Networks - Lab #2

PREPARED FOR

DR. Ayman Khalfla

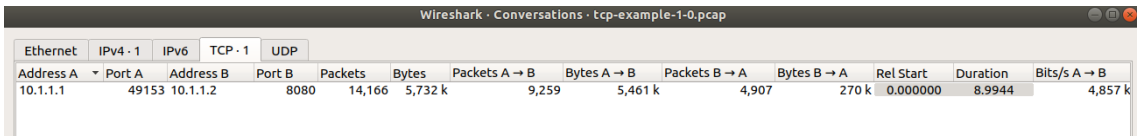
PREPARED BY

Ahmed Akram Ahmed Shawky	18010056
Rana Ayman Hussien Abdel-Razek	18010662

DEC 25 - 2021

Q1

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



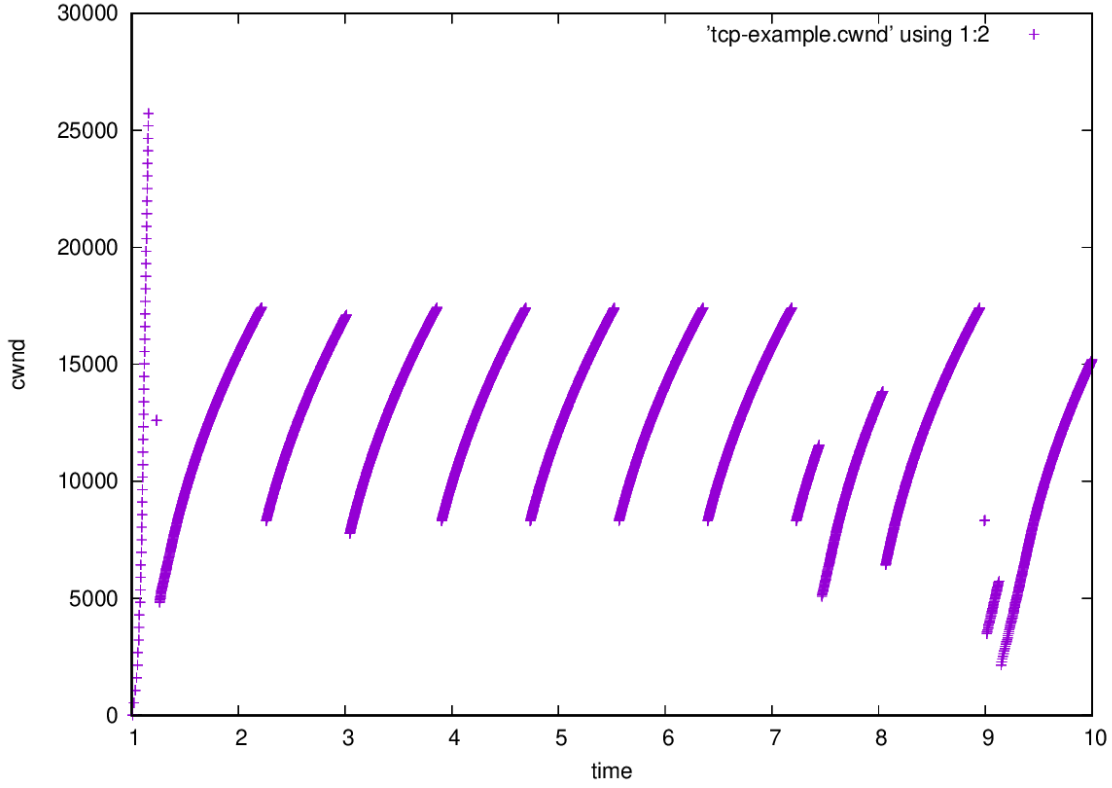
- **What is the average throughput of the TCP transfer? What is the maximum expected value of throughput? Is the achieved throughput approximately equal to the maximum expected value? If it is not, explain the reason for the difference.**

- Average receiving throughput = $\frac{5{,}461\ k}{8.9944}$ = 4,857.245 kb/sec = 4.7 Mb/sec
- Maximum expected value of throughput = 5 Mbps
- 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?**

  The size of the cwnd increases quadratically at the start after Slow Start. After this, the size increases linearly and reduces periodically. During the time of simulation, it reduces 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:


'tcp-example.cwnd' using 1:2
cwnd
time
30000
25000
20000
15000
10000
5000
0
1
2
3
4
5
6
7
8
9
10

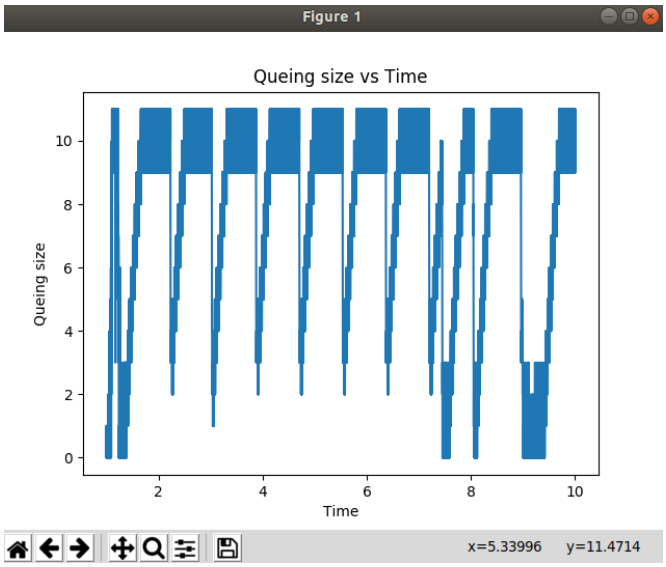


Figure 1
Queing size vs Time
Queing size
Time
x=5.33996 y=11.4714

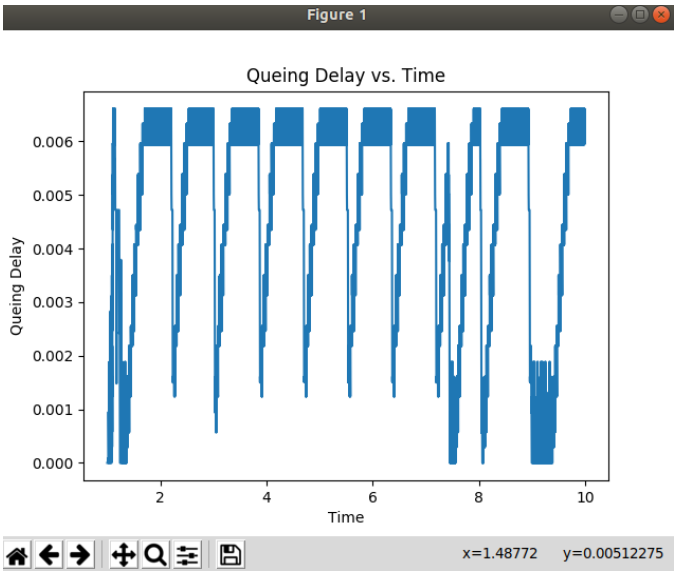
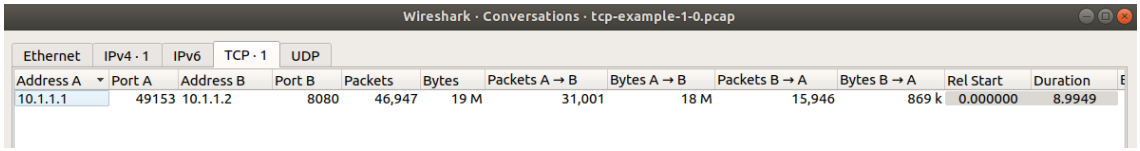


Figure 1
Queing Delay vs. Time
Queing Delay
Time
x=1.48772 y=0.00512275

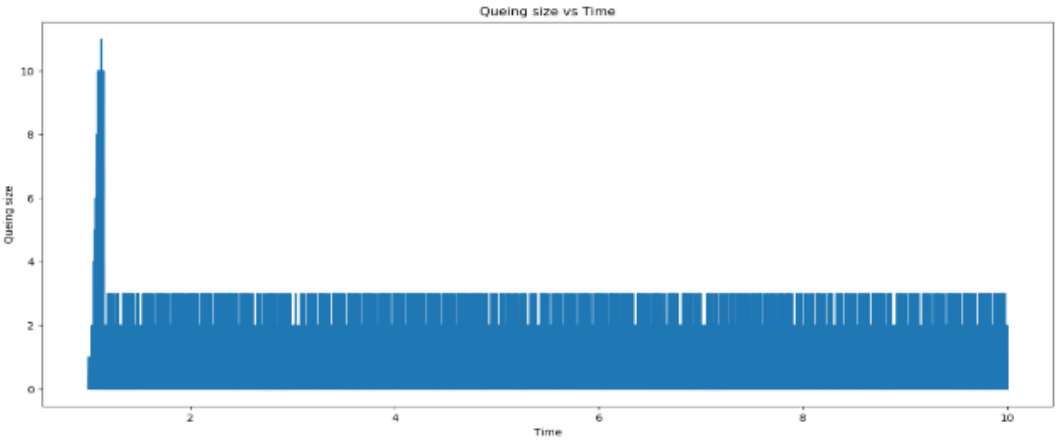
Q2

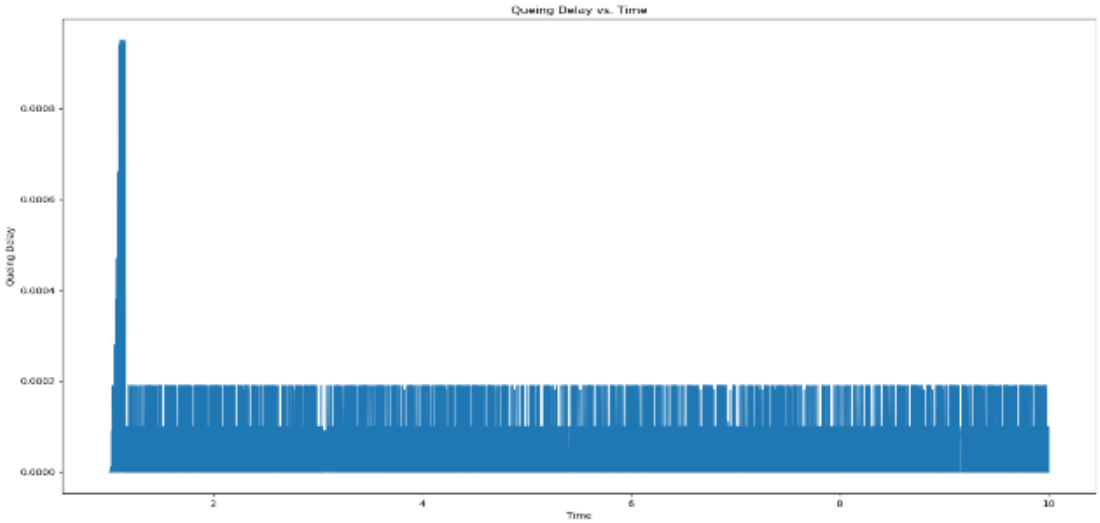
2. Start with the default config. Change the link bandwidth to 50Mbps (from 5 Mbps).



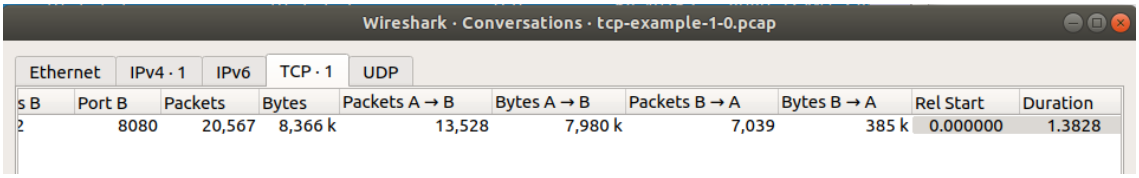
- **What is the average throughput of the TCP transfer? What is the maximum expected value of throughput? Is the achieved throughput approximately equal to the maximum expected value? If it is not, explain the reason for the difference.**

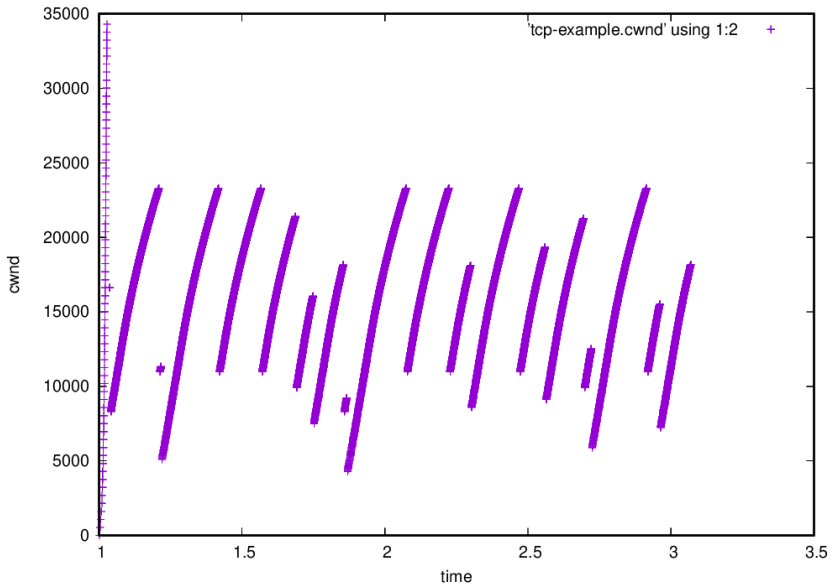
    - Average receiving throughput: 16Mbps
    - Maximum expected value of throughput = 50Mbps
    - No since the error rate is relatively high so the sender receives multiple three duplicate acks that prevents the cwnd from increasing and keeps decreasing to it's half, also the RTT is relatively high so 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

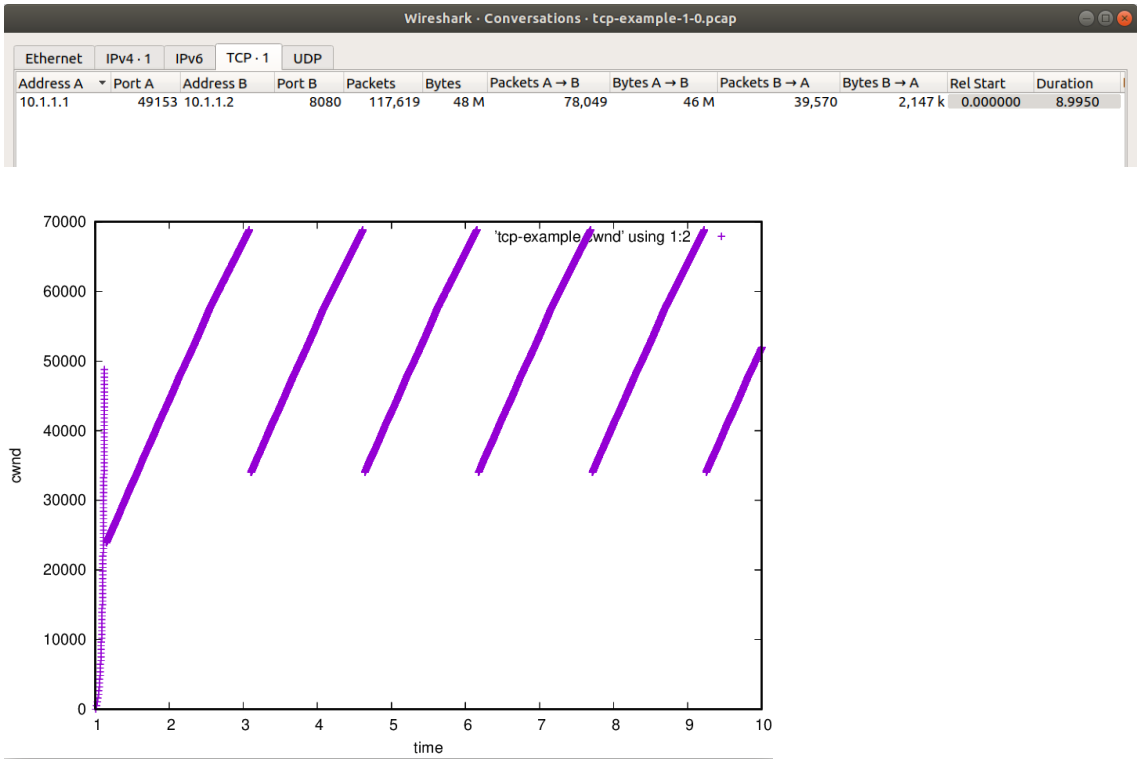
- **What other parameters in the simulation (amongst the ones exposed to you above) can you change to make sure that the throughput is close to the maximum expected value, for this link bandwidth? (Try out a few different simulations, and see what gets you close to the maximum.)**
    - At delay = 1ms
    - Average receiving throughput: 45.1Mbps
    - This shows that at a delay of 1ms, TCP spends much less time in congestion avoidance than at 5ms, which leads to a much higher throughput, since the average size of cwnd remains higher and the link is utilised much better. Thus, we can say that the throughput is lower at the higher latency because every time after fast recovery, TCP doesn't receive ACKs fast enough to increase cwnd quickly.





- At error rate = 0.00000001

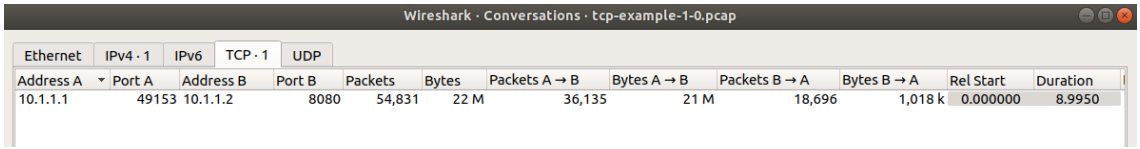
  Average receiving throughput: 40.9Mbps

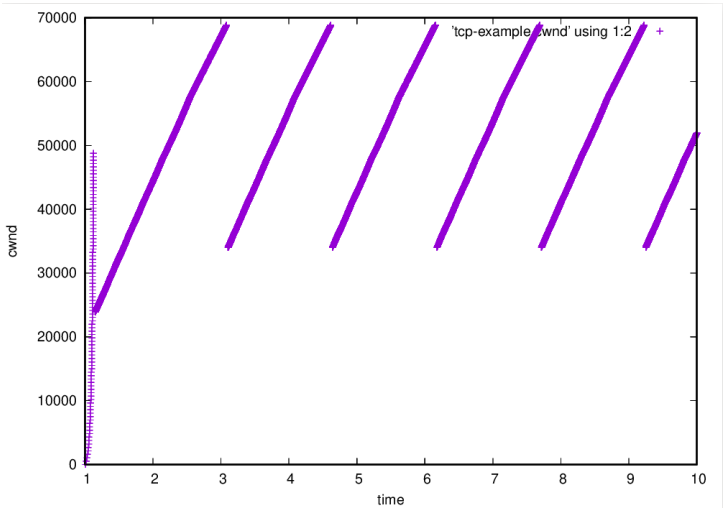


- At Queue size = 1000

  Average receiving throughput: 18.677Mbps

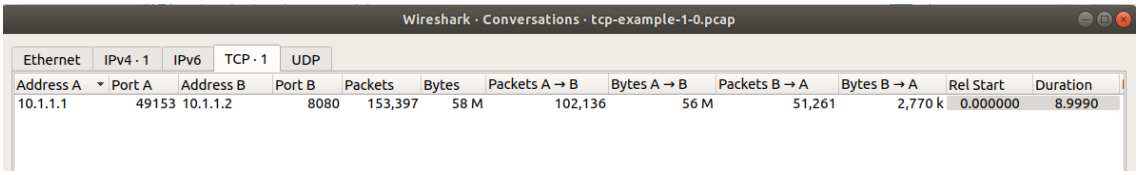
  IIncreasing queue size keeps everything else the same, however, doesn't have any effect on the throughput, since there's almost no queue anyway.

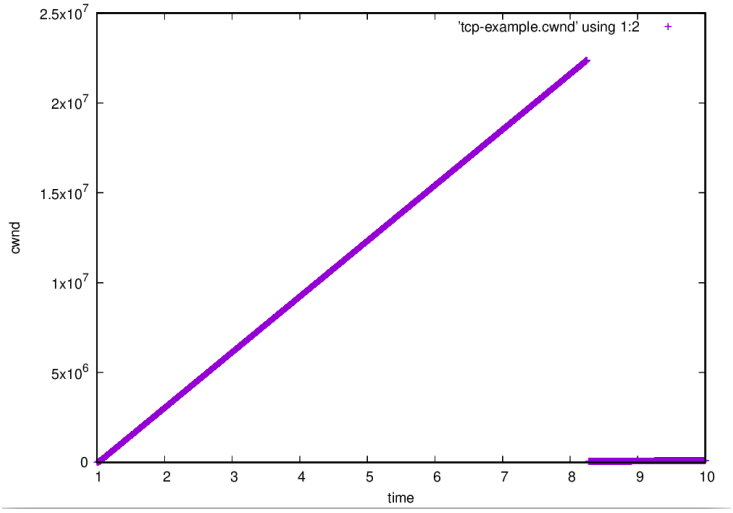




- At delay =1ms & Queue size = 1000 & error rate = 0.00000001

  Average receiving throughput: 49.78Mbps





Q3

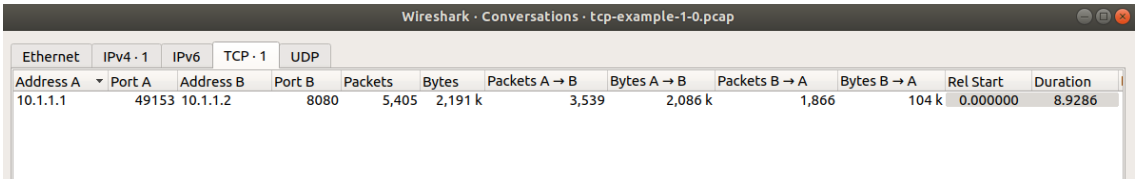
3. Start with the default config. Change the link delay to 50 ms.

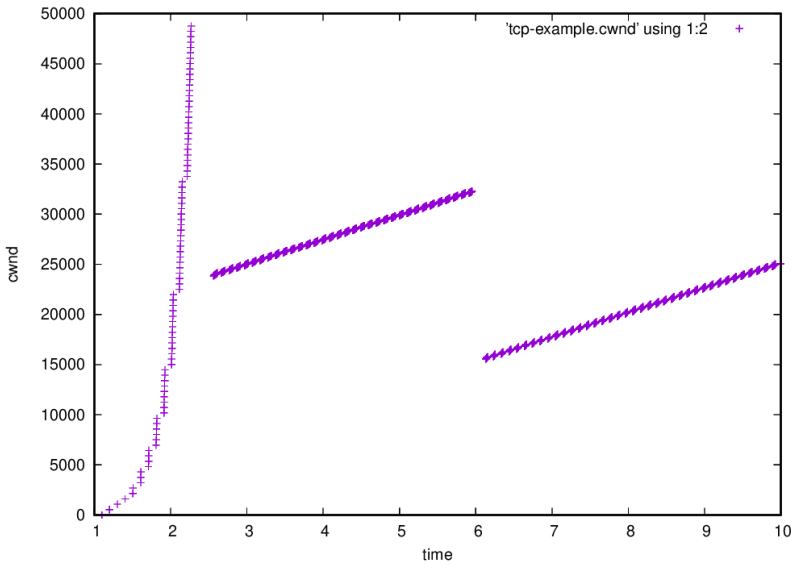
- **What is the average throughput of the TCP transfer? What is the maximum expected value of throughput? Is the achieved throughput approximately equal to the maximum expected value? If it is not, explain the reason for the difference.**

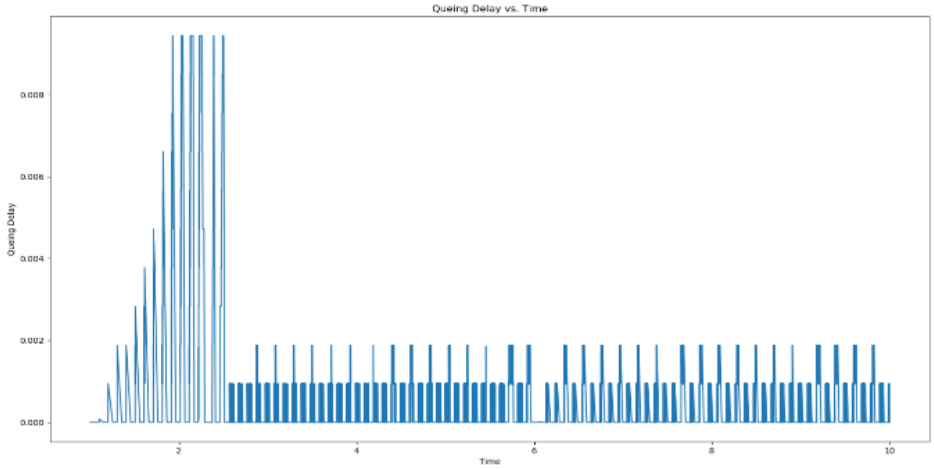
  Average receiving throughput: 1.825Mbps

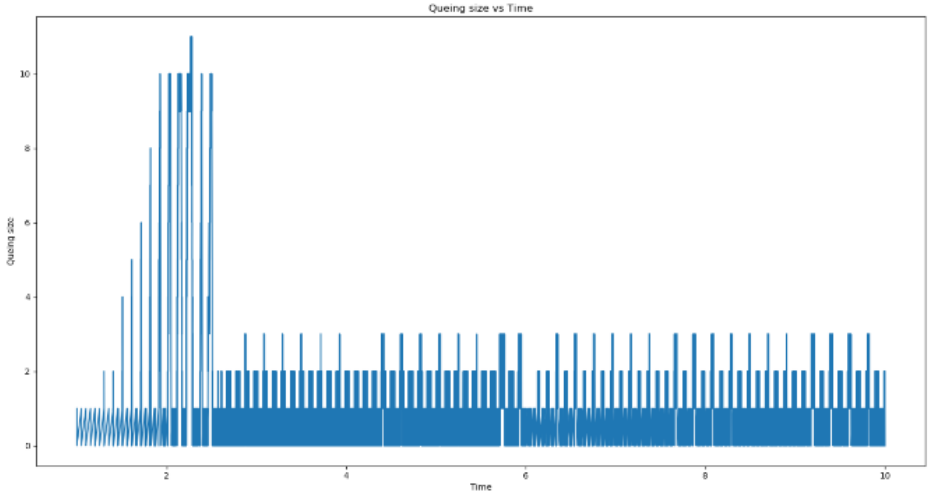
  Maximum expected value of throughput = 5Mbps

  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.




'tcp-example.cwnd' using 1:2
cwnd
time
0
5000
10000
15000
20000
25000
30000
35000
40000
45000
50000
1
2
3
4
5
6
7
8
9
10

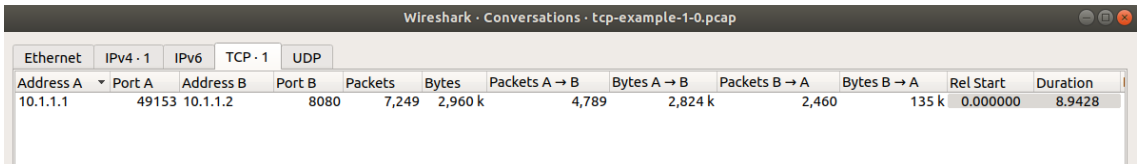

Queing Delay vs. Time
Queing Delay
Time
0.000
0.002
0.004
0.006
0.008
2
4
6
8
10

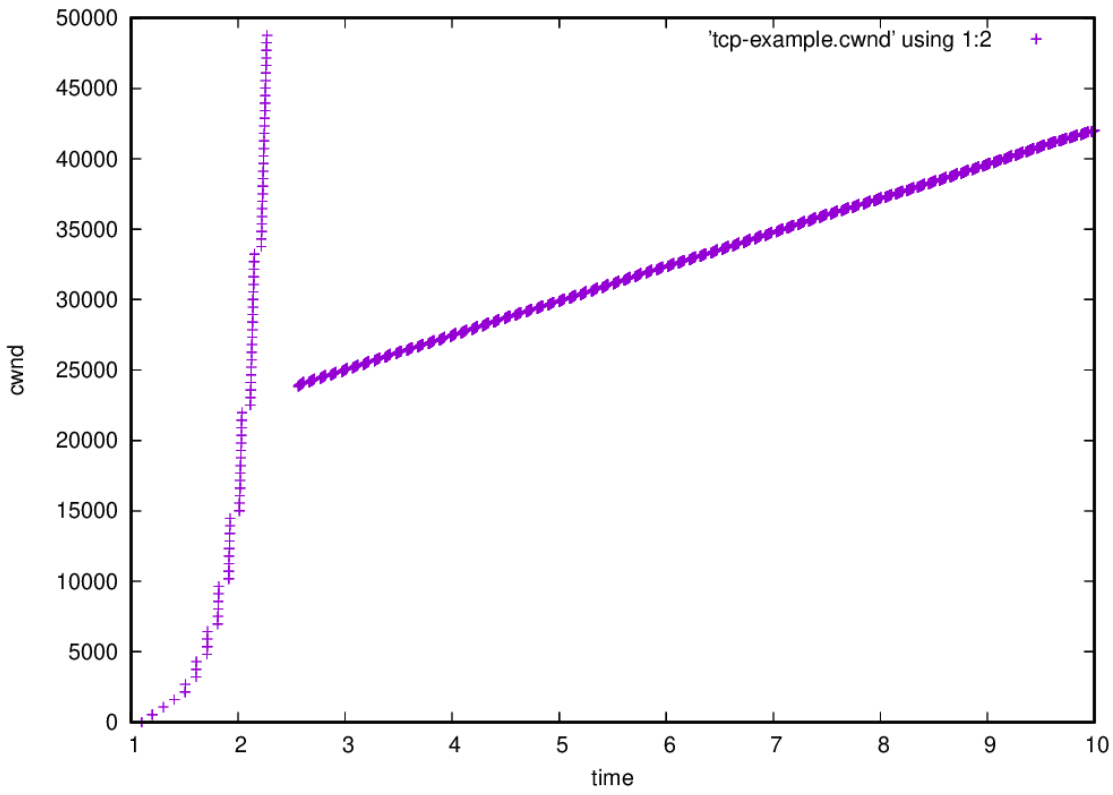

Queing size vs Time
Queing size
Time
0
2
4
6
8
10
2
4
6
8
10

- **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.)**

- At error rate = 0.000000001

  Average receiving throughput: 2.467Mbps





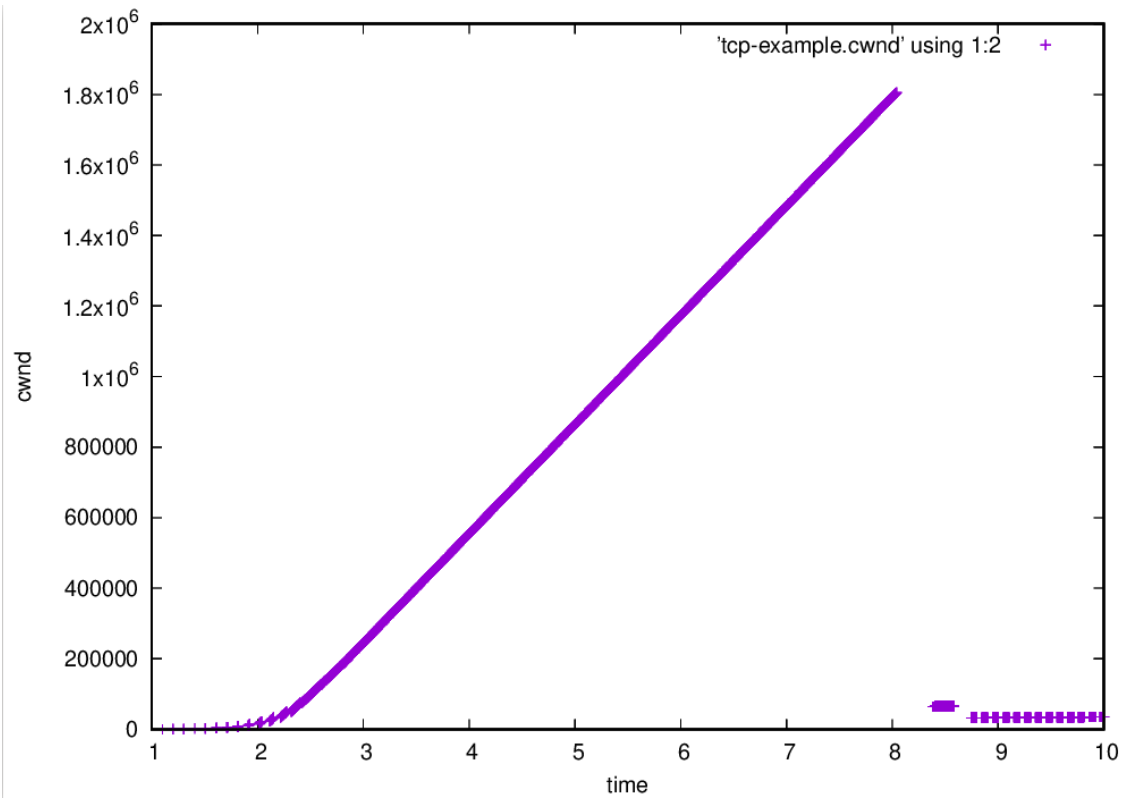
- At queue size = 1000

  Average receiving throughput: 3.9959Mbps

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
10.1.1.1	49153	10.1.1.2	8080	12,666	4,803 k	8,274	4,563 k	4,392	240 k	0.000000	8.9212



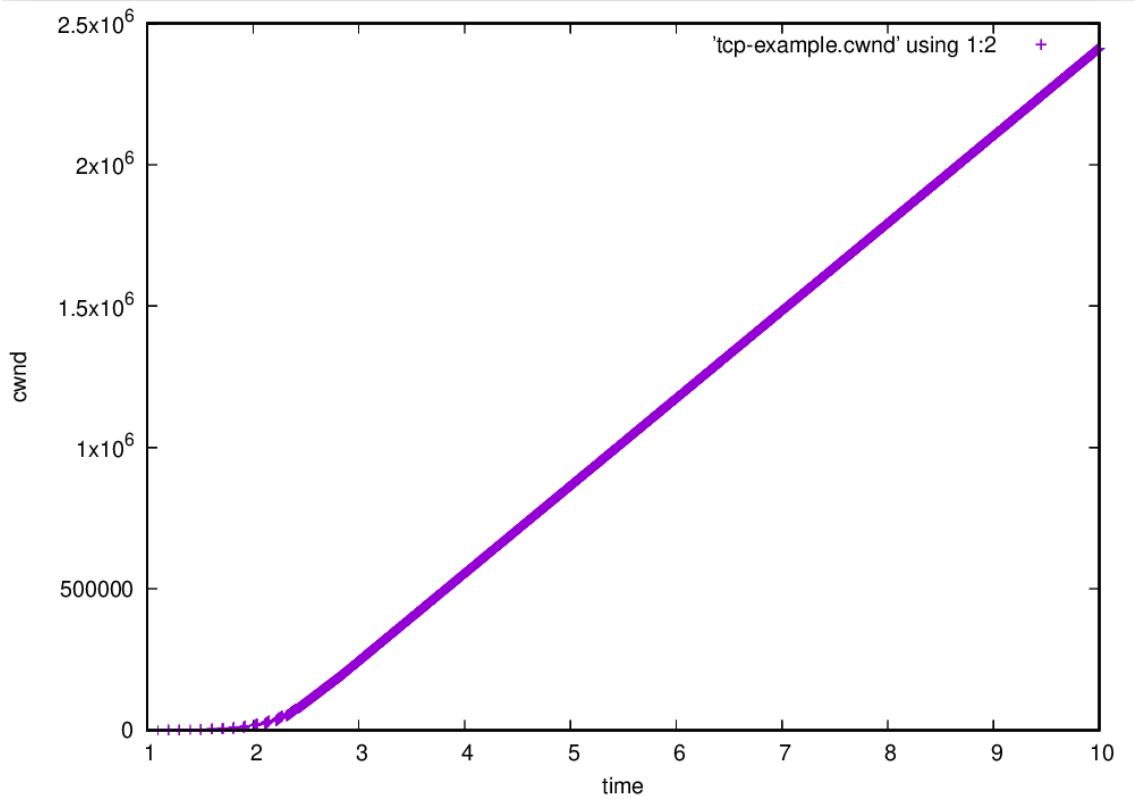
- At error rate = 0.000000001 and queue size = 1000

  Average receiving throughput: 4.307Mbps

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
10.1.1.1	49153	10.1.1.2	8080	13,595	5,178 k	9,063	4,934 k	4,532	244 k	0.000000	8.9493



Q4

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.**

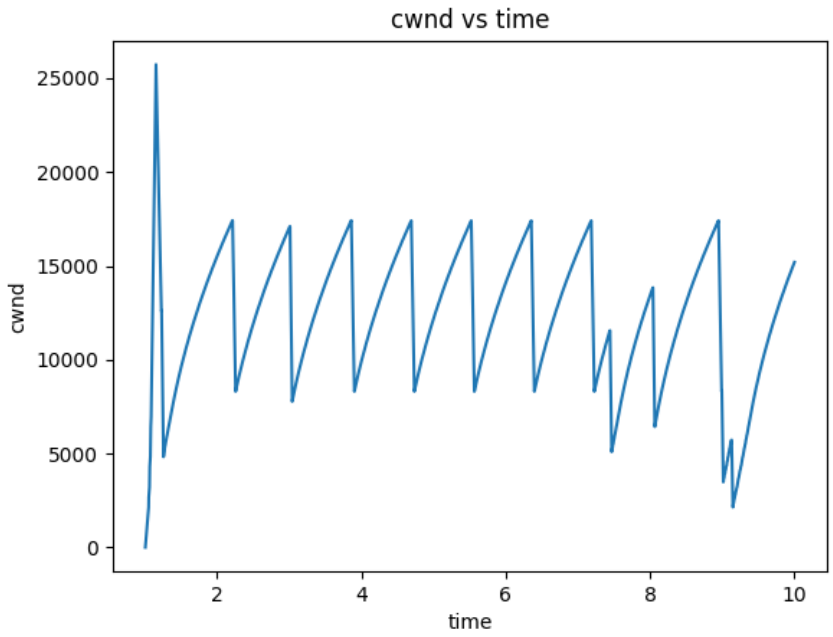
- The average throughput is approximately equal since average cwnd size is also approximately equal.
- Cwnd: Increasing the queuing size opened the way for the cwnd to increase its size so I will spend more in the congestion avoidance phase by increasing the cwnd size by one each RTT, till reaching the 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.

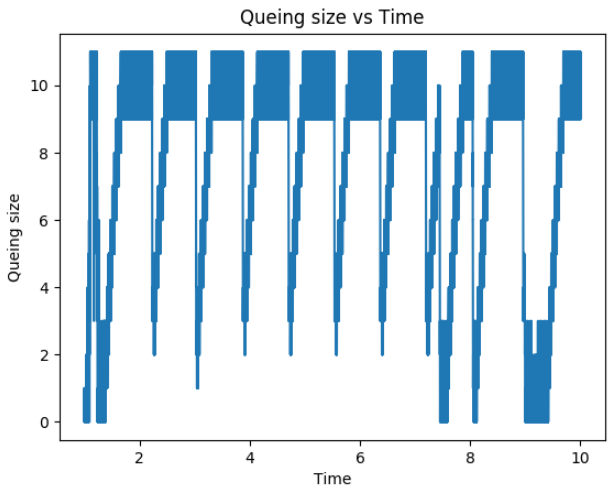
- At Default config.

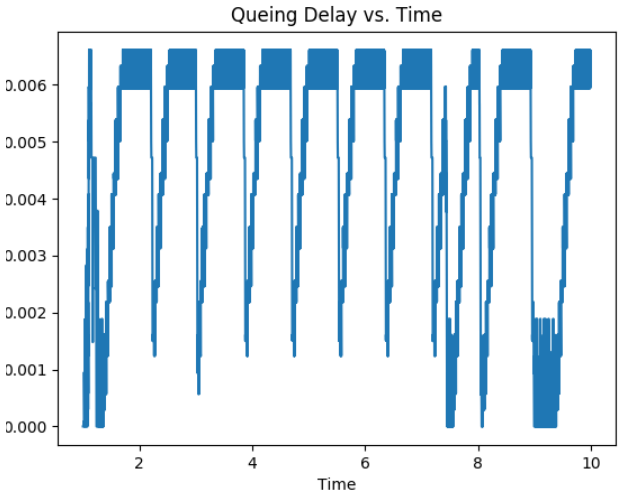
Wireshark · Conversations · tcp-example-1-0.pcap

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

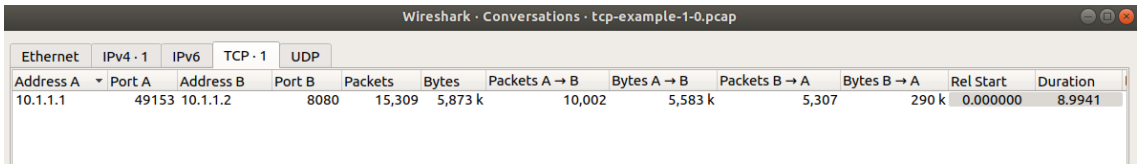
Address A	Port A	Address B	Port B	Packets	Bytes	Packets A → B	Bytes A → B	Packets B → A	Bytes B → A	Rel Start	Duration
10.1.1.1	49153	10.1.1.2	8080	14,166	5,732 k	9,259	5,461 k	4,907	270 k	0.000000	8.9944

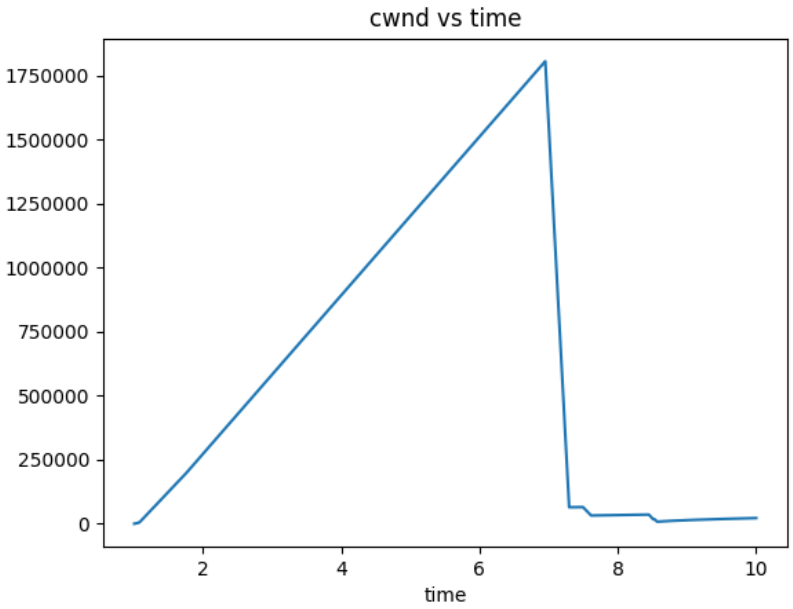


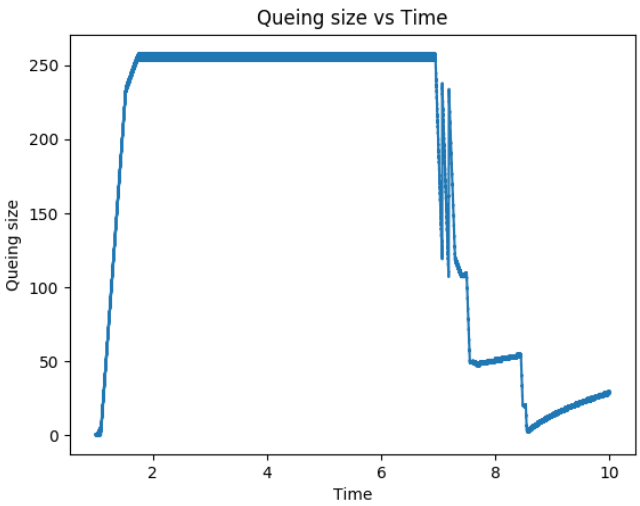


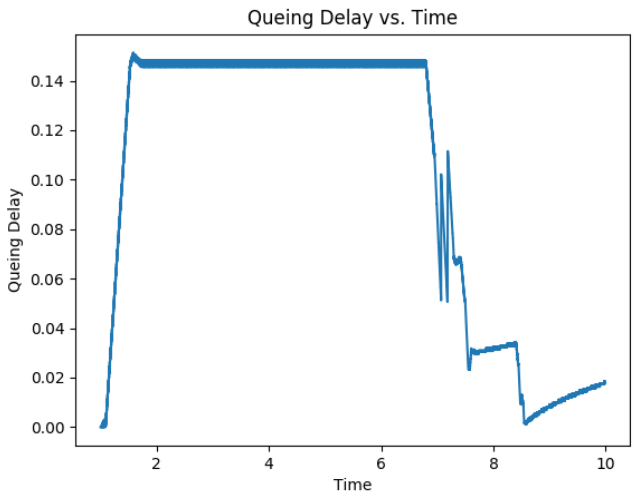


- At Queue size = 1000









- **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: