Question 1 [6 Marks]

a. Explain how does the connectionless and connection-oriented transport layer multiplexing and demultiplexing work? **[2]**

In connectionless multiplexing the source and destination port numbers in attached with the message at the sender side of transport layer and in demultiplexing they are detach at receiver side.
In connection oriented we need source and destination port numbers and IP addresses for multiplexing and demultiplexing.

b. What is the advantage of using a pipeline protocol over the stop-and-wait protocol? **[1]**
It increases the utilization of bandwidth from sender to receiver, because we can send multiple packets without waiting for the ACK of all previously send packets.

c. Differentiate between flow control and congestion control. **[1]**
Flow control is performed to avoid overwhelming (sending data greater than receiver window) the receiver).
congestion control is performed to avoid packet drops in between the devices from sender to receiver.

d. Why is cumulative ack not suitable in Selective Repeat (SR) protocol? **[1]**
If we send cumulative ack from receiver to sender in SR, then dropped packets will not send again from sender to receiver.

e. When does Selective Repeat (SR) work like Go-Back-N (GBN)? **[1]**
If window size and sequence number have very less gap then SR will work like GBN.

Question 2 [2+8=10 Marks]

a. Suppose Bob joins a BitTorrent torrent, but he does not want to upload any data to any other peers (so called free-riding). Bob claims that he can receive a complete copy of the file that is shared by the swarm. Is Bob's claim possible? Why or why not?
Yes. His first claim is possible, as long as there are enough peers staying in the swarm for a long enough time. Bob can always receive data through optimistic unchoking by other peers

b. Consider distributing a file of F = 15 Gbits to N peers. The server has an upload rate of u_s = 30 Mbps, and each peer has a download rate of d_i = 2 Mbps and an upload rate of u. We have N peers in total.

$$D_{P2P} = max\{F/u_s,\ F/d_{min},\ NF/(u_s + \sum_{i=1}^{N} u_i)\}$$

Where, $F = 15 \text{ Gbits} = 15 * 1024 \text{ Mbits}$

$u_s = 30 \text{ Mbps}$

$d_{min} = d_i = 2 \text{ Mbps}$

i. Assume u = 700 Kbps. Do the math and compare the total amount of time needed to distribute the file for N = 10 and N = 100. How does increasing N affect this comparison?

Ui = 700 kbps => 0.7 Mbits

Peers	D_{p2p} in Mbits
N = 10	Max $\{\frac{15360}{30}, \frac{15360}{2}, \frac{10*15360}{30+(10*0.7)}\}$ = 7680
N = 100	Max $\{\frac{15360}{30}, \frac{15360}{2}, \frac{100*15360}{30+(100*0.7)}\}$ = 15360

Increasing number of peers will result in increasing overall file distribution time among all peers.

ii. Assume N = 50. Compare this time for u = 300 Kbps and u = 700 Kbps. How does increasing peer's upload rate (u) affect this comparison?

Peers = 50

Peers upload rate	D_{p2p} in Mbits
u = 300kbps => 0.3 Mbits	Max $\{\frac{15360}{30}, \frac{15360}{2}, \frac{50*15360}{30+(50*0.3)}\}$ = 17066.67
u = 700kbps => 0.7 Mbits	Max $\{\frac{15360}{30}, \frac{15360}{2}, \frac{50*15360}{30+(50*0.7)}\}$ = 11,815.38

Increasing file upload rate of peers will result in decreasing overall file distribution time among all peers.

Question 3 [10 Marks]

Host A and B are communicating over a TCP connection, and Host B has already received from A all bytes up through byte 126. Suppose Host A then sends two segments to Host B back-to-back. The first and second segments contain 80 and 40 bytes of data, respectively. In the first segment, the sequence number is 127, the source port number is 302, and the destination port number is 80. Host B sends an acknowledgment whenever it receives a segment from Host A.

a. In the second segment sent from Host A to B, what are the sequence number, source port number, and destination port number? **[2]**

sequence number is 207

source port number is 302

destination port number is 80.

b. If the first segment arrives before the second segment, in the acknowledgement of the first arriving segment, what is the acknowledgment number, the source port number, and the destination port number? **[2]**

the acknowledgement number is 207

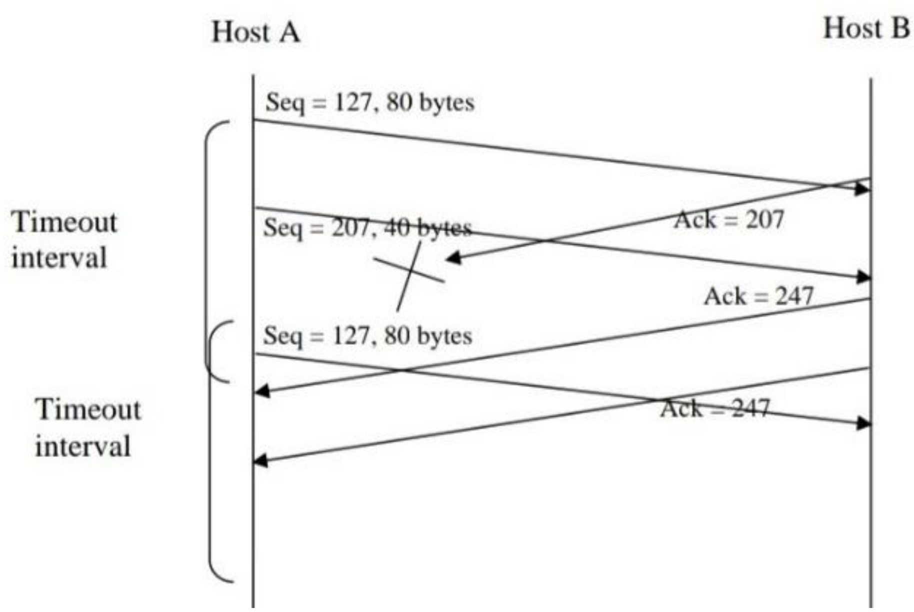
the source port number is 80

the destination port number is 302.

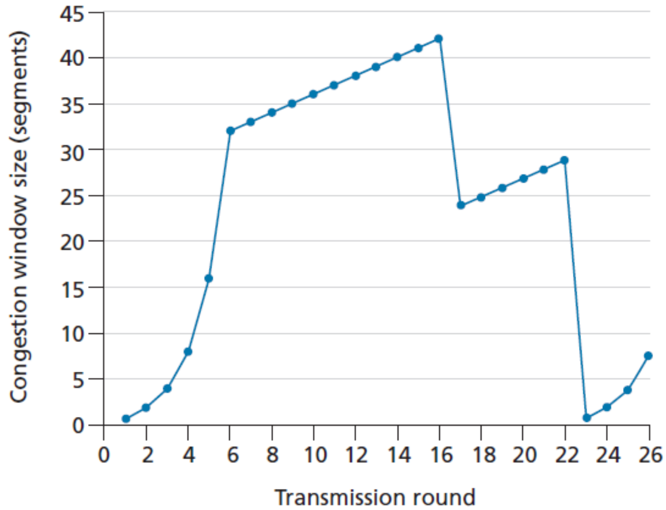
c. If the second segment arrives before the first segment, in the acknowledgement of the first arriving segment, what is the acknowledgment number? **[2]**

the acknowledgement number is 127, indicating that it is still waiting for bytes 127 and onwards.

d. Suppose the two segments sent by A arrive in order at B. The first acknowledgment is lost and the second acknowledgment arrives after the first timeout interval. Draw a timing diagram, showing these segments and all other segments and acknowledgments sent. (Assume there is no additional packet loss.) For each segment in your figure, provide the sequence number and the number of bytes of data; for each acknowledgement that you add, provide the acknowledgment number. **[4]**



Question 4 [15 Marks]



TCP window size as a function of time

Consider the Figure above. Assuming TCP Reno is the protocol experiencing the behavior shown above, answer the following questions. In all cases, you should provide a short discussion justifying your answer.

a. Identify the intervals of time when TCP slow start is operating. **[1]**

TCP slowstart is operating in the intervals [1,6] and [23,26]

b. Identify the intervals of time when TCP congestion avoidance is operating. **[1]**

TCP congestion avoidance is operating in the intervals [6,16] and [17,22]

c. After the 16th transmission round, is segment loss detected by a triple duplicate ACK or by a timeout? **[1]**

After the 16th transmission round, packet loss is recognized by a triple duplicate ACK. If there was a timeout, the congestion window size would have dropped to 1.

d. After the 22nd transmission round, is segment loss detected by a triple duplicate ACK or by a timeout? **[1]**

After the 22nd transmission round, segment loss is detected due to timeout, and hence the congestion window size is set to 1.

e. What is the initial value of ssthresh at the first transmission round? **[1]**

The threshold is initially 32, since it is at this window size that slow start stops and congestion avoidance begins.

f. What is the value of ssthresh at the 18th transmission round? **[1]**

The threshold is set to half the value of the congestion window when packet loss is detected. When loss is detected during transmission round 16, the congestion windows size is 42. Hence the threshold is 21 during the 18th transmission round.

g. What is the value of ssthresh at the 24th transmission round? **[1]**

The threshold is set to half the value of the congestion window when packet loss is detected. When loss is detected during transmission round 22, the congestion windows size is 29. Hence the threshold is 14 (taking lower floor of 14.5) during the 24th transmission round.

h. During what transmission round is the 70th segment sent? **[2]**

During the 1st transmission round, packet 1 is sent; packet 2-3 are sent in the 2nd transmission round; packets 4-7 are sent in the 3rd transmission round; packets 8-15 are sent in the 4th transmission round; packets 16-31 are sent in the 5th transmission round; packets 32-63 are sent in the 6th transmission round; packets 64 – 96 are sent in the 7th transmission round. Thus packet 70 is sent in the 7th transmission round.

i. Assuming a packet loss is detected after the 26th round by the receipt of a triple duplicate ACK, what will be the values of the congestion window size and of ssthresh? **[2]**

The threshold will be set to half the current value of the congestion window (8) when the loss occurred and congestion window will be set to the new threshold value + 3 MSS. Thus the new values of the threshold and window will be 4 and 7 respectively.

j. Suppose TCP Tahoe is used (instead of TCP Reno), and assume that triple duplicate ACKs are received at the 16th round. What are the ssthresh and the congestion window size at the 19th round? **[2]**

Threshold is 21, and congestion window size is 4.

k. Again suppose TCP Tahoe is used, and there is a timeout event at 22nd round. How many packets have been sent out from 17th round till 22nd round, inclusive? **[2]**

round 17, 1 packet; round 18, 2 packets; round 19, 4 packets; round 20, 8 packets; round 21, 16 packets; round 22, 21 packets. So, the total number is 52.

Question 5 [9 Marks]

Suppose that the three measured SampleRTT values are 106 ms, 120 ms, and 140 ms. Compute the EstimatedRTT after each of these SampleRTT values is obtained, using a value of α = 0.125 and assuming that the value of EstimatedRTT was 100 ms just before the first of these three samples were obtained. Compute also the DevRTT after each sample is obtained, assuming a value of β = 0.25 and assuming the value of DevRTT was 5 ms just before the first of these three samples was obtained. Last, compute the TCP TimeoutInterval after each of these samples is obtained.

(First calculating EstimatedRTT and then DevRTT is also correct)

DevRTT = (1- beta) * DevRTT + beta * | SampleRTT - EstimatedRTT |
EstimatedRTT = (1-alpha) * EstimatedRTT + alpha * SampleRTT
TimeoutInterval = EstimatedRTT + 4 * DevRTT

After obtaining first SampleRTT 106ms:
DevRTT = 0.75*5 + 0.25 * | 106 - 100 | = 5.25ms
EstimatedRTT = 0.875 * 100 + 0.125 * 106 = 100.75 ms
TimeoutInterval = 100.75+4*5.25 = 121.75 ms

After obtaining 120ms:
DevRTT = 0.75*5.25 + 0.25 * | 120 – 100.75 | = 8.75 ms
EstimatedRTT = 0.875 * 100.75 + 0.125 * 120 = 103.16 ms
TimeoutInterval = 103.16+4*8.75 = 138.16 ms

After obtaining 140ms:
DevRTT = 0.75*8.75 + 0.25 * | 140 – 103.16 | = 15.77 ms
EstimatedRTT = 0.875 * 103.16 + 0.125 * 140 = 107.76 ms
TimeoutInterval = 107.76+4*15.77 = 170.84 ms