Computer Networks
Topic: Network Simulation using Ns3

1.
A.
Ans. The maximum expected throughput value should be **5Mbps**, because it would create a bottleneck between the end-end throughput as bandwidth of n0-n1>n1-n2. So the lower value creates the bottleneck and hence is itself the maximum throughput possible which in our case is 5Mbps.
B.
Ans. Max Bandwidth = 5Mbps
Total delay (delay of N0-N1 + N1-N2) = 10ms+15ms = 25ms
RTT = 2*total delay = 2*25 = 50ms
Bandwidth Delay Product = total bandwidth*total delay

⇒ 5Mbps*50ms ⇒ 250kb
Application Payload Size = 1460 Bytes
No. of packets = bandwidth/application payload size
⇒ 250000 / (1460*8) ⇒ **21.4 packets**

C.

Wireshark · Conversations · tcp-example-2-0.pcap

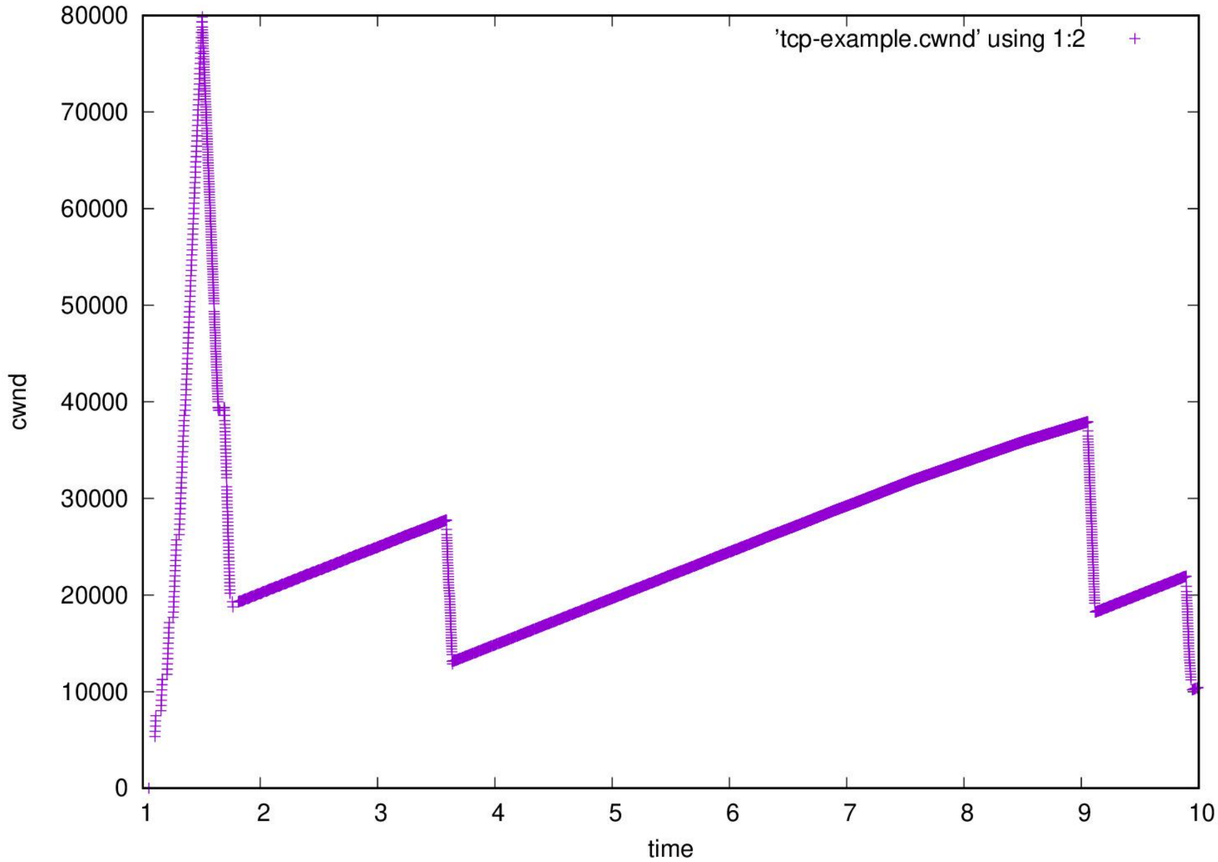
Ethernet | IPv4 · 1 | IPv6 | TCP · 1 | UDP

Address A	Address 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	10.1.2.2	11,349	4,587k	7,408	4,369k	3,941	217k	0.000000	8.9737	3,895k	

Here, as we can see the Avg Throughput is 3895k bits/s.

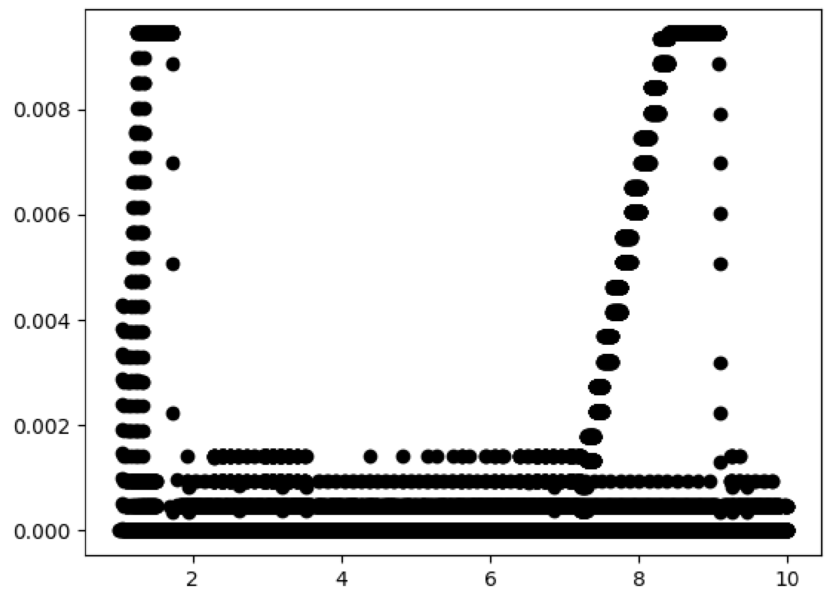
D.
Ans. No, the achieved throughput is not equal to maximum expected throughput as there are many hidden delays that act during actual implementation i.e queuing delay, transmission delay, propagation delay, network congestion etc. Due this reasons the achieved output always remain less than theoretical computed values.

E.



'tcp-example.cwnd' using 1:2
cwnd
time
0
10000
20000
30000
40000
50000
60000
70000
80000
1
2
3
4
5
6
7
8
9
10

F.

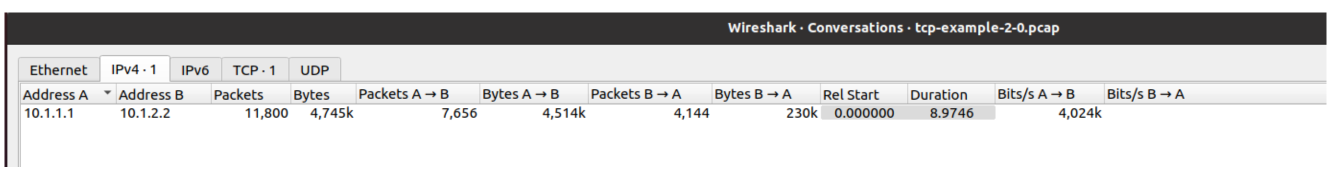


G.

Ans. One of the observations that we can see from above graphs is that the queuing delay also increases when cwnd increases. This is also relevant from a theoretical point of view that when congestion increases ,queuing delay will also increase and that is signified by the plots as well.

2.

A.

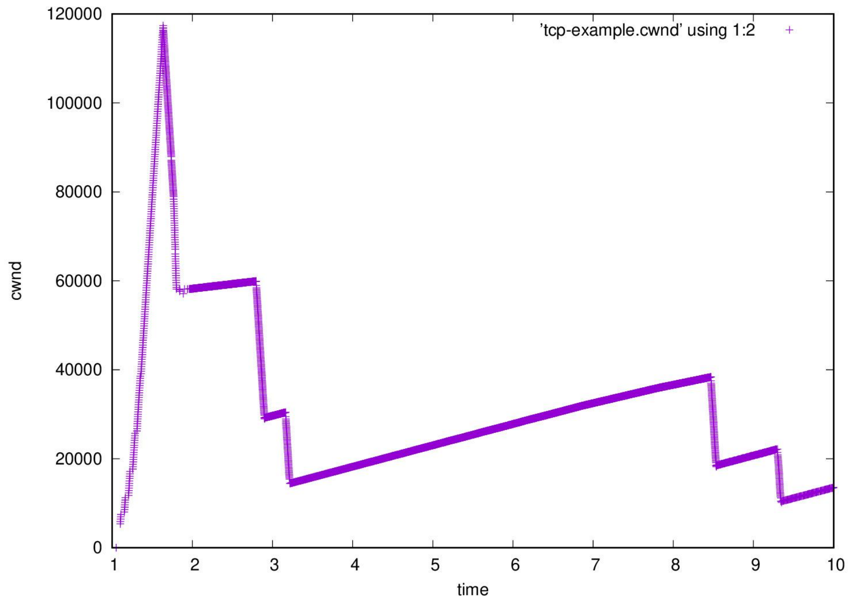

Wireshark · Conversations · tcp-example-2-0.pcap

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

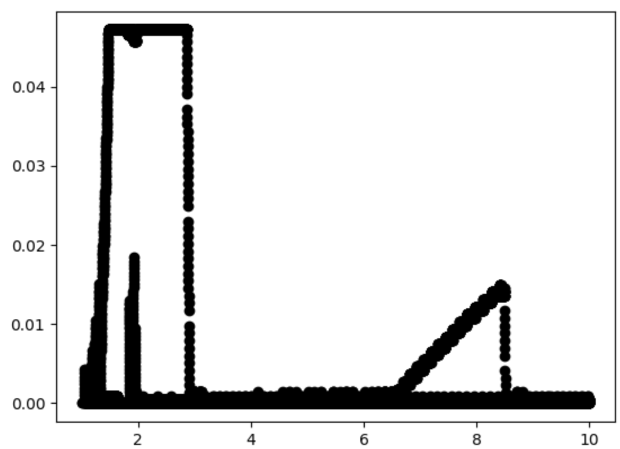
Address A	Address B	Packets	Bytes	Packets A → B	Bytes A → B	Packets B → A	Bytes B → A	Rel Start	Duration	Bits/s A → B	Bits/s B → A
10.1.1.1	10.1.2.2	11,800	4,745k	7,656	4,514k	4,144	230k	0.000000	8.9746	4,024k	

Here, as we can see the Avg Throughput is 4074k bits/s.

B.



C.



D.

Ans. By increasing the queue size to 50 we are also increasing the size of the congestion window. So from the graph of Q1, we can see that cwnd size is greater in q2 plot as compared to Q1.

3

A.

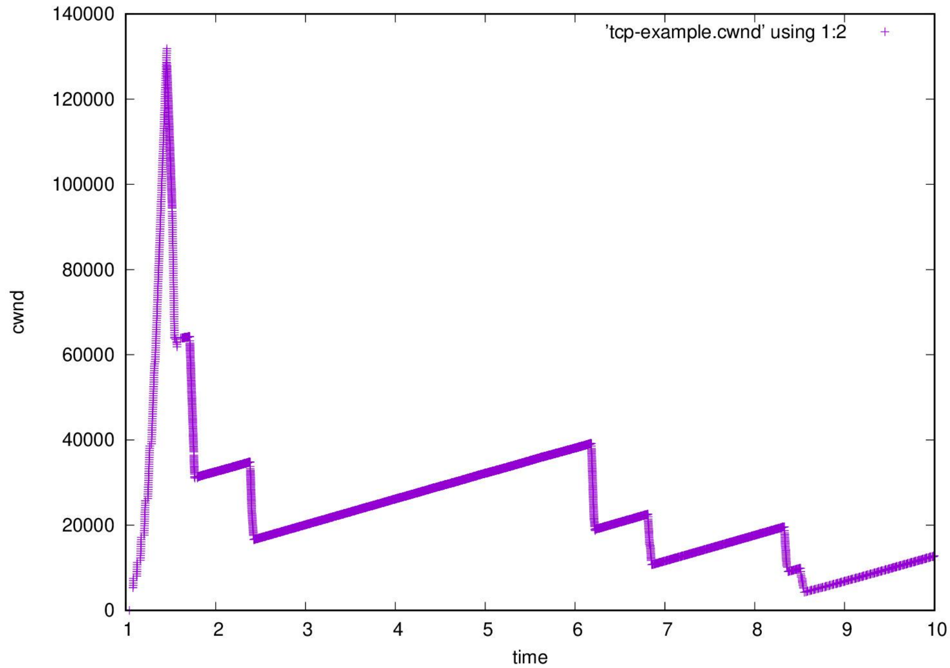
Wireshark · Conversations · tcp-example-2-0.pcap

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

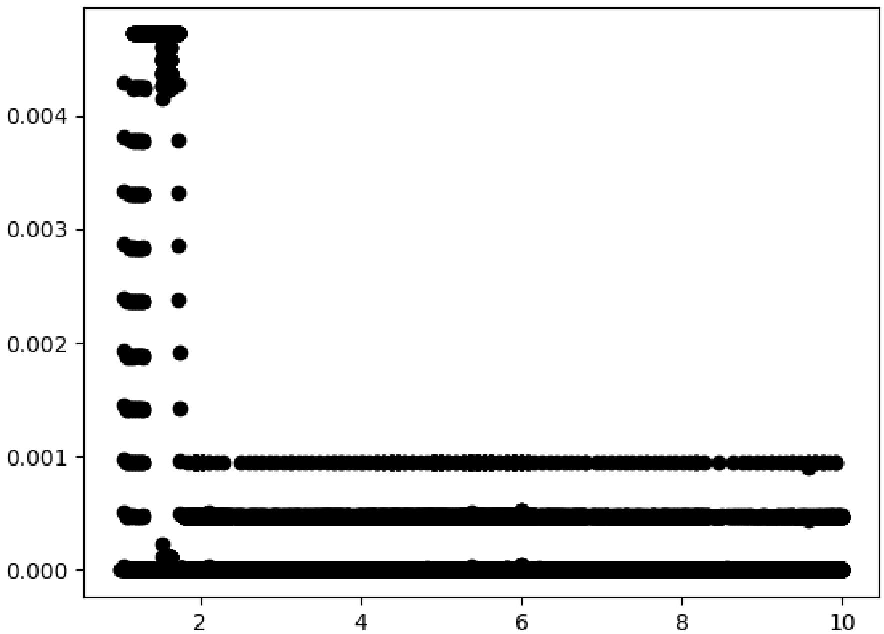
Address A	Address B	Packets	Bytes	Packets A → B	Bytes A → B	Packets B → A	Bytes B → A	Rel Start	Duration	Bits/s A → B	Bits/s B → A
10.1.1.1	10.1.2.2	13,868	5,562k	8,981	5,292k	4,887	270k	0.000000	8.9746	4,717k	

Here, as we can see the Avg Throughput is 4717k bits/s.

B.


'tcp-example.cwnd' using 1:2
cwnd
time
140000
120000
100000
80000
60000
40000
20000
0
1
2
3
4
5
6
7
8
9
10

C.



D.

Ans in this question as we have N1-N2 , as we have increased the bandwidth to 10Mbps and reduced the delay to 10ms. The configuration of both the nodes become the same and will also result in higher bandwidth as there would be any bottleneck bandwidth as both the nodes will have the same bandwidth. So this will result in higher transmission rate and congestion at end of N1 will be significantly decreased. The same can also be seen in the plots of q3 and q1 , in which the plot of q3 has an increased cwnd size as compared to q1.

4

A.

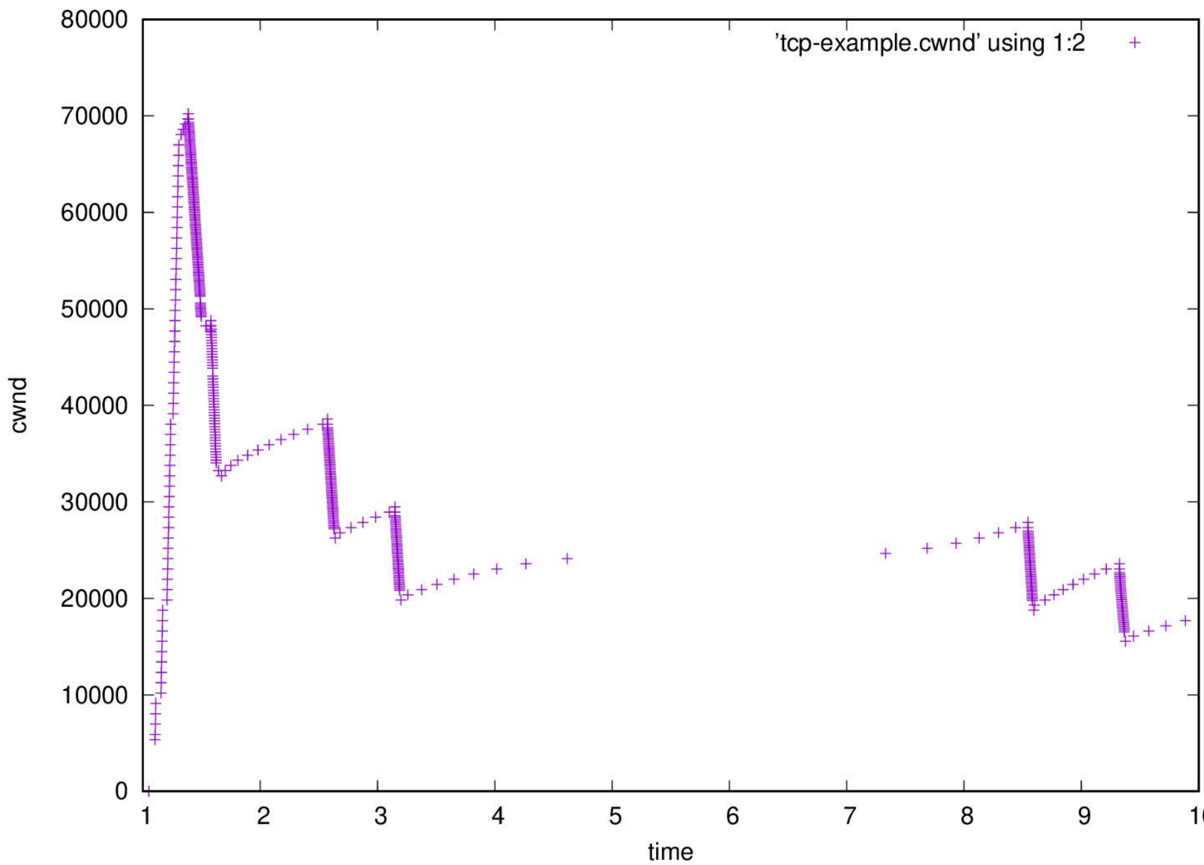
Wireshark · Conversations · tcp-example-2-0.pcap

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

Address A	Address 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	10.1.2.2	11,944	4,816k	7,774	4,585k	4,170	230k	0.000000	8.9743	4,087k	

Here, as we can see the Avg Throughput is 4087k bits/s.

B.



C.

Ans. I have noted the following insights by comparing cwnd q1 vs q4.

1. Transmission Rate: As we have used TcpCubic in place of TcpNewReno, the cubic function helps in increasing the transmission rate and increased cwnd size.
2. Slow Start and Fast Recovery:if we consider previous plots we see a increase in cwnd from q1 to q2 but if we compare q2 to this q4 then we can say the cwnd has the highest in this part due to cubic function being used.
3. Congestion Avoidance: The major difference is in the congestion avoidance window, the tcp cubic function offers enhanced congestion avoidance because of cubic function being used. In TcpNewReno protocol, the cwnd increased linearly while in TcpCubic increases by cubic function.