

Q.1 Answer

a. What is the maximum expected value (theoretical) of throughput (in Mbps)? Why?

The maximum expected throughput is typically constrained by the bandwidth of the bottleneck link in the network, which in this case is the link between N1 and N2 (with a bandwidth of 7 Mbps). While TCP performance depends on various factors such as congestion control, network delay, and packet loss, the theoretical maximum throughput can be considered as the bandwidth of the bottleneck link under ideal conditions (assuming no losses, minimal congestion, and a well-configured TCP variant like `TcpCubic`).

Thus, the maximum expected theoretical throughput is **7 Mbps**, as it is determined by the slowest link (N1-N2).

b. How much is Bandwidth-Delay-Product (BDP)? Express your answer in terms of the number of packets.

The **Bandwidth-Delay Product (BDP)** is the amount of data that can be in transit in the network at any given time, which is calculated as:

$$\text{BDP} = \text{Bandwidth} \times \text{Round-Trip Time (RTT)}$$

To compute the BDP for the link between N0 and N1, we will use the following:

- **Bandwidth (N0-N1)** = 10 Mbps = 10,000,000 bits per second
- **Delay (N0-N1)** = 100 ms = 0.1 seconds

The BDP (in bits) is:

$$\text{BDP} = 10,000,000 \text{ bits/sec} \times 0.1 \text{ sec} = 1,000,000 \text{ bits}$$

Now, to express the BDP in terms of the number of packets, we need to divide the BDP by the size of the TCP packets. Assuming the application payload is **1460 bytes** (which is typical for TCP), and including the TCP and IP headers (which are usually about 40 bytes), the total packet size is:

$$\text{Packet size} = 1460 \text{ bytes} + 40 \text{ bytes} = 1500 \text{ bytes} = 12000 \text{ bits}$$

Now, the number of packets that can fit into the BDP is:

$$\text{Number of packets} = \frac{\text{BDP (in bits)}}{\text{Packet size (in bits)}} = \frac{1,000,000 \text{ bits}}{12000 \text{ bits}} \approx 83.33 \text{ packets}$$

Therefore, the **Bandwidth-Delay Product (BDP)** is approximately **83 packets**.

d. Compare CWND Plots of Q.1 and Q.2; Insights

Analysis of CWND Evolution:

In **Q.1** (with the default queue size), the CWND will likely exhibit the typical slow start behavior, gradually increasing until it reaches the bandwidth-delay product or a congestion threshold (as defined by the network conditions and the TCP variant in use).

In **Q.2** (with the queue size increased to 1000), you may see a different CWND growth pattern due to the increased buffer size. The larger queue size allows for more data to be buffered in the network, which may lead to the TCP sender having more data to send without waiting for acknowledgments, potentially allowing the CWND to grow faster initially. However, this might also cause longer delay periods when the buffer fills up, leading to congestion and slower growth afterward.

Insights to Gain from Comparing the Plots:

Impact of Larger Buffer Size: A larger buffer size may lead to an increased CWND more quickly, especially during the slow start phase, as packets can be buffered while waiting for acknowledgments. However, the system might become more prone to congestion, which can result in a decrease in the rate of CWND growth as the buffer fills up.

Queueing Delay: When the queue size is increased, you may notice higher queueing delays as the network may experience bursts of traffic due to the larger buffer size. This could be observed as spikes in the queueing delay plot, which corresponds to packets waiting longer in the queue before being dequeued.

Efficiency and Buffer Overflow: The larger buffer size may lead to better utilization of network resources in terms of throughput, but it can also lead to inefficiency and higher delays when congestion occurs, especially if the buffer overflows and drops packets.

Practical Conclusion:

A larger buffer size may improve throughput in some cases, but it could also increase delays due to buffer overflow or excessive queuing. It's essential to balance buffer size with network traffic and congestion control parameters to optimize performance.

Change N1-N2 Bandwidth to 10 Mbps and N1-N2 Delay to 100ms

a. What is the average computed throughput of the TCP transfer?

With the change in bandwidth to 10 Mbps and the delay to 100ms, the throughput of the TCP transfer will depend on factors like congestion, buffer sizes, and network congestion control mechanisms. However, in an ideal scenario without significant packet loss, we can estimate the throughput as:

$$\text{Throughput} = \frac{\text{Window Size}}{\text{Round Trip Time (RTT)}}$$

Given that the bandwidth is 10 Mbps (megabits per second) and the round-trip delay is 100ms, the throughput will be limited by the bandwidth and congestion window size. The throughput may be closer to the available bandwidth if the network conditions allow the CWND to grow large enough to take advantage of the full bandwidth.

b. Plot CWND with Time

The CWND (Congestion Window Size) will show typical TCP congestion control behavior. Initially, during the slow start phase, the CWND will increase exponentially, but as congestion is detected, it will slow down. The larger bandwidth (10 Mbps) may allow a larger CWND, while the 100ms delay will influence the growth rate due to the increased round-trip time.

```
# Example to plot CWND with time using Python
import matplotlib.pyplot as plt

# Sample CWND and Time data (replace with actual simulation data)
time = [0, 1, 2, 3, 4, 5, 6, 7, 8, 9] # Time in seconds
cwnd = [1, 2, 4, 8, 16, 32, 64, 128, 256, 512] # CWND in segments

plt.plot(time, cwnd)
plt.xlabel('Time (s)')
plt.ylabel('CWND (segments)')
plt.title('CWND vs Time')
plt.grid(True)
plt.show()
```

This plot shows how the CWND grows over time as the TCP sender adjusts its window size in response to network conditions.

c. Plot Queueing Delay with Time

The queueing delay is the time a packet spends in the queue before being transmitted. With a delay of 100ms and bandwidth of 10 Mbps, the queueing delay will depend on the packet arrival rate and the queue size. A higher bandwidth and longer delay typically cause larger queues, leading to higher queueing delays.

```
# Example to plot queueing delay with time using Python
queueing_delay = [0, 0.02, 0.04, 0.06, 0.08, 0.1, 0.12, 0.15, 0.18, 0.2] # Delay in seconds

plt.plot(time, queueing_delay)
plt.xlabel('Time (s)')
plt.ylabel('Queueing Delay (s)')
plt.title('Queueing Delay vs Time')
plt.grid(True)
plt.show()
```

This plot shows how the queueing delay changes over time. As more packets arrive, the queue may grow, causing an increase in delay.

d. Compare Queueing Delay Plots of Q.1. and Q.3.; What Insights Did You Gain?

In **Q.1** (with default parameters), the queueing delay may have been lower compared to **Q.3** (with bandwidth increased to 10 Mbps and delay increased to 100ms). The larger bandwidth allows the sender to transmit data more quickly, but the increased round-trip delay can cause longer periods of packet waiting in the queue.

Insights:

- **Impact of Larger Bandwidth and Delay:** Increasing the bandwidth (to 10 Mbps) allows for more data to be transmitted, but the increased round-trip delay (100ms) means the sender has to wait longer for acknowledgments. This can cause packets to accumulate in the queue, leading to higher queueing delays.
- **Effect of Buffer Size:** Larger queues might result in more packet buffering, which increases queueing delay, especially when the transmission rate exceeds the rate at which the receiver can process packets.
- **Network Congestion:** In both Q.1 and Q.3, network congestion can cause the queue size to increase, but with the larger bandwidth in Q.3, we may observe bursts of data being sent quickly, filling the queue and causing higher delays.

In conclusion, increasing both bandwidth and delay can lead to higher throughput but may also cause increased queueing delays, especially when the network is heavily utilized. Balancing bandwidth, delay, and buffer size is essential for efficient network performance.