Part 1: Theory (35 points)

Question 1 [20 points]: Below, there are some statements concerning UDP and TCP. Comment on their validity. Provide a precise, clear and brief justification of your answers **[2 points each]**

1) The **checksum** on the header of a UDP packet ensures the reliable transportation of the packet to its destination. **False** (checksum is error detection mechanism, checks for altered bits)

2) Consider the **congestion control of TCP**. When the timer of the sender expires, the value of **ssthresh** is set to the half of its previous value. **False** (ssthresh = cwnd / 2)

3) During the slow start phase of TCP, the sending rate is increasing exponentially every RTT. **True**

4) A **triple duplicate ACK** event indicates a possibly high congestion in the network. **False** (3 duplicate ACKs indicate a mild congestion in the network. Some packets still arrive)

5) When the **cwnd** value reaches the ssthresh, TCP jumps from the slow start phase to the congestion avoidance phase. **True**

6) During a TCP session, it is possible that a host sends segments to multiple receivers. **False** (A TCP session is a point-to-point connection.)

7) Consider a TCP session between a Host A and a Host B. If the receive buffer of A is full, A can not send any data to B due to the **TCP's flow control**. **False** (B can not send to A. A can still send as the buffer of B is not full)

8) TCP's flow control would be useless, if the reading rate of the application at the receiver is always faster than the sender's sending rate. **True**

9) Can you think of systems/applications that use both UDP and TCP protocols in their operation? **(DNS, Skype, SNMP, …)**

10) When an app uses UDP as a transport layer protocol, it is impossible to provide guarantees for the data delivery. (Be careful here!) **False** (UDP does not provide reliable data transportation. However, an application can still provide reliability by implementing this functionality on the application layer.)

Question 2 [15 points]: Provide brief and clear answers to the following questions:

1) Discuss two important differences between the flow control vs. congestion control of TCP. **[5 points]**

- Flow control controls traffic from sender to receiver (local phenomenon) / Congestion control, controls the whole network's traffic (global phenomenon).
- In flow control, receiver's buffer is prevented from being overwhelmed / In congestion control, the whole network is prevented from being overwhelmed.
- In flow control, the sender limits his sending rate based on precise information provided by the receiver (rwnd) / In congestion control, the sender limits his sending rate based on assumptions from various events (timeouts, 3 dup ACKs).

2) Describe why a programmer would choose to run an application over UDP instead of TCP. **[5 points]**

- In situations where you really want to get a simple answer quickly. (DNS)
- When you are delivering data that can be lost because newer data coming in will replace that previous data and there is no great impact from small losses. (Video streaming, gaming data).
- For multicast traffic. UDP can multicast to multiple hosts whereas TCP cannot do this at all.
- Real-time application where low-latency is necessary.

3) Do you consider that the fast retransmit(on 3 duplicate ACKs) acts "complementary" to the timer timeout on TCP? **[5 points]**

- TCP can work without fast retransmit (only with timeouts).
- Fast retransmit acts faster than timeouts, thus improving TCP's performance. (By retransmitting possibly lost segments before the timer expires).

Part 2: Problems on TCP (55 points)

Question 1 [25 points]: Assume two hosts A and B that communicate over a TCP connection. Host B has received all the bytes from Host A until the 227 byte. Afterwards, Host A sends two segments to Host B one after another. The first segment contains a payload of 30 bytes, while the second one contains 60 bytes. The source port from Host's A side is 306 and the destination port is 80. Host B sends an acknowledgment each it receives a TCP segment.

1) Draw the TCP flow diagram between Host A and Host B, assuming that the two segments arrive in the correct order, without any packet lost. You can ignore Host's B sequence number as no relevant information is given. What is the sequence number of the second segment? Explain. **[3 points]**

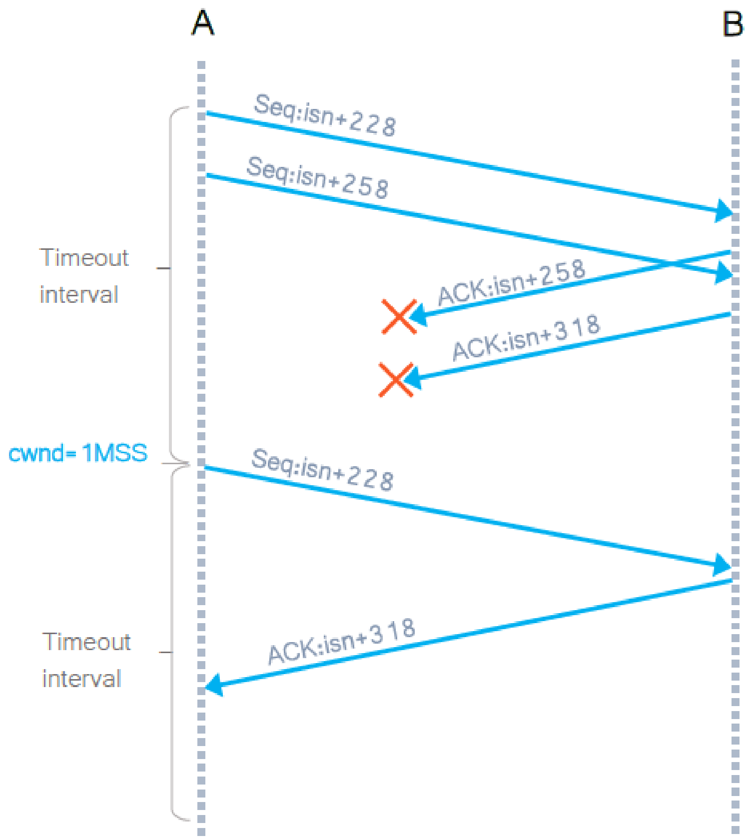
2) What are the source and destination port numbers of the acknowledge segments sent from Host B to Host A? **[2 points]**

Answer: Source port number is 80 / destination port number is 306

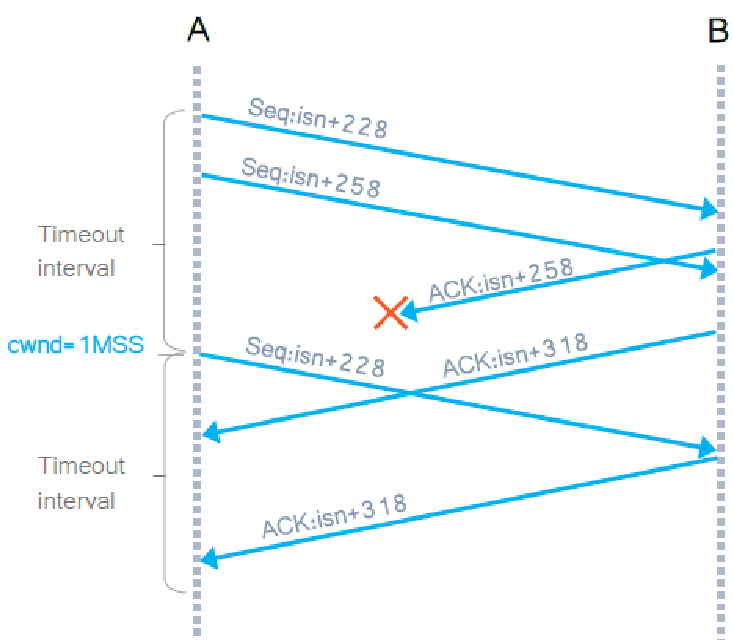
3) Assume that the second segment reaches Host B before the first segment. What is the acknowledgement number returned by Host B as a response to the arrival of the second segment? Explain. **[5 points]**

Answer: Host B is expecting the 228th byte but he receives the 258th . So, his ACK would be 228.

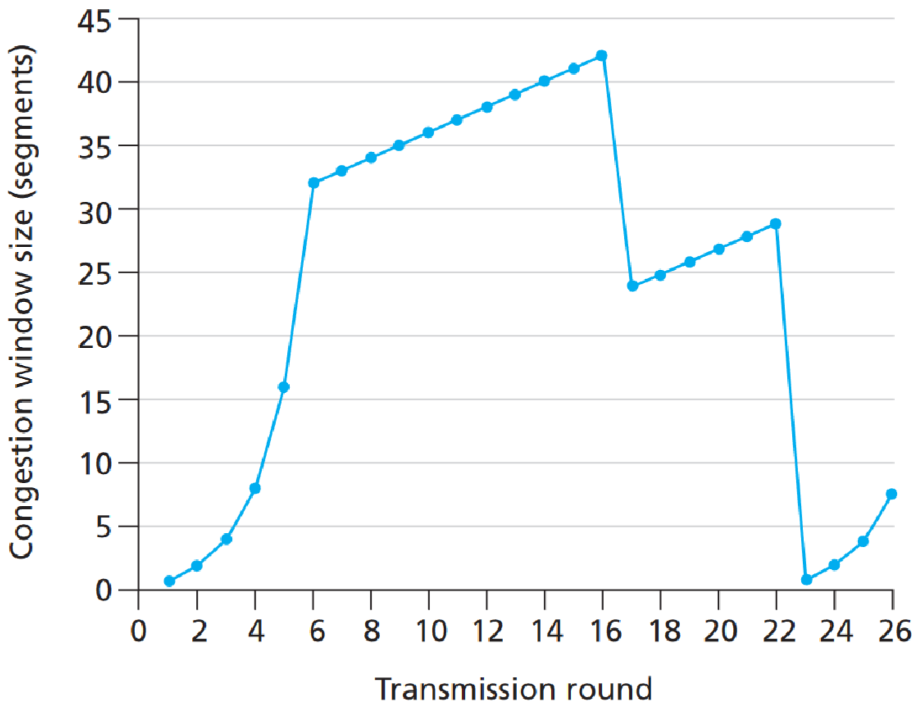
4) Assume that the two segments arrive at Host B in the correct order but both of the responses of Host B are lost. Draw the TCP flow diagram between Host A and Host B, and include the retransmission that will happen in the diagram. Which TCP mechanism will trigger the retransmission? Can Host A be certain about what actually happened? **[7.5 points]**



5) Assume that the two segments arrive in order at Host B. However, the first acknowledgment is lost, while the second one arrives at Host A, after the timer of the first packet has expired. Draw the TCP flow diagram between Host A and Host B. Does Host A retransmit a packet? Is a retransmission actually necessary? Explain. **[7.5 points]**



Question 2 [30 points]: The diagram below demonstrates an example of the behavior of a specific TCP flow over time. Answer the following questions, with sufficient justification.



1) Identify the time intervals when TCP slow start operates. How can we understand that? **[2 points]**

   Answer: [1-6] , [23-26]

2) Identify the time intervals when TCP congestion avoidance operates. How can we understand that? **[3 points]**

   Answer: [6-16] , [17-22]

3) After the 16th transmission round, a sudden drop to the size of congestion window is observed. What could have caused that? Speculate about the traffic in the network at this time period? **[5 points]**

   Answer: 3 duplicate ACKs / possibly mild network traffic

4) After the 22nd transmission round, a more severe drop of the size of the congestion window is observed. What could have caused that? Speculate about the traffic on the Internet at this time period? **[5 points]**

   Answer: Timeout / possibly heavy network traffic

5) At which transmission round the 89th segment is sent? How did you estimate it? **[5 points]**

Answer:
[Transmission rounds 1-6] (slow start) -> $2^0+2^1+2^2+2^3+2^4+2^5$ = 63 total segments (1-63 segments)
[Transmission rounds 7] (congestion avoidance) -> 2^5+1 = 33 total segments (64-97 segments)

6) What is the value of cwnd during the 17th transmission round? Justify your answer. **[5 points]**

$cwnd_1 = 2^0 = 1$
$cwnd_2 = 2^1 = cwnd_1 * 2 = 2$
$cwnd_3 = 2^2 = cwnd_2 * 2 = 4$
$cwnd_4 = 2^3 = cwnd_3 * 2 = 8$
$cwnd_5 = 2^4 = cwnd_4 * 2 = 16$
$cwnd_6 = 2^5 = cwnd_5 * 2 = 32$
$cwnd_7 = cwnd_6 + 1 = 33$
$cwnd_8 = cwnd_7 + 1 = 34$
….
$cwnd_{16} = cwnd_{15} + 1 = 42$

Answer: $cwnd_{17} = cwnd_{16} / 2 = 42 / 2 = 21$ (16th round -> 3 duplicate ACKs)

7) What is the value of ssthresh during the 24th transmission round? Explain. **[5 points]**

$cwnd_{17} = 21$
$cwnd_{18} = cwnd_{17} + 1 = 22$
….
$cwnd_{22} = cwnd_{21} + 1 = 26$

Answer: $ssthresh_{23} = cwnd_{22} / 2 = 26 / 2 = 13$ (22th round -> timeout)