
Question 1: How is a UDP socket fully identified?

- A A UDP socket is fully identified by the destination IP address.
- B A UDP socket is fully identified by the source IP address and IP destination address.
- C A UDP socket is fully identified by the source IP address, the source port, the destination IP address and the destination port.
- D A UDP socket is fully identified by the destination IP address and the destination port.

Question 2: How is a TCP connection fully identified?

- A A TCP connection and the corresponding sockets are fully identified by the destination IP address.
- B A TCP connection and the corresponding sockets are fully identified by the source IP address, the source port, the destination IP address and the destination port.
- C A TCP connection and the corresponding sockets are fully identified by the source IP address and the destination IP address.
- D A TCP connection and the corresponding sockets are fully identified by the destination IP address and the destination port.

Question 3: Is it possible for an application to support reliable data transfer even when running over UDP?

- A Yes, if reliable data transfer mechanisms are implemented in the **application** layer.
- B No, UDP does not support reliable data transfer.
- C Yes, UDP's checksum field allows this.
- D Yes, if reliable data transfer mechanisms are implemented in the **physical** layer.

Question 4: Suppose a process in Host X has a UDP socket with port number 6776. Imagine both Host Y and Host Z each send a UDP segment to Host X with destination port 6776. Will both these segments be directed to the same socket at Host X?

- A No, Host X will have a separate "connection socket" for Host Y and for Host Z.
- B None of the segments will be received because a server (welcoming) socket is needed first.
- C Yes, both segments will be directed to the same socket.
- D It is impossible to know because Host X may have several UDP sockets with port number 6776.

Question 5: Suppose a Web server is running in Host X on port 80. Imagine this Web server uses persistent connections and is currently receiving HTTP requests from both Host Y and Host Z. Are all of the requests from both hosts being received through the same socket at Host X?

- A No, Host X will have a separate "connection socket" for each persistent connection.
- B Yes, both persistent connections connect and remain connected to the same "server socket".
- C Yes, Host X will have a single socket for both connections, which are distinguished by their source IP address and port.
- D No, because for each HTTP request, a new connection must be established.

Question 6: Considering typical reliable data transfer protocols, such as the ones in the book, why are sequence numbers needed?

- A Sequence numbers are required for **a sender** to find out whether an arriving packet contains new data or is a retransmission.
- B Sequence numbers are only required if the network layer generates duplicate packets.
- C Sequence numbers are required for **a receiver** to find out whether an arriving packet contains new data or is a retransmission.
- D Sequence numbers are required for **a sender** to find out whether data has been lost and if a retransmission is needed.

Question 7: Considering typical reliable data transfer protocols, such as the ones in the book, why are timers needed?

- A To handle delays in the channel. If a packet or its ACK is not received within the duration of the timer for the packet, then the timer is reset with a larger duration.
- B To handle duplicate packets in the channel. If a duplicate ACK for a transmitted packet is received within the duration of the timer for that packet, then the ACK is discarded.
- C To handle congestion in the channel. If a packet or its ACK is not received within the duration of the timer for the packet, then the timer is reset with a larger duration.
- D To handle losses in the channel. If a packet or its ACK is not received within the duration of the timer for the packet, then it is retransmitted.

Question 8: Consider a stop-and-wait data-transfer protocol that provides retransmissions but uses **only** negative acknowledgements (NAK). Would such protocol work in a channel without bit errors but with losses?

- A No, because NAKs are only useful for corrupted packets and there are no bit errors in the channel.
- B No, because not receiving a NAK could mean that it was lost and not that the data packet was correctly received.
- C Yes, because if the channel has no bit errors a NAK would always be received.
- D Yes, because the sender could implement a timer to retransmit any lost packets even if a NAK is not received.

Question 9: What is meant by transport-layer demultiplexing?

- A Taking data from one socket (one of possibly many sockets), encapsulating a data chunk with header information in a **transport** layer segment, and eventually passing this segment to the **network** layer.
- B Taking data from multiple sockets, all associated with the same destination IP address, adding destination port numbers to each piece of data, and then concatenating these to form a **transport** layer segment, and eventually passing this segment to the **network** layer.
- C Receiving a transport-layer segment from the **network** layer, extracting the payload, and delivering the data to the correct socket in the application layer.
- D Receiving a transport-layer segment from the **application** layer, extracting the payload, determining the destination IP address for the data, and then passing the segment and the IP address back down to the network layer.

Question 10: Why is the UDP header length field needed?

- A To make the header an even number of bytes.
- B The UDP length field gives the length of the UDP header + data, helping the receiver know how many bytes belong to this datagram.
- C Because this field is needed to determine if we are using 32 or 128 bit IP addresses.
- D To determine if there is any corruption in the segment.