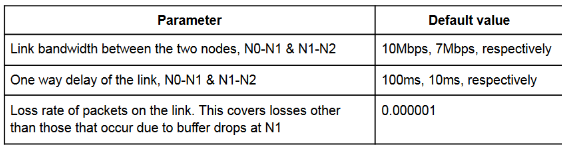
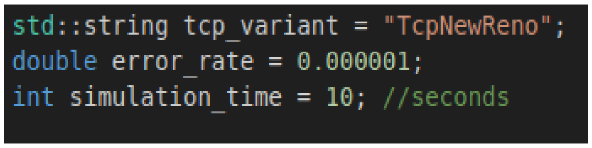
Computer Network Assignment 4

Keshav Chhabra(2022247)

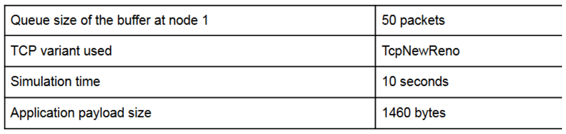
PART 1



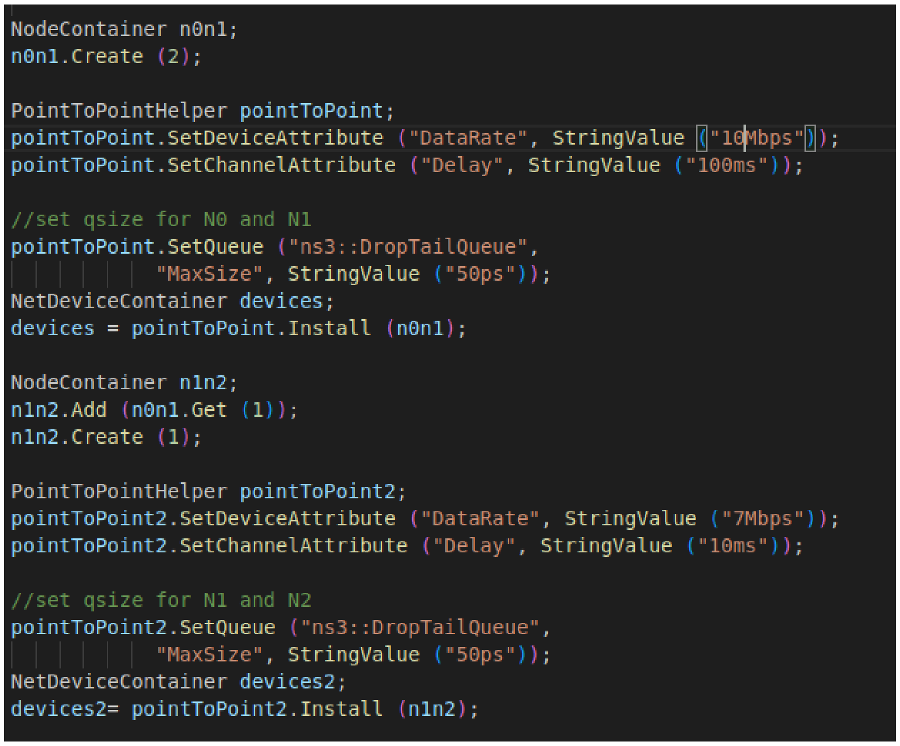
Parameter	Default value
Link bandwidth between the two nodes, N0-N1 & N1-N2	10Mbps, 7Mbps, respectively
One way delay of the link, N0-N1 & N1-N2	100ms, 10ms, respectively
Loss rate of packets on the link. This covers losses other than those that occur due to buffer drops at N1	0.000001



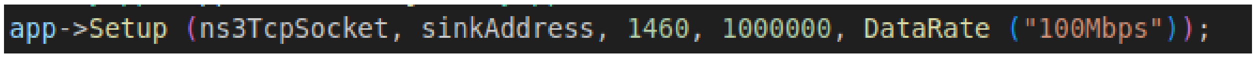
```
std::string tcp_variant = "TcpNewReno";
double error_rate = 0.000001;
int simulation_time = 10; //seconds
```



Queue size of the buffer at node 1	50 packets
TCP variant used	TcpNewReno
Simulation time	10 seconds
Application payload size	1460 bytes



```
NodeContainer n0n1;
n0n1.Create (2);

PointToPointHelper pointToPoint;
pointToPoint.SetDeviceAttribute ("DataRate", StringValue ("10Mbps"));
pointToPoint.SetChannelAttribute ("Delay", StringValue ("100ms"));

//set qsize for N0 and N1
pointToPoint.SetQueue ("ns3::DropTailQueue",
            "MaxSize", StringValue ("50ps"));
NetDeviceContainer devices;
devices = pointToPoint.Install (n0n1);

NodeContainer n1n2;
n1n2.Add (n0n1.Get (1));
n1n2.Create (1);

PointToPointHelper pointToPoint2;
pointToPoint2.SetDeviceAttribute ("DataRate", StringValue ("7Mbps"));
pointToPoint2.SetChannelAttribute ("Delay", StringValue ("10ms"));

//set qsize for N1 and N2
pointToPoint2.SetQueue ("ns3::DropTailQueue",
            "MaxSize", StringValue ("50ps"));
NetDeviceContainer devices2;
devices2= pointToPoint2.Install (n1n2);
```



```
app->Setup (ns3TcpSocket, sinkAddress, 1460, 1000000, DataRate ("100Mbps"));
```

The **error rate** specifies the probability that any given packet will be dropped (lost) due to factors other than buffer overflow as required by the assignment .

The Setup method configures the application (MyApp) with:

1. A TCP socket(ns3TcpSocket) to send packets.
2. The destination address(SinkAddress) of the receiver.
3. A packet size of(1460) bytes.
4. A plan to send up to 1,000,000 packets.
5. A sending rate of (100 Mbps).

Part 1

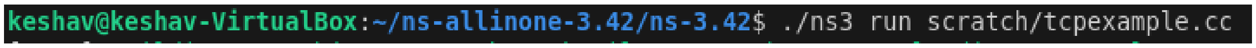
a) The expected(theoretical) maximum throughput cannot exceed the bandwidth of the bottleneck link, which is **7 Mbps** (N1-N2).

---

b) BDP = Bandwidth × Round-Trip Time (RTT)
RTT is the round-trip delay for packets traveling through N0 -> N1 -> N2 and back:
RTT = 2 x (N0-N1 delay + N1-N2 delay)
RTT = 2 x (100 ms + 10 ms) = 220 ms = 0.22 seconds.
BDP= 7× 106bps × 0.22seconds=1,540,000bits.
Packet size= 1460 bytes × 8=11,680bits.
BDP (in packets)= 1,540,000bits / 11,680 bits/packets ≈ 132 packets.

---

c) The average computed throughput of the TCP transfer



```
keshav@keshav-VirtualBox:~/ns-allinone-3.42/ns-3.42$ ./ns3 run scratch/tcpexample.cc
```

Parameter	Value
Address A	10.1.1.1
Port A	49153
Address B	10.1.2.2
Port B	8080
Packets	12,907
Bytes	5 MB
Stream ID	0
Packets A → B	8,438
Bytes A → B	5 MB
Bytes B → A	245 kB
Rel Start	0.000000
Duration	8.9998
Bits/s A → B	4,126 kbps
Bits/s B → A	217 kbps

$$\text{Throughput (in Mbps)} = \frac{\text{Total Bytes Transferred (in bits)}}{\text{Duration (in seconds)} \times 10^6}$$

Inspecting was done using **wireshark** opening the .pcap file at the receiver end .

Bytes A -> B: 3 MB = 3 x 10^6 bytes

Convert to bits: 3 x 10^6 x 8 = 24 x 10^6 bits

Bytes B -> A: 245 kB = 245 x 10^3 bytes

Convert to bits: 245 x 10^3 x 8 = 1.96 x 10^6 bits

Total Data Transferred: 24 x 10^6 + 1.96 x 10^6 = 25.96 x 10^6 bits

Duration = 8.9998 seconds

Throughput = (25.96 x 10^6) / (8.9998 x 10^6) Mbps ~ 3.08 Mbps

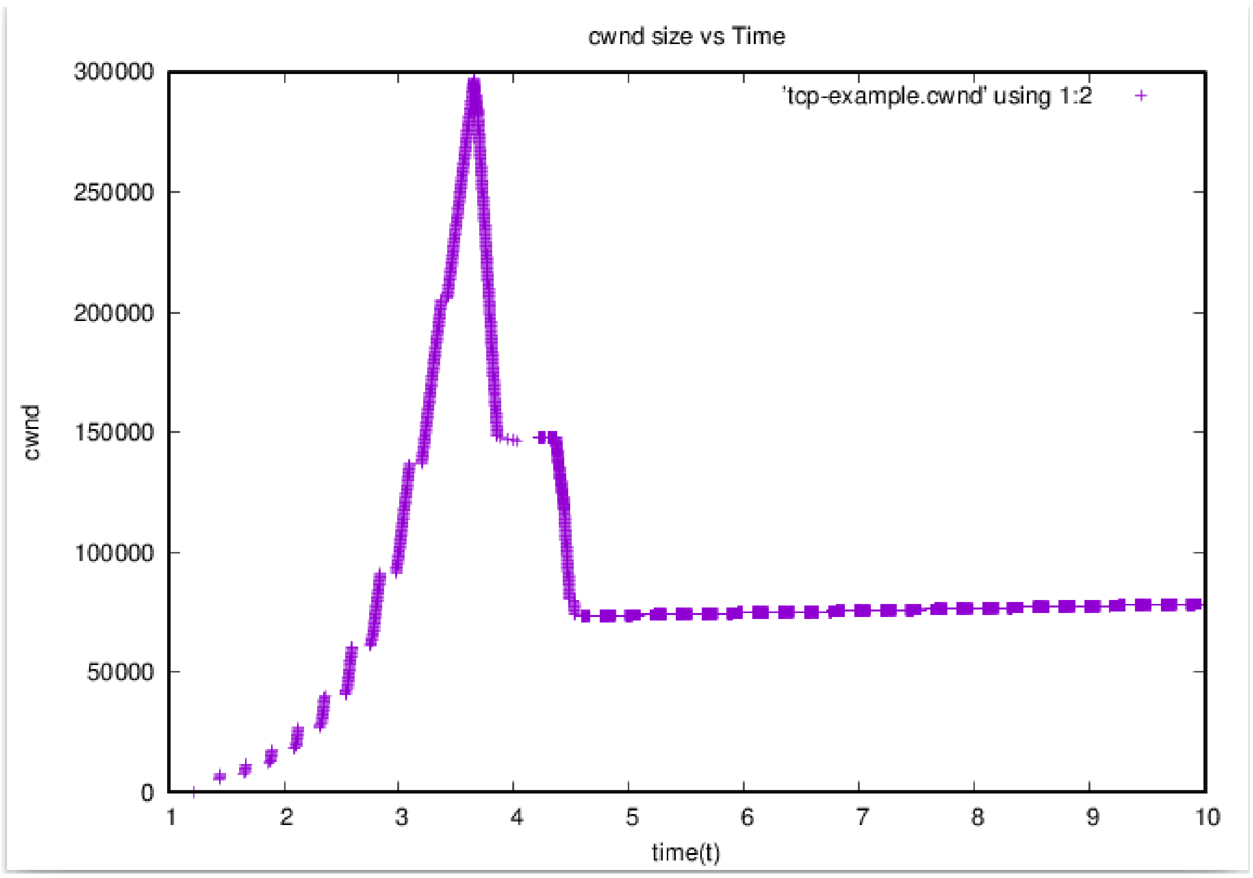
---

d) The calculated throughput (**3.08 Mbps**) is significantly lower than the maximum expected link bandwidth of **10 Mbps** or **7 Mbps**.

TCP Congestion Control: The protocol used in the simulation is **TcpReno**, which involves congestion control mechanisms. TCP limits its transmission rate to avoid network congestion, especially when packet loss occurs or delays are observed, causing a lower throughput compared to the theoretical maximum bandwidth.
Round-Trip Time (RTT): The delay characteristics of the links (100 ms for N0-N1 and 10 ms for N1-N2) contribute to the overall round-trip time. High RTT can lower the effective throughput of a TCP connection, especially when the window size is small or there are latency-sensitive applications.

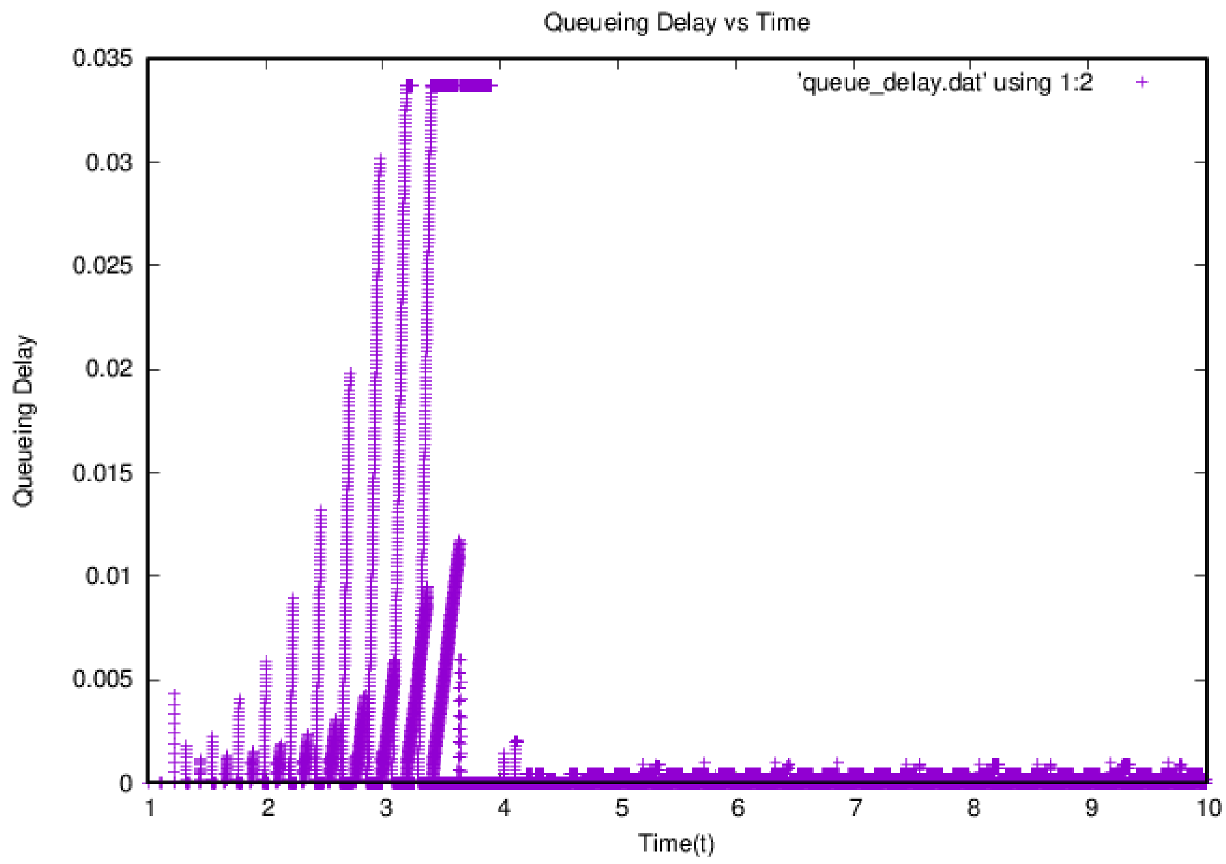
---

e)



---

f)



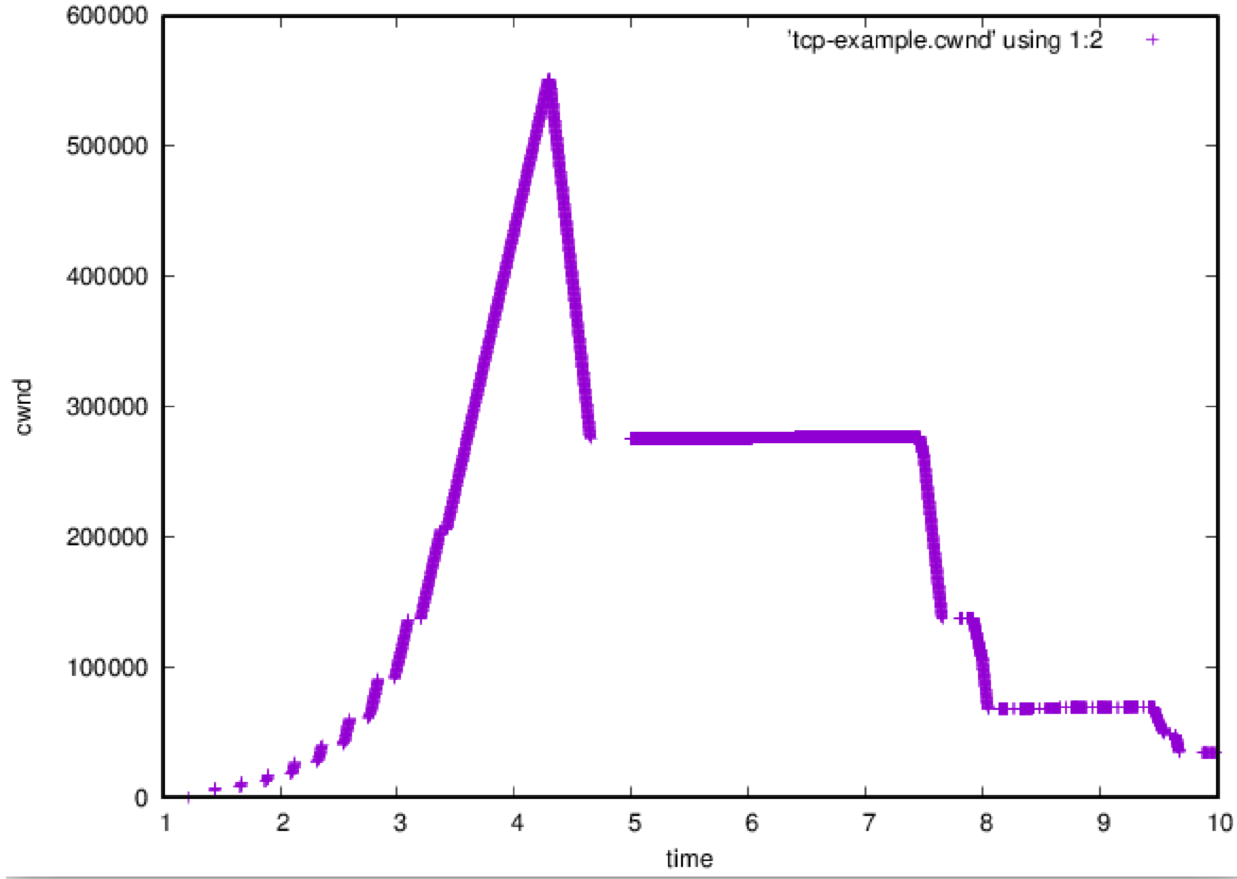
Yes, the plots in the images (queuing delay vs time and congestion window/cwnd size vs time) are directly related.

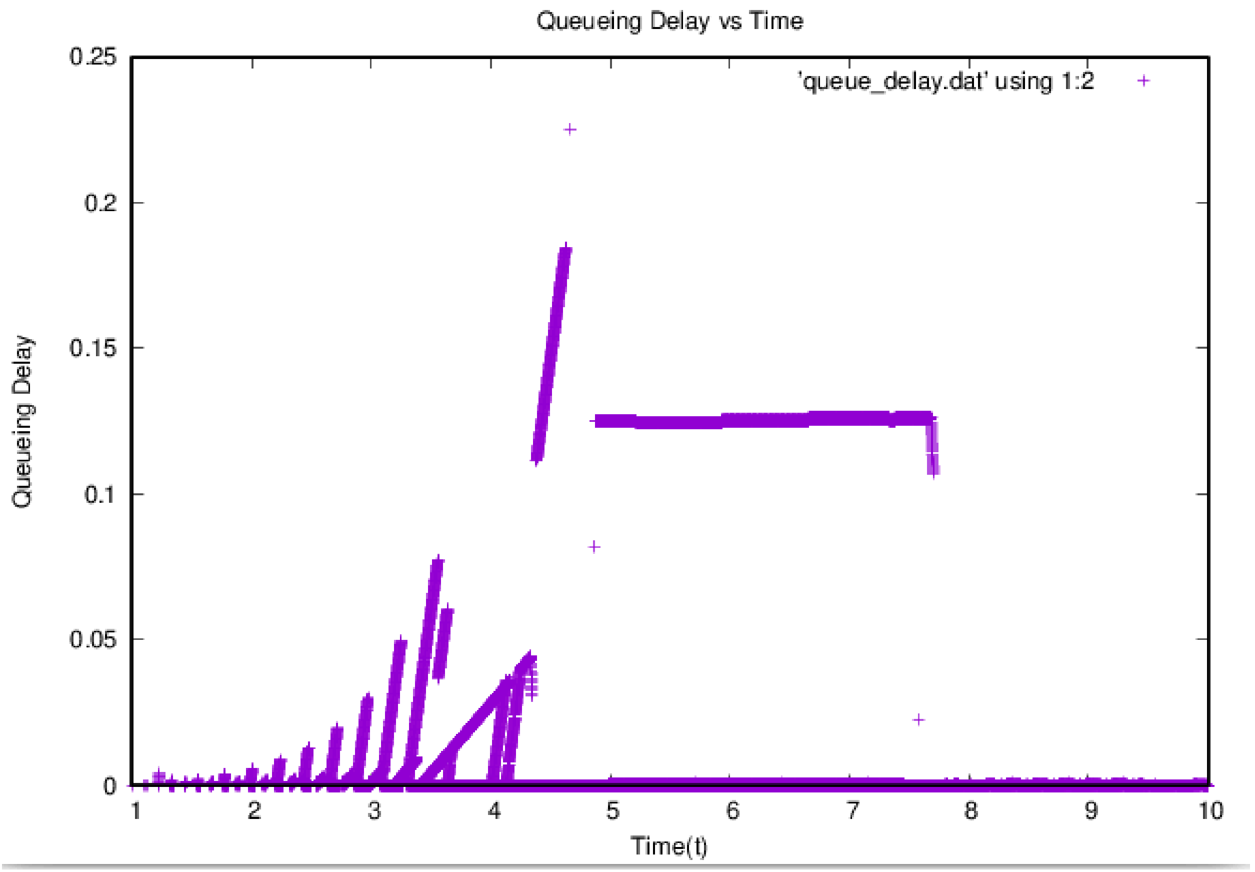
During periods of high congestion window (cwnd) size, we typically see corresponding spikes in queuing delay. This is evident around 3-4s , as the cwnd size ramps up, more bytes are sent and have to suffer queueing delays correspondingly .

When the congestion window is large, more packets are being sent into the network

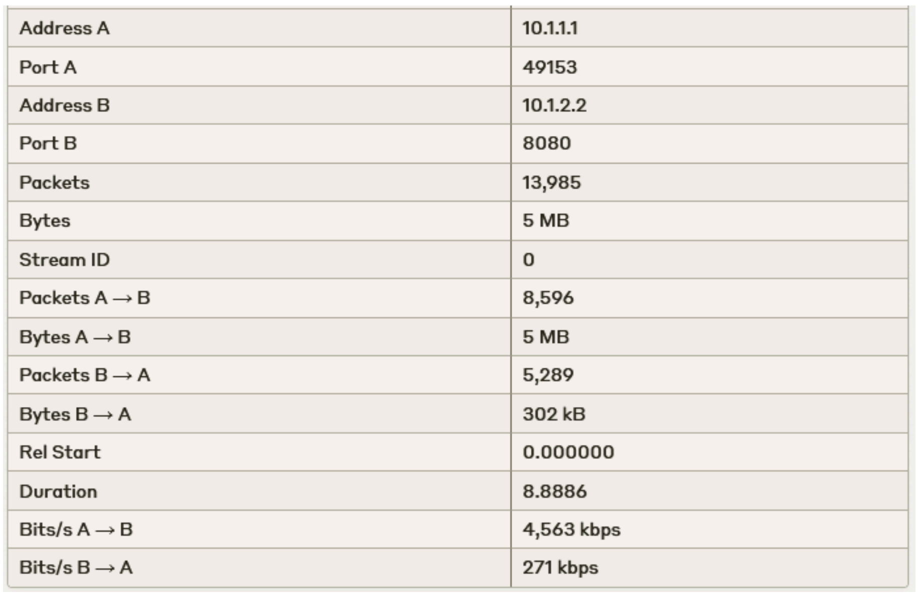
As the buffer fills, packets must wait longer in the queue, causing increased queuing delays

When congestion is detected (through packet loss or increased delays), TCP NewReno reduces the congestion window, which then helps reduce the queuing delay .


'tcp-example.cwnd' using 1:2
cwnd
time
0
100000
200000
300000
400000
500000
600000
1
2
3
4
5
6
7
8
9
10



a) **What is the average computed throughput of the TCP transfer?**



Address A	10.1.1.1
Port A	49153
Address B	10.1.2.2
Port B	8080
Packets	13,985
Bytes	5 MB
Stream ID	0
Packets A → B	8,596
Bytes A → B	5 MB
Packets B → A	5,289
Bytes B → A	302 kB
Rel Start	0.000000
Duration	8.8886
Bits/s A → B	4,563 kbps
Bits/s B → A	271 kbps

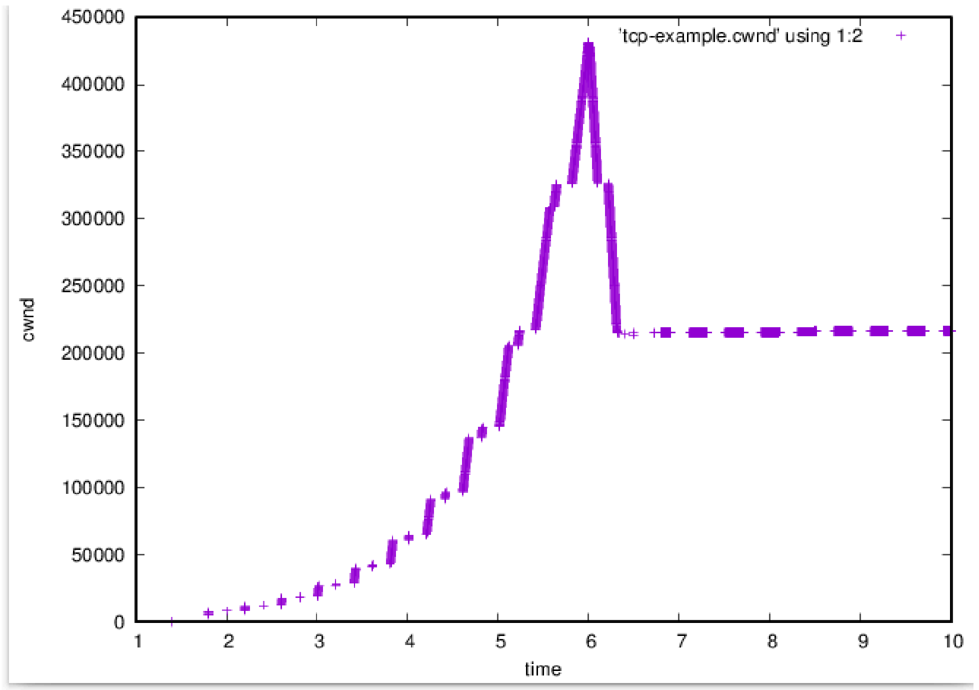
A -> B Throughput: Throughput A -> B = 41,943,040 / 8.8886 ≈ 4,719,997.63 bps ≈ 4,720 kbps

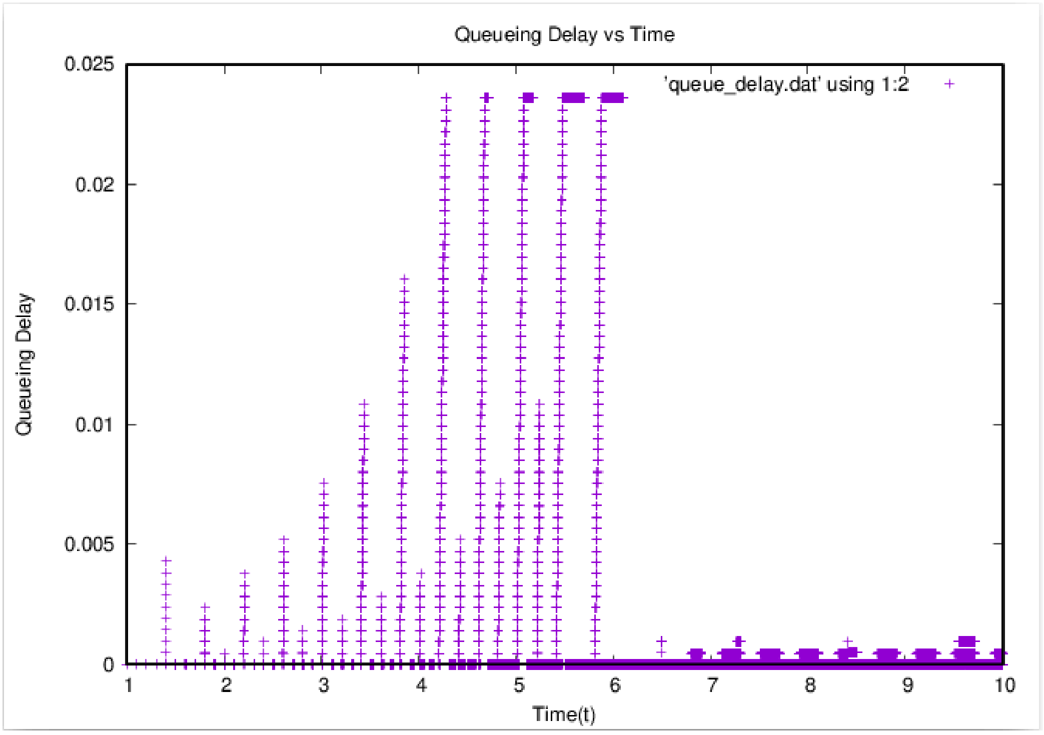
kbps B -> A Throughput: Throughput B -> A = 2,473,984 / 8.8886 ≈ 278,344.41 bps ≈ 278 kbps

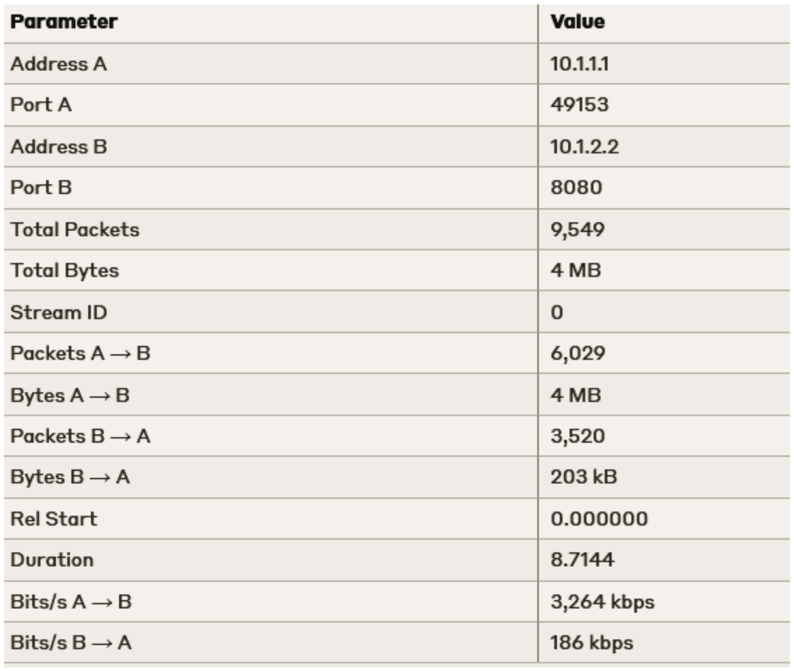
---

d) **Compare CWND plots of Q.1. and Q.2.; what insights did you gain?**

1000p queue: Peak reaches around 550,000
50p queue: Peak reaches around 300,000 The larger queue size allows for a higher maximum CWND value.
These differences illustrate the trade-off between throughput and latency in TCP congestion control, where larger buffers can increase throughput but at the cost of potentially higher latency and more dramatic oscillations.







Parameter	Value
Address A	10.1.1.1
Port A	49153
Address B	10.1.2.2
Port B	8080
Total Packets	9,549
Total Bytes	4 MB
Stream ID	0
Packets A → B	6,029
Bytes A → B	4 MB
Packets B → A	3,520
Bytes B → A	203 kB
Rel Start	0.000000
Duration	8.7144
Bits/s A → B	3,264 kbps
Bits/s B → A	186 kbps

a) **What is the average computed throughput of the TCP transfer?**

For Direction A → B:

Bytes A → B = 4 MB = 4 × 1024 × 1024 = 4,194,304 bytes

Duration = 8.7144 seconds

Throughput A → B = 4,194,304 bytes / 8.7144 seconds

Throughput A → B = 481,307.85 bytes/second

Converting to Mbps = (481,307.85 × 8) / (1024 × 1024) ≈ 3.67 Mbps

For Direction B → A:

Bytes B → A = 203 kB = 203 × 1024 = 207,872 bytes

Duration = 8.7144 seconds

Throughput B → A = 207,872 bytes / 8.7144 seconds

Throughput B → A = 23,853.95 bytes/second

Converting to Mbps = (23,853.95 × 8) / (1024 × 1024) ≈ 0.182 Mbps

Total Bidirectional Throughput:

Total bytes = 4,194,304 + 207,872 = 4,402,176 bytes

Total throughput = 4,402,176 / 8.7144 = 505,161.8 bytes/second

In Mbps = (505,161.8 × 8) / (1024 × 1024) ≈ 3.85 Mbps

This matches closely with the given bits/s values in the data:

A → B: 3,264 kbps ≈ 3.264 Mbps (calculated 3.67 Mbps)

B → A: 186 kbps ≈ 0.186 Mbps (calculated 0.182 Mbps)

d) Compare queuing delay plots of Q.1. and Q.3.; what insights did you gain?

(10Mbps, 100ms): Maximum delay around 0.023 seconds

(7Mbps, 10ms): Maximum delay around 0.033 seconds The lower bandwidth in the second scenario (7Mbps) leads to higher peak queuing delays.

The reduced bandwidth (7Mbps) in the second scenario creates more queuing as packets arrive faster than they can be transmitted

The shorter delay (10ms vs 100ms) in the second scenario leads to quicker feedback but more intense queuing buildup

---