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**.