CS 4220 Computer Networks
Assignment 2

1 What is the goal of the transport layer?

Solution: The goal is to provide efficient, reliable, and cost-effective service to users and processes in the application layer.

2 Explain the difference between multiplexing and demultiplexing at the transport layer.

Solution: Multiplexing combines data from many applications for transmission, while demultiplexing separates them upon arrival.
Multiplexing: At the sender side, the transport layer collects data from multiple application processes, adds appropriate header information (like source and destination port numbers), and sends them down to the network layer as separate segments.
Demultiplexing: At the receiver side, the transport layer examines the header fields (IP addresses and port numbers) of incoming segments and delivers each one to the correct application socket.

3 Explain the difference between UDP and TCP. Give one example for each protocol.

Solution: UDP is connectionless. It does not require connection establishment or release, and it does not have flow or congestion control. Example: DNS. TCP requires connection establishment and release. It has flow control, congestion control, and error control. Example: HTTP.

4 In our rdt protocols, why did we need to introduce sequence numbers?

Solution: Sequence numbers are needed to distinguish new packets from retransmissions. If a packet or its ACK is lost, the sender may resend the same packet. Without sequence numbers, the receiver cannot tell whether it is receiving new data or a duplicate, which could lead to delivering the same data more than once.

5 In our rdt protocols, why did we need to introduce timers?

Solution: Timers are needed to detect packet or ACK loss. If a sender does not receive an acknowledgment within a certain time, the timer expires and triggers retransmission. Without a timer, the sender would not know whether the packet or its ACK was lost, and reliable delivery could not be guaranteed.

6 Consider a system using Go-Back-N where the sender window size is 4. If the sender transmits frames 0, 1, 2, and 3, and frame 1 is lost, describe what happens next at both the sender and receiver sides.

Solution: The sender transmits frames 0, 1, 2, and 3. Frame 1 is lost, so the receiver discards frames 2 and 3 and keeps sending ACK 0 (the last acknowledged frame). The sender, upon receiving no acknowledgment for frame 1, retransmits frames 1, 2, and 3. The receiver receives these retransmitted frames, acknowledges them, and the transmission continues with the next set of frames.

7 In a system using Selective Repeat with a window size of 4, if frame 2 is lost but frames 3, 4, and 5 are received correctly, what steps does the sender take to ensure all frames are delivered in the correct order?

Solution: The receiver acknowledges frames 3, 4, and 5 but waits for frame 2. The sender identifies the missing acknowledgment for frame 2 and retransmits it. The receiver accepts the retransmitted frame 2 and delivers frames 2, 3, 4, and 5 in the correct order. The sender continues transmission.

8 Imagine that a two-way handshake rather than a three-way handshake was used to set up connections. What would be the problem?

Solution: With a two-way handshake, a delayed or duplicated connection request could cause one side to establish a connection even though the other side did not intend to or has already closed it. This creates the half-open problem, where one host believes the connection is established while the other does not, leading to an inconsistent state and wasted resources.

9 Explain how TCP establishes the connection.

Solution: The steps of a three-way handshake. The goal is for both sides to agree on initial sequence numbers and confirm that each is ready to transmit data.
The steps are:

- SYN: The client sends a TCP segment with the SYN (synchronize) flag set and an initial sequence number x. This indicates the client wants to start a connection.
- SYN-ACK: The server responds with a segment that has both SYN and ACK flags set. It acknowledges the client's sequence number (ACK = x + 1) and sends its own initial sequence number y.
- ACK: The client replies with an ACK segment acknowledging the server's sequence number (ACK = y + 1).

10 Why does UDP exist? Would it not have been enough just to let the client send raw IP packets?

Solution: No. IP packets contain IP addresses, which specify a destination machine. Once such a packet arrives, how would the network handler know which process or application to give it? UDP packets contain a destination port. This information is essential so they can be delivered using the correct process.