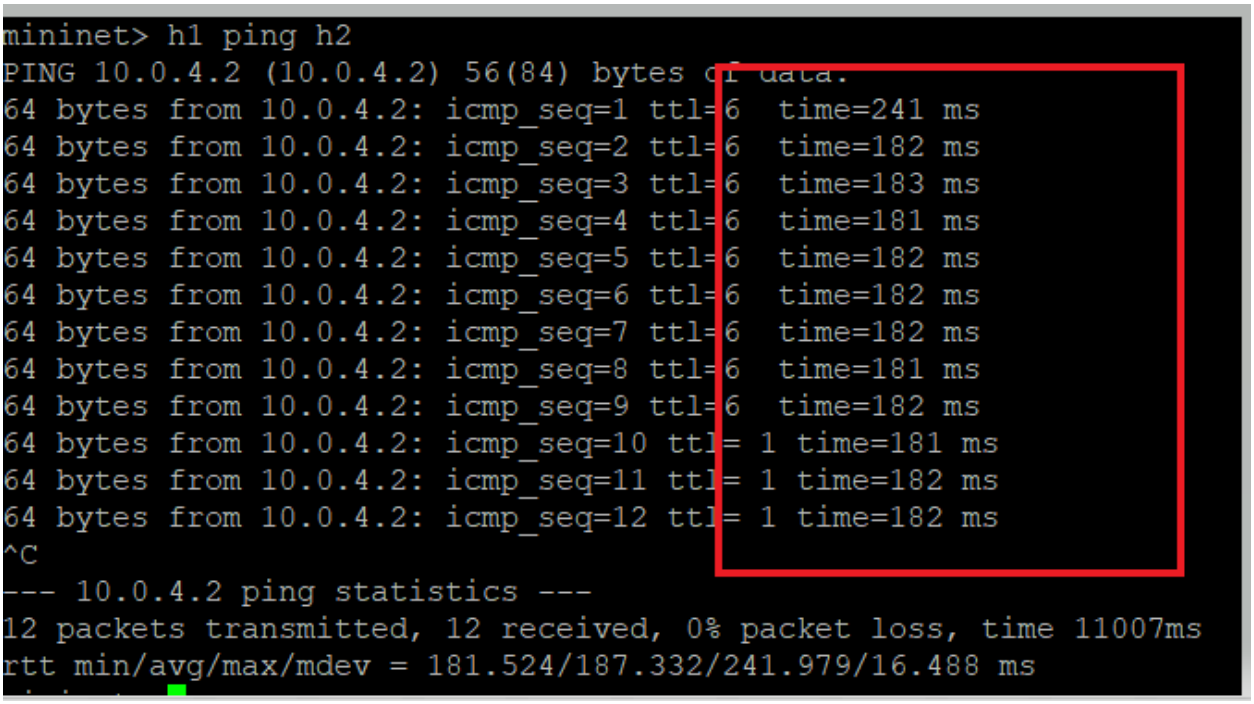
Server-side node (h2)



In this particular case, the buffer size is very small when compared to the flow rate. The burst size (or the bucket size) is 1 mbit but the queue size(limit) is only 10 Kbit, obviously whenever the bucket is freed, the queue size cannot hold all the packets, so it will drop packets instantaneously, which is translated in terms of cwnd decreasing in the first RTT itself. Every time the bucket(burst) is freed there will be drop in packets. As a result, the cwnd will never recover and it will be kind of stalled, which is evident from the above experiment.

Assuming that the RTT is 180 ms (30 ms delay at each router – 3 routers – to and fro is approx. 180 ms)

Confirmed using ping:

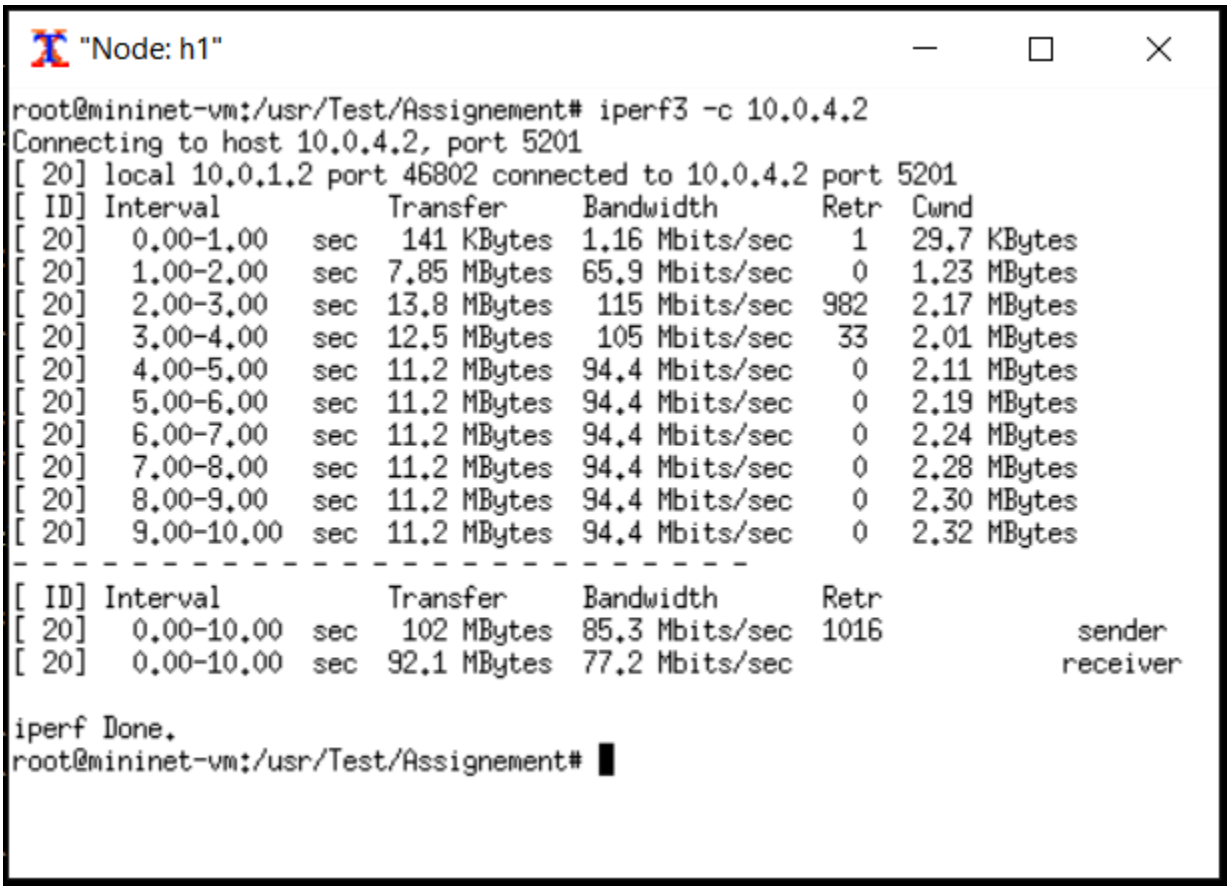


BDP (Bandwidth Delay Product) = Bandwidth * RTT

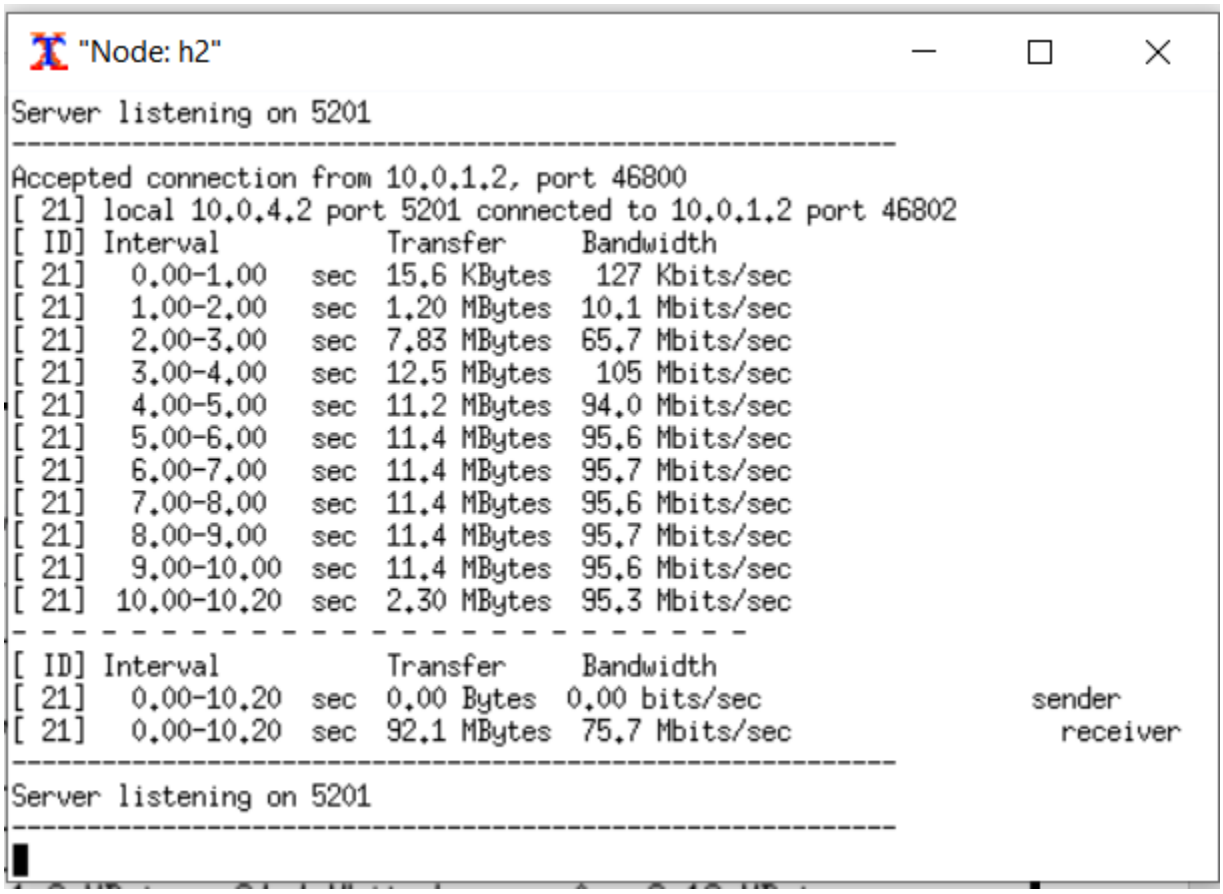
BDP (limit = 10 kbits) = 63.7 kbits/s * 180 ms = 11.4 kbits

For Buffer size(limit) is 5 mbits

1. Client side



2. Server side



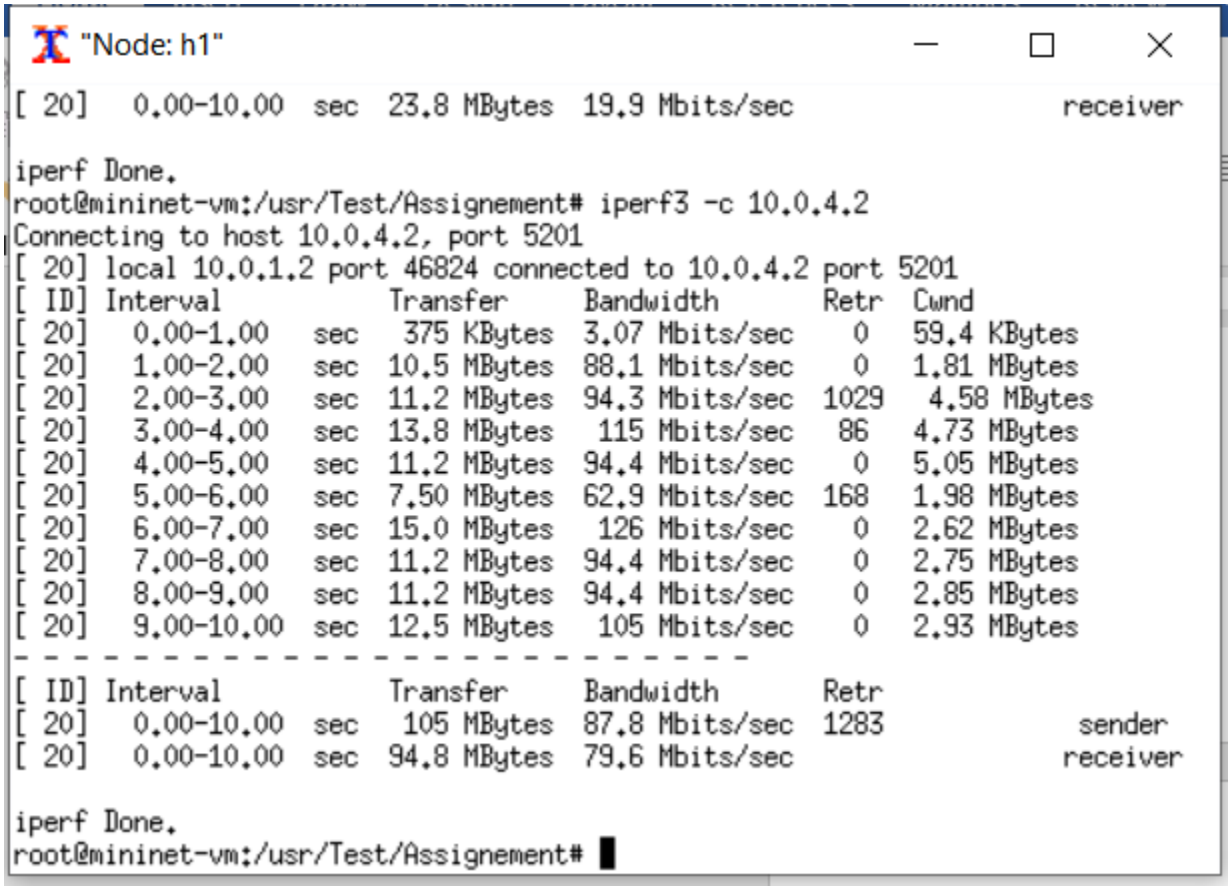
In this case, as the buffer size (limit) is increase to 5 mbits, which is greater than the bucket size (1 mbits), we can see a considerable increase the throughput and there is only drop in the cwnd, when the interface/router tried to go overboard than the allowed bandwidth, which is where the packets were dropped, other than that, when the interface/router is working within the limits of the set bandwidth, it is working as expected. There are minor losses which I think can be attributed to the delay we introduced into the router. Here, we can also see that the cwnd is almost consistent, as the buffer size(limit) and the burst size are having similar capacity, the feedback between dropping the packets and the cwnd correction is very responsive.

BDP (Bandwidth Delay Product) = Bandwidth * RTT

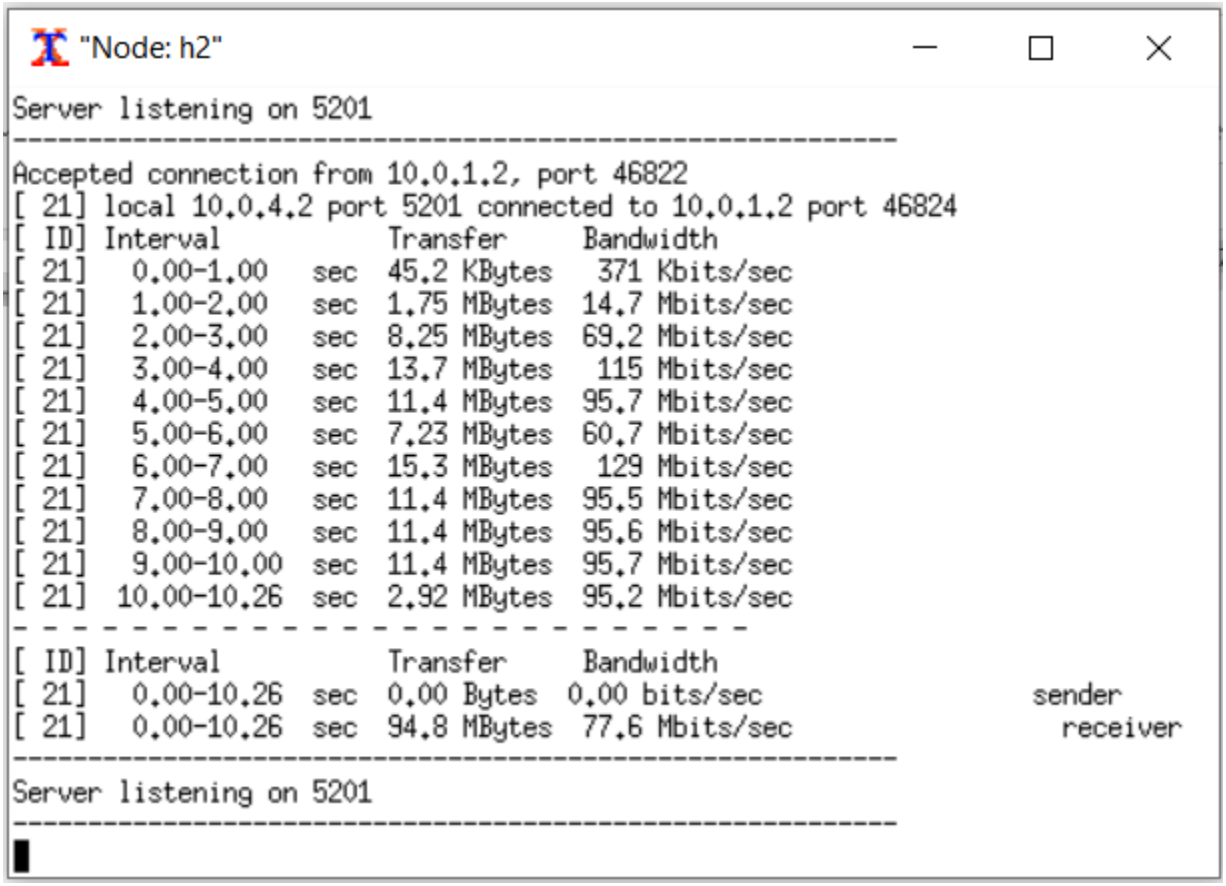
BDP (limit = 5 mbits) = 85.3 mbits/s * 180 ms = 15.35 mbits

For Buffer Size (limit) is 25 mbits

1. Client side



2. Server Side



BDP (Bandwidth Delay Product) = Bandwidth * RTT

BDP (limit = 25 mbits) = 87.8 mbits/s * 180 ms = 15.8 mbits

In this case, the buffer size is almost 25 times the bucket size, so there is a very little chance of dropping the packets because the buffer capacity. So here most of the retransmissions are happening because the concerned packet is staying the queue for long, but not due to packet drop (as we have enough buffer capacity). Here also the throughput is similar to the previous version, mainly due to the fact that it tries to pump way over set bandwidth, because of this it observes reasonably high re-transmissions, because of which the cwnd get down and it has to start afresh which kind off balances the extremes and gives a throughput similar to the previous setting.