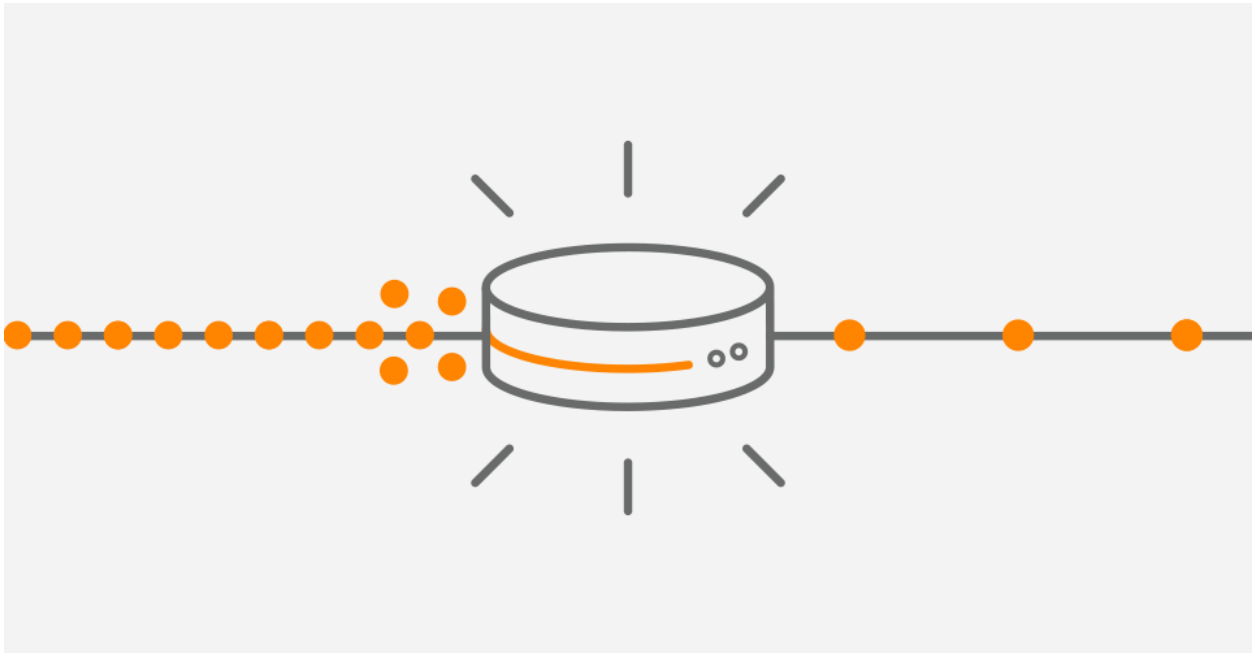
Networks

Understanding TCP
Assignment #2



Github Repo : https://github.com/khadijaAssem/Understanding-TCP

Introduction

Average TCP throughput at the receiver:

Wireshark · Conversations · tcp-example-1-0.pcap

Ethernet | IPv4 · 1 | IPv6 | TCP · 1 | UDP

Address A	Port A	Address B	Port B	Packets	Bytes	Packets A → B	Bytes A → B	Packets B → A	Bytes B → A	Rel Start	Duration	Bits/s A → B	Bits/s B → A
10.1.1.1	49153	10.1.1.2	8080	14,166	5,732 k	9,259	5,461 k	4,907	270 k	0.000000	8.9944	4,857 k	240 k

1. Run the simulation with the default parameters and answer the following questions.

- **What is the average throughput of the TCP transfer?**

$$\text{Average receiving throughput} = \frac{5{,}461k}{8.9944} = 4{,}860\text{k bit/sec} = 4.9\text{Mbps}$$

- **What is the maximum expected value of throughput?**

  Maximum expected value of throughput = 5Mbps

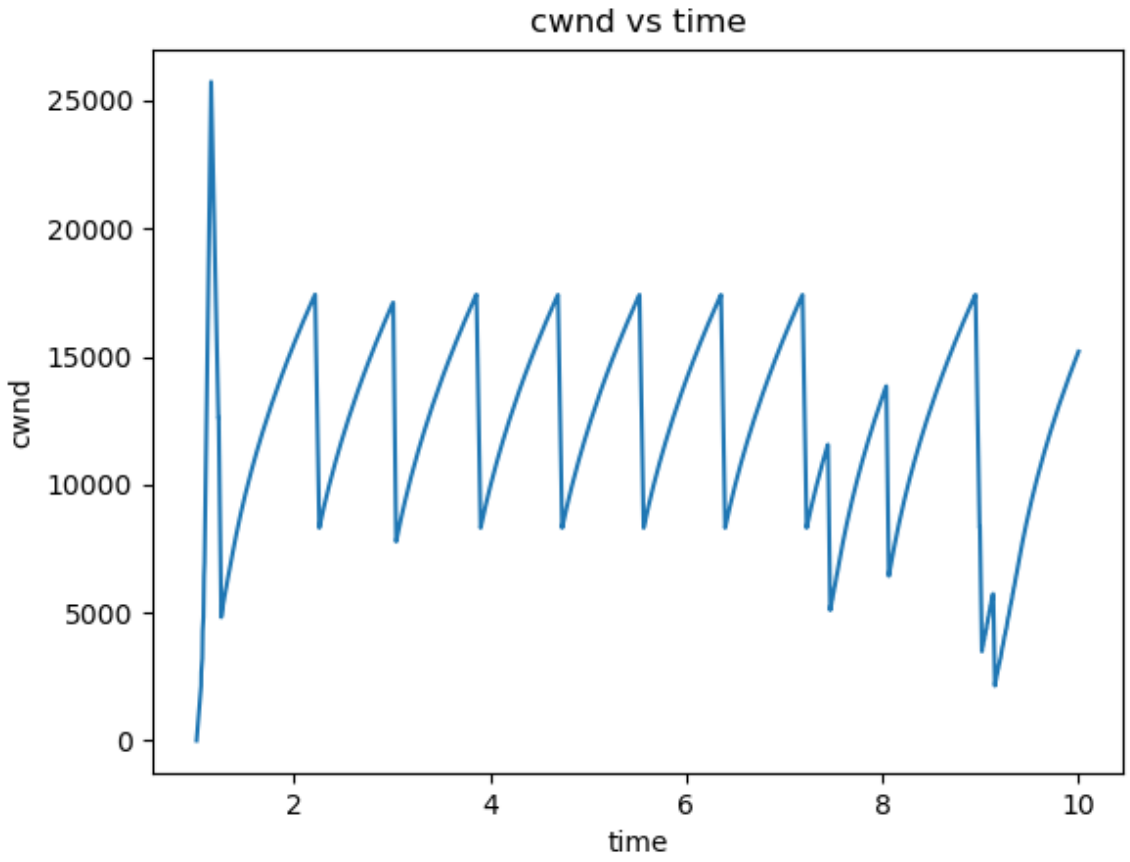
- **Is the achieved throughput approximately equal to the maximum expected value? If it is not, explain the reason for the difference.**

  Yes, the average achieved throughput is quite close to the maximum expected throughput.

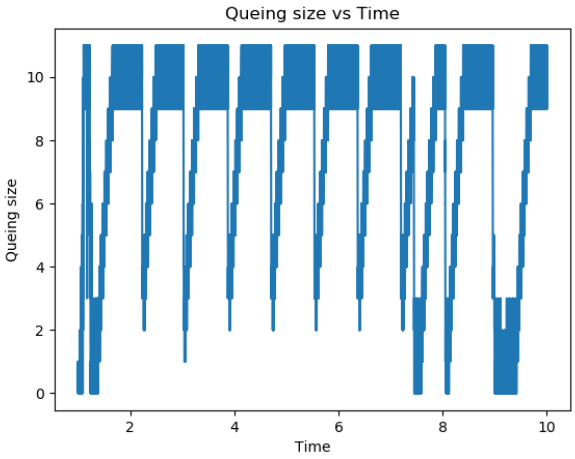
- **How many times did the TCP algorithm reduce the cwnd, and why?**

  14 times, since it detects a segment loss of a triple duplicate ACK. So it goes to half the current congested window size to reduce the congestion.

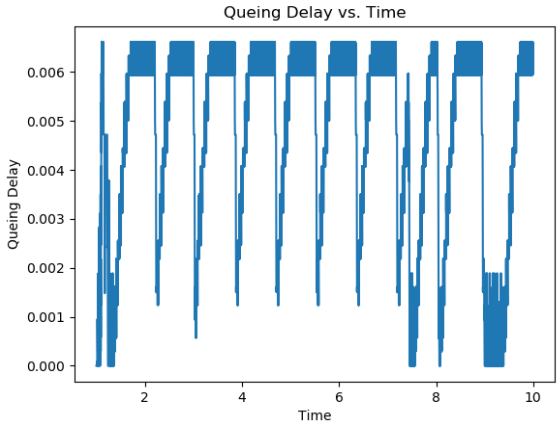
  This is based on the following graph observations:



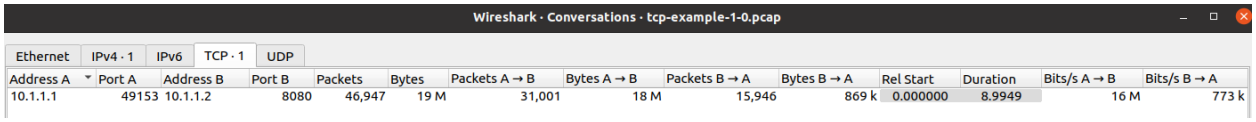
Queuing size over time graph



Queuing delay over time graph



2. Start with the default config. Change the link bandwidth to 50Mbps (from 5Mbps).



- **What is the average throughput of the TCP transfer?**

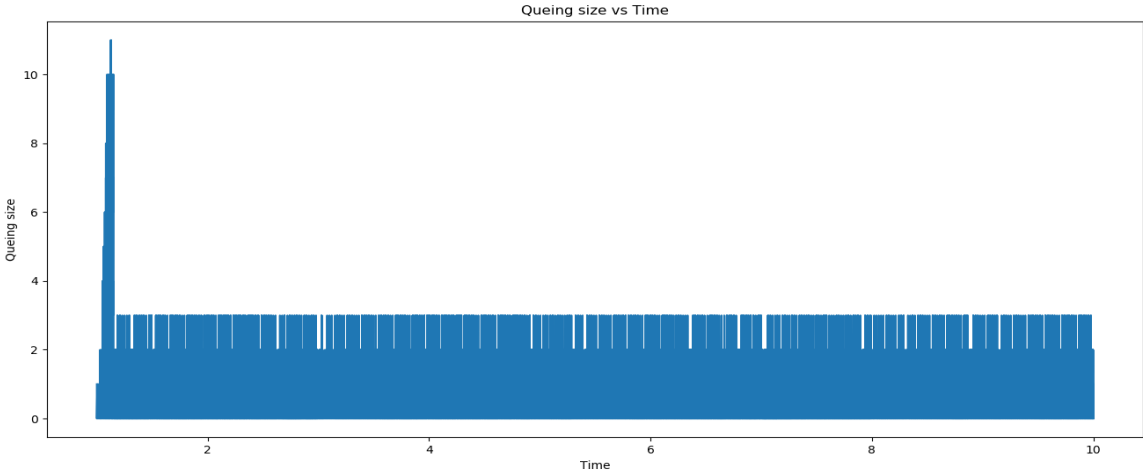
  Average receiving throughput: 16Mbps

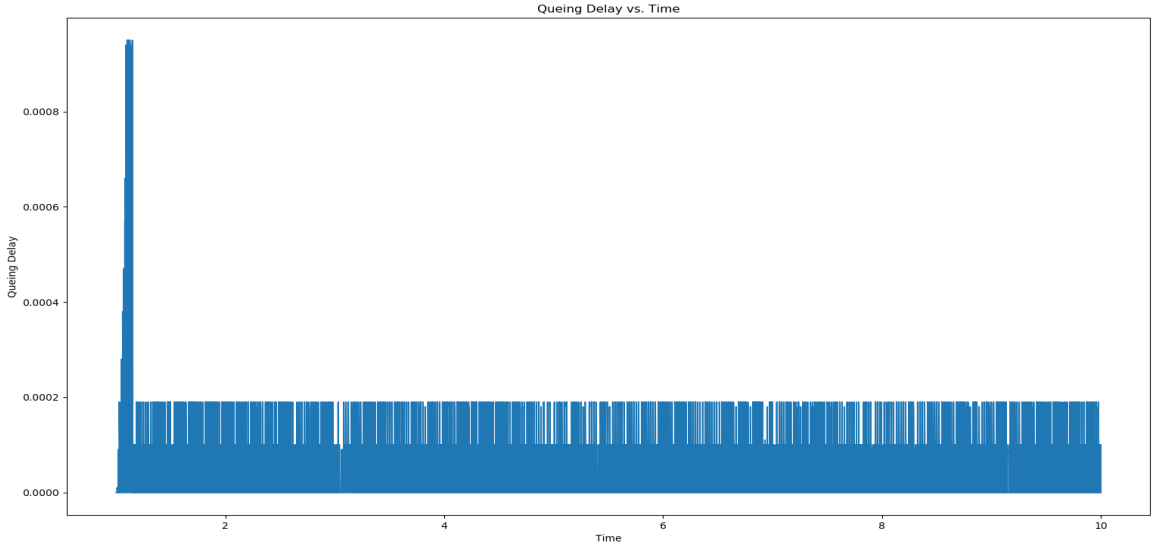
- **What is the maximum expected value of throughput?**

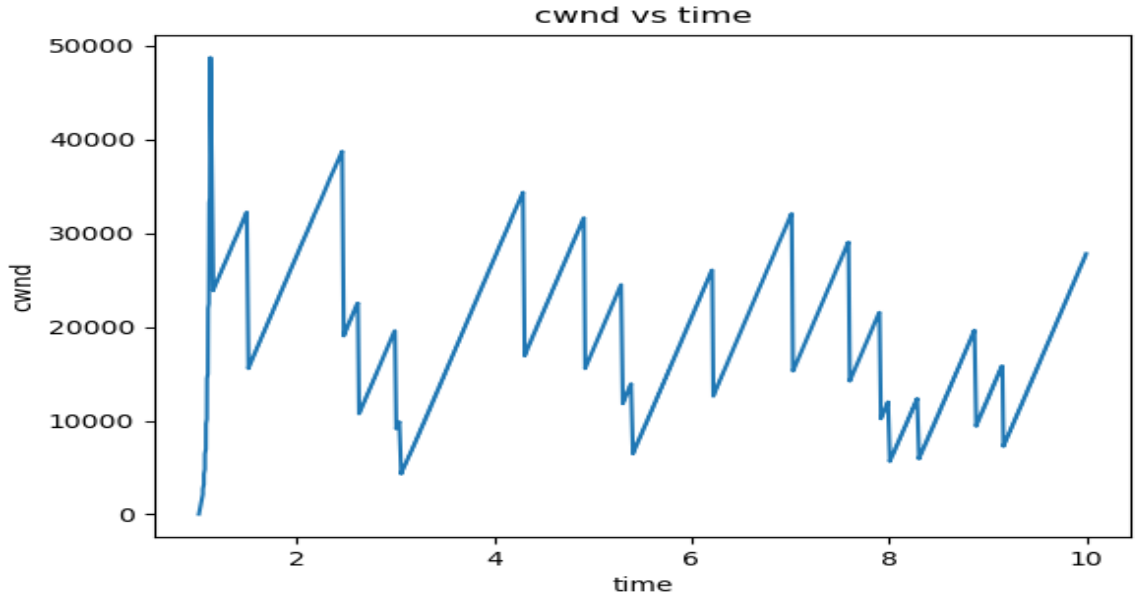
  Maximum expected value of throughput = 5Mbps

- **Is the achieved throughput approximately equal to the maximum expected value? If it is not, explain the reason for the difference.**

  No since the error rate is relatively high so the sender receives multiple three duplicate acks that prevents the cwnd from increasing and keeps decreasing to it's half, also the RTT is relatively high so the rate of cwnd increase is relatively slow. These reasons keep the throughput relatively low so it doesn't harness the available bandwidth.


Queing size vs Time
Queing size
Time
0
2
4
6
8
10


Queing Delay vs. Time
Queing Delay
Time
0.0000
0.0002
0.0004
0.0006
0.0008
2
4
6
8
10


cwnd vs time
cwnd
time
0
10000
20000
30000
40000
50000
2
4
6
8
10

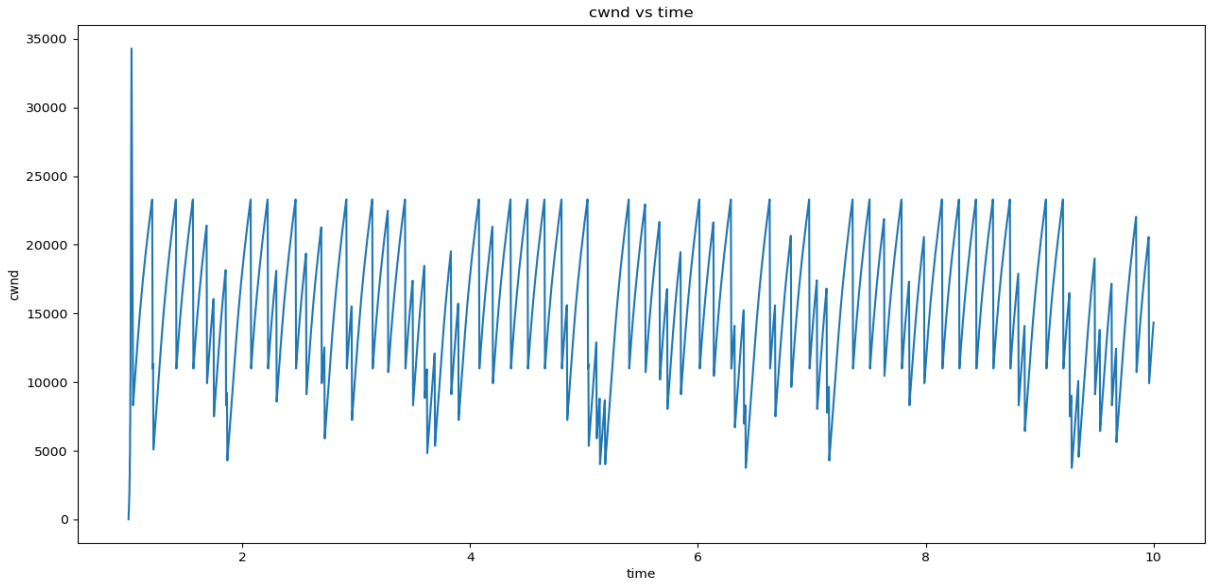
○ **What other parameters in the simulation (amongst the ones exposed to you above) can you change to make sure that the throughput is close to the maximum expected value, for this link bandwidth? (Try out a few different simulations, and see what gets you close to the maximum.)**

Decreasing the propagation delay to 2 ms

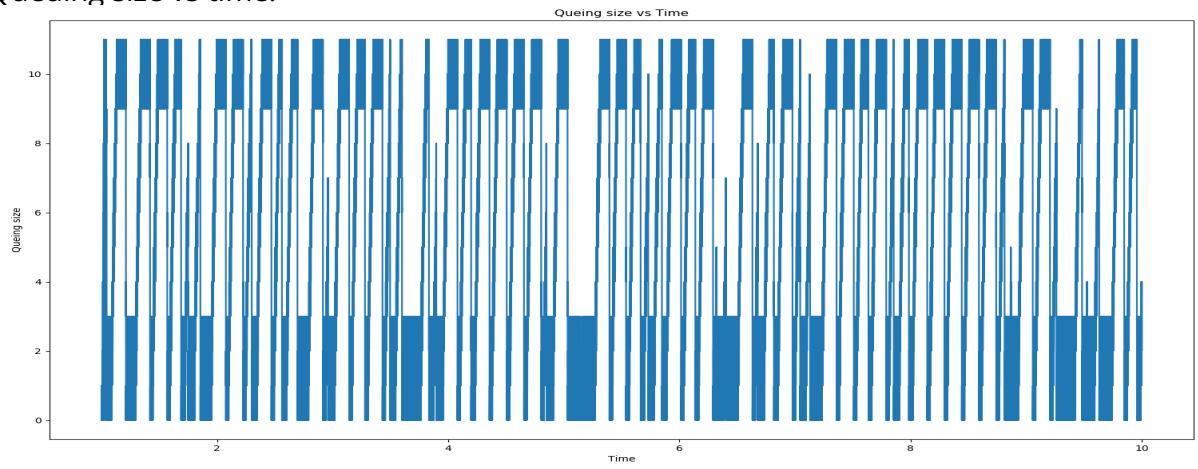
Wireshark · Conversations · tcp-example-1-0.pcap

Ethernet | IPv4 · 1 | IPv6 | TCP · 1 | UDP

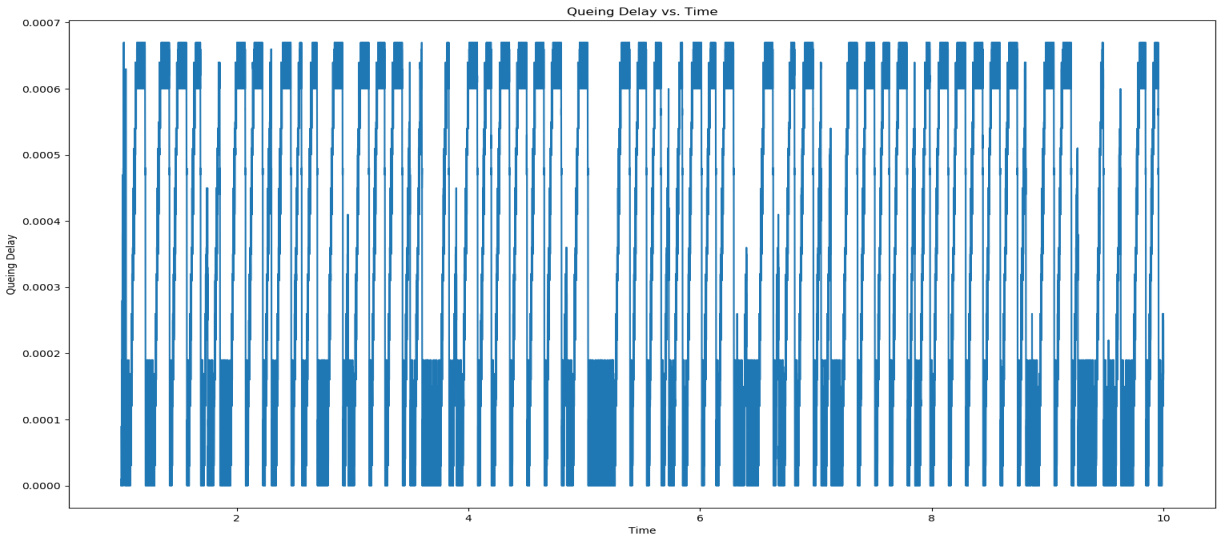
Address A	Port A	Address B	Port B	Packets	Bytes	Packets A → B	Bytes A → B	Packets B → A	Bytes B → A	Rel Start	Duration	Bits/s A → B	Bits/s B → A
10.1.1.1	49153	10.1.1.2	8080	134,100	54 M	88,376	52 M	45,724	2,494 k	0.000000	8.9989	46 M	2,217 k



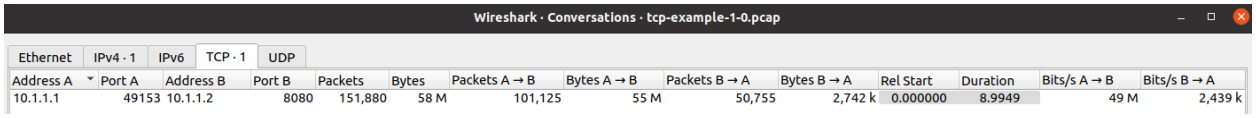
Queuing size vs time:



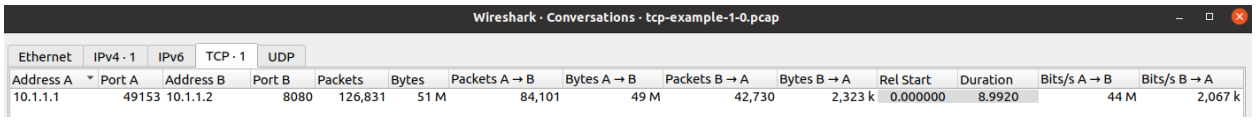
Queuing delay vs time:

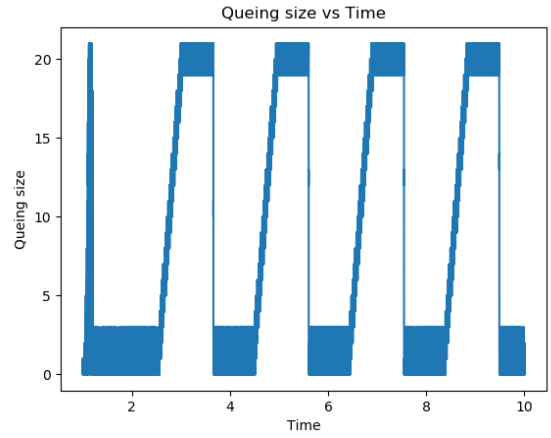


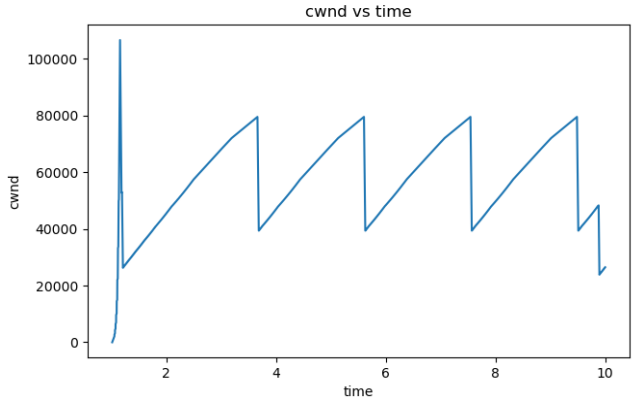
Decreasing the error rate to 0.00000001



Decreasing the error rate to 0.00000001 and decreasing propagation delay to 1ms with queuing size 20



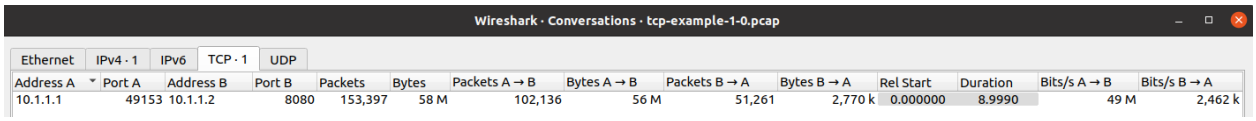




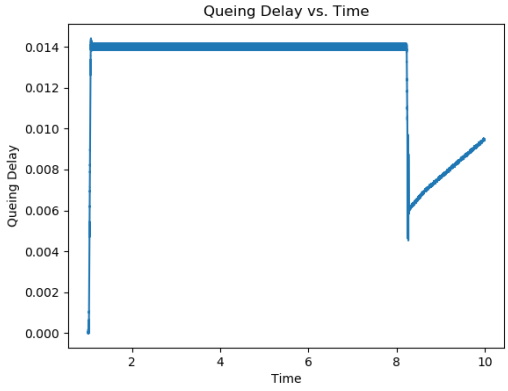
The queue keeps filling then this cause packets loss

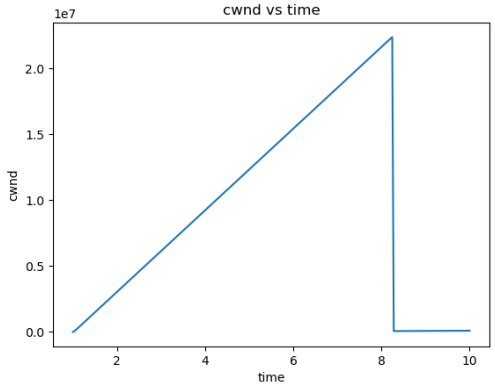
The queue keeps filling then this cause packets loss even the cwnd size keeps dropping since multiple duplicate Acks

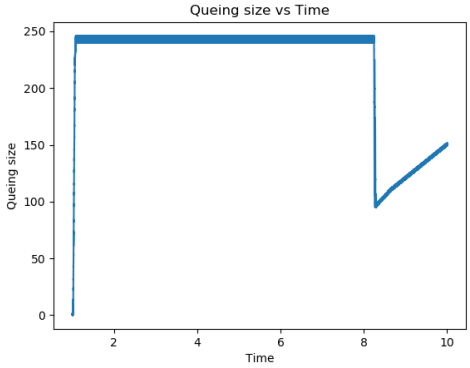
So by increasing queuing size to 1000



Throughput comes near to the maximum expected throughput (more utilization of bandwidth)

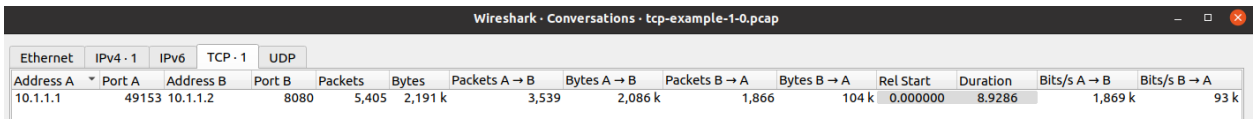






The congestion window size keeps increasing and the queue is not being fully filled

3. Start with the default config. Change the link delay to 50 ms.



- **What is the average throughput of the TCP transfer?**

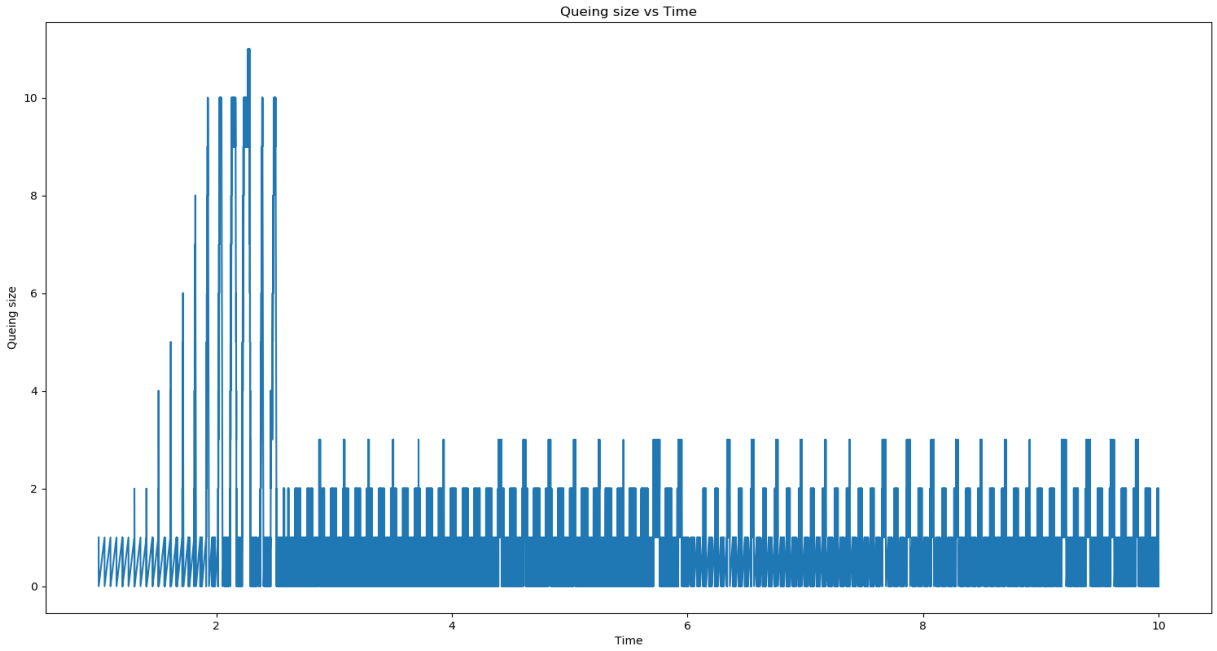
  Average receiving throughput: 1,869 Kbits/sec = 1.9Mbps

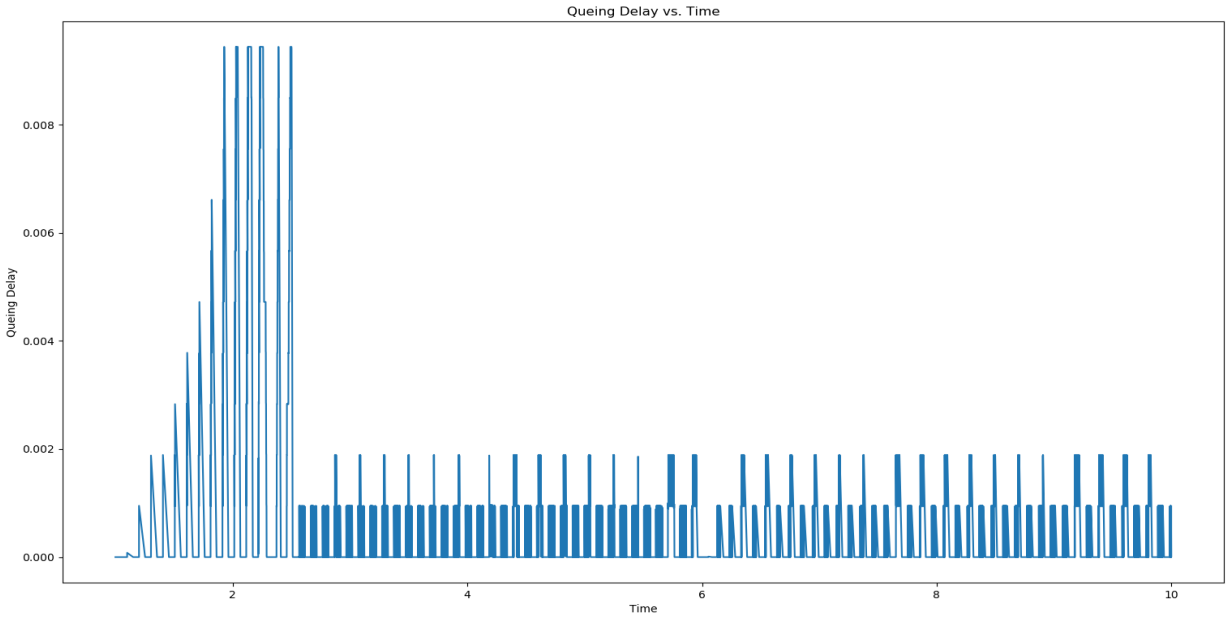
○ **What is the maximum expected value of throughput?**

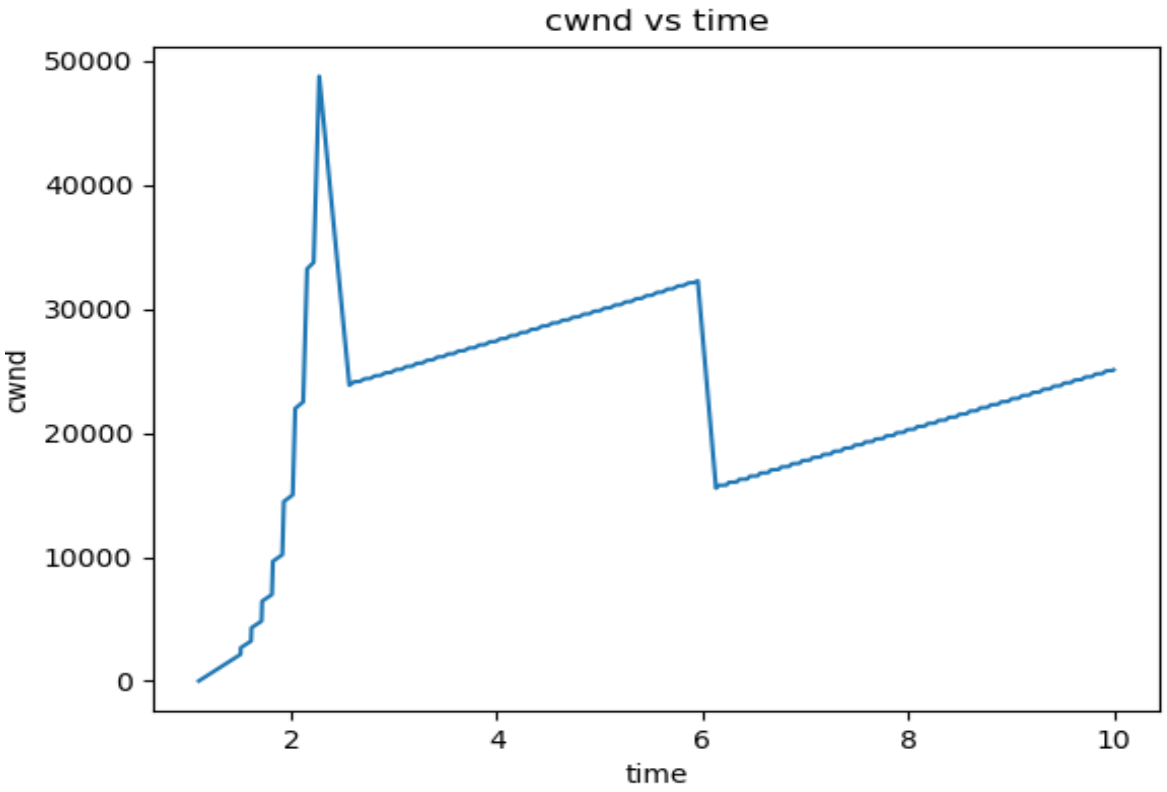
Maximum expected value of throughput = 5Mbps

○ **Is the achieved throughput approximately equal to the maximum expected value? If it is not, explain the reason for the difference.**

No since the RTT is relatively high so the rate of cwnd increase is relatively slow. These reasons keep the throughput relatively low so it doesn't harness the available bandwidth.

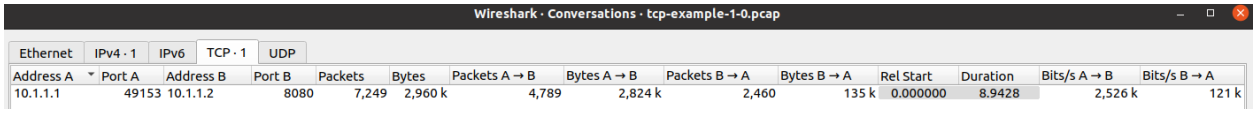




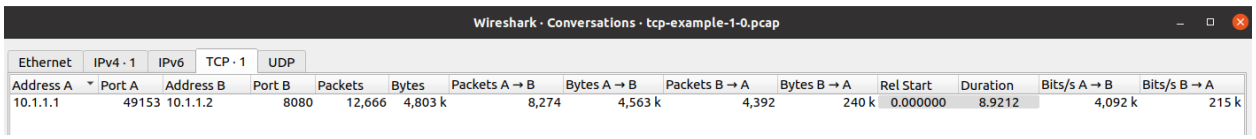


○ **What other parameters (amongst the ones exposed to you above) can you change to make sure that the throughput is close to the maximum expected value, for this link delay? (Try out a few different simulations, and see what gets you close to the maximum.)**

Decreasing the error rate to 0.000000001:

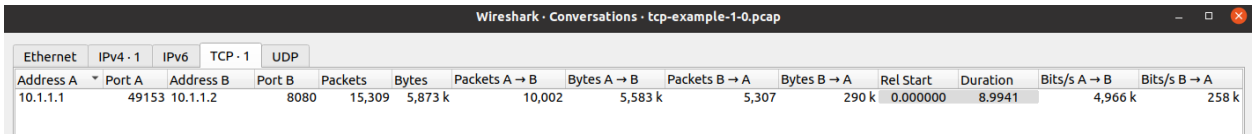


Increasing queue size to 1000:



4. Start with the default config. Change the queue size to 1000 packets.

○ **Compare the TCP throughput in both cases and explain what you see. Further, explain what happens to the queue occupancy, queueing delay, and cwnd in both cases.**

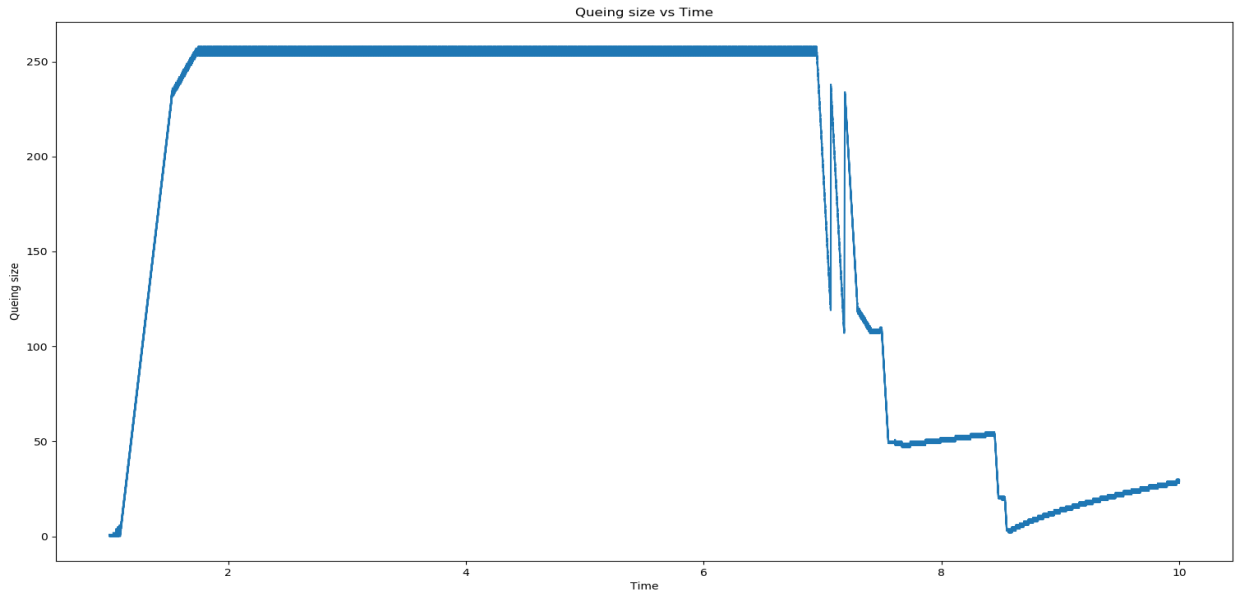


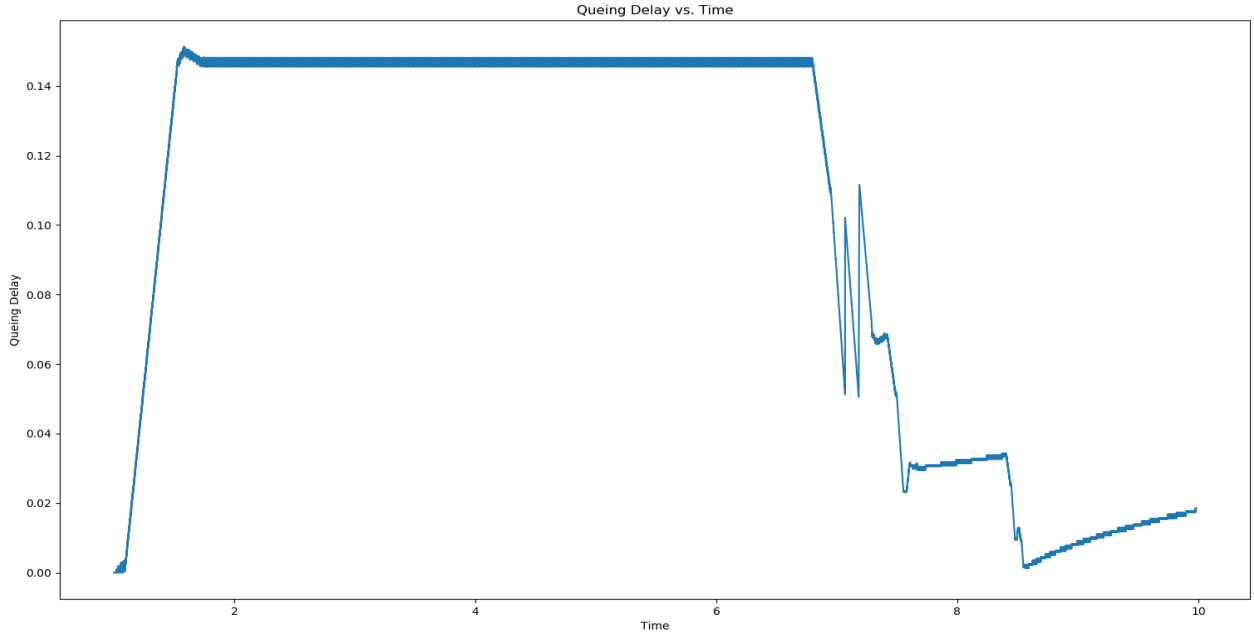
The average throughput is approximately equal since average cwnd size is also approximately equal.

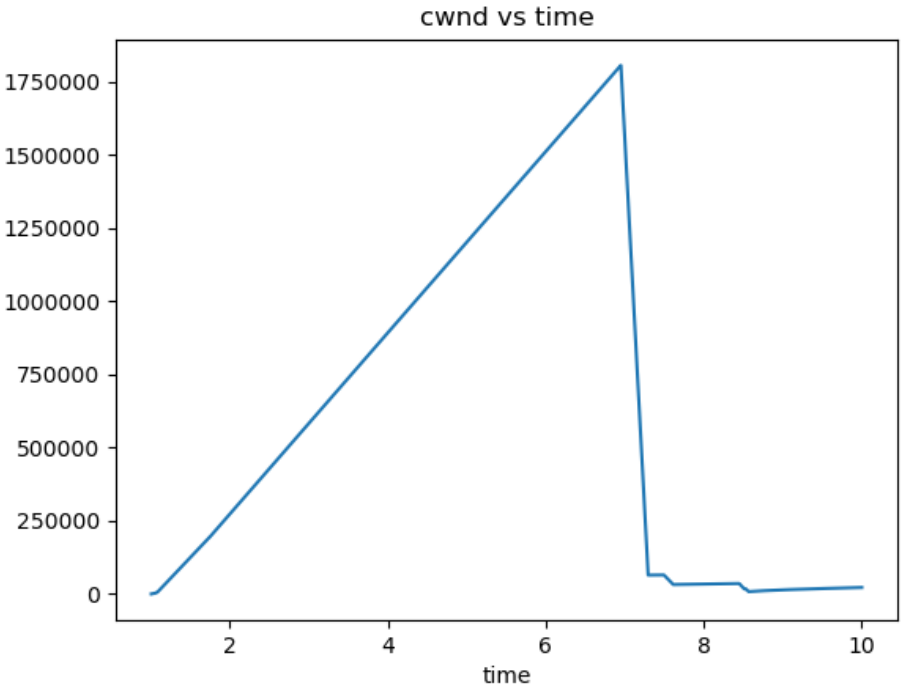
Cwnd: Increasing the queuing size opened the way for the cwnd to increase its size so I will spend more in the congestion avoidance phase by increasing the cwnd size by one each RTT, till reaching the maximum value, at this point the sender is somehow overwhelming the bandwidth so this causes packets loss which causes the sender to reach the timeout this causes the cwnd to decrease to zero and go back to the slow start phase again. However in the default configuration it oscillated between the maximum cwnd size and the minimum keeping a balanced cwnd size.

queue occupancy: The queue occupancy reaches approximately 258 and never goes beyond this number, however with the default configuration it oscillates between full and then empty and then full again.

queueing delay: like queue occupancy reaches approximately 0.18 and never goes beyond this number, however with the default configuration it oscillates between maximum and minimum queueing delay.


Queing size vs Time
Queing size
Time
0
50
100
150
200
250
2
4
6
8
10


Queing Delay vs. Time
Queing Delay
Time
0.00
0.02
0.04
0.06
0.08
0.10
0.12
0.14
2
4
6
8
10



○ **What is the optimum queue size that must be used in this simulation? Justify your answer**

The optimal queue size is 258 because by doing based on this experiment the queue occupancy doesn't go beyond this number even by decreasing the error rate to zero

Queuing size vs time with zero error rate: