CS 176A: Homework 2

Bharat Kathi

Part 1:

1. For each of the following fields in the TCP header, describe the function/purpose of that field:

a) Sequence number
Used to track the order of bytes sent in a TCP connection. It helps the receiver reassemble packets correctly and detect duplicate or lost packets.

b) Acknowledgement number
Specifies the next expected byte from the sender. It confirms receipt of data and ensures reliable communication.

c) ACK bit
A control flag indicating that the acknowledgment field is valid. When set to 1, it signals that the acknowledgment number should be processed.

d) Receiver advertised window
Specifies the amount of available buffer space on the receiver's side. It helps manage flow control to prevent the sender from overwhelming the receiver.

e) Source port number
Identifies the sending application's port number, allowing the receiver to determine which process should handle the received segment.

2. For each of the following TCP connection management stages, name the message type that accomplishes that function (i.e. SYN, FIN, etc.)

a) A message indicating that the sending side is terminating the connection
FIN

b) A message from server to client ACKing receipt of a SYN message and indicating the willingness of the server to establish a TCP connection with the client
SYN-ACK

c) A message from client to server initiating a connection request
SYN

d) A message sent in response to a request to terminate a connection
ACK

e) A general purpose error message during connection set up or tear down, indicating the referenced connection should be shut down.
RST

3. Describe what is meant by transport-layer multiplexing and demultiplexing.

Multiplexing allows multiple applications to use the network simultaneously by assigning different port numbers to each application. The transport layer at the sender combines data from different applications and passes it to the network layer. Demultiplexing occurs at the receiver’s transport layer, where it examines the destination port number and directs the data to the appropriate application.

4. Why is the UDP header length field needed?

The UDP header length field specifies the total length of the UDP segment, including the header and data. It is needed because IP packets may have padding, and UDP needs an explicit length field to determine where the segment ends.
It also allows receivers to know how much data to extract from the packet.

5. Consider the following plot of TCP window size as a function of time. Assuming TCP Reno is the protocol experiencing the behavior shown in the figure, 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.
0-6

b) Identify the intervals of time when TCP congestion avoidance is operating.
6-16

c) After the 16th transmission round, is segment loss detected by a triple duplicate ACK or by a timeout?
Triple duplicate ack

d) After the 22nd transmission round, is segment loss detected by a triple duplicate ACK or by a timeout?
timeout

e) What is the initial value of ssthresh at the first transmission round?
Sshthresh = 32

f) What is the value of ssthresh at the 18th transmission round?
Sshthresh = 16

g) What is the value of ssthresh at the 24th transmission round?
Sshthresh = 12

h) 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?
Sshthresh = 5
Congestion window = 8

6. In the following question, you will compare the performance of GBN, SR and TCP Reno (no delayed ACK). Assume that the timeout values for all three protocols are sufficiently long such that 5 consecutive data segments and their corresponding ACKs can be received (if not lost in the channel) by the receiving host (Host B) and the sending host (Host A) respectively. Suppose Host A sends 5 data segments to Host B, and the 2nd segment (sent from A) is lost. In the end, all 5 data segments are correctly received by Host B.

a) For each of the three protocols, how many segments has Host A sent in total and how many ACKs has Host B sent in total? What are their sequence numbers?

Go-Back-N (GBN)

- Segments sent: 9 (1, 2, 3, 4, 5, then retransmitting 2, 3, 4, 5)
- ACKs received: 5 (ACK 1, ACK 2, ACK 3, ACK 4, ACK 5)

Selective Repeat (SR)

- Segments sent: 6 (1, 2, 3, 4, 5, then retransmitting 2)
- ACKs received: 6 (ACK 1, ACK 2, ACK 3, ACK 4, ACK 5, ACK 2)

TCP Reno

- Segments sent: 6 (1, 2, 3, 4, 5, then retransmitting 2)
- ACKs received: 6 (ACK 1, ACK 2, ACK 3, ACK 4, ACK 5, ACK 2)

b) If the timeout values for all three protocols are much longer than 5 RTT, then which protocol will successfully deliver all 5 data segments in the shortest time interval?

TCP Reno will deliver all 5 segments the fastest.

Part 2:

1. What is the IP address and TCP port number used by the client computer (source) that is transferring the alice.txt file to gaia.cs.umass.edu? To answer this question, it's probably easiest to select an HTTP message and explore the details of the TCP packet used to carry this HTTP message, using the "details of the selected packet header window" (refer to Figure 2 in the "Getting Started with Wireshark" Lab if you're uncertain about the Wireshark windows).

192.168.86.68.12:80

2. What is the IP address of gaia.cs.umass.edu? On what port number is it sending and receiving TCP segments for this connection?

128.119.245.12:80

3. What is the sequence number of the TCP SYN segment that is used to initiate the TCP connection between the client computer and gaia.cs.umass.edu? (Note: this is the "raw" sequence number carried in the TCP segment itself; it is NOT the packet # in the "No." column in the Wireshark window. Remember there is no such thing as a "packet number" in TCP or UDP; as you know, there are sequence numbers in TCP and that's what we're after here. Also note that this is not the relative sequence number with respect to the starting sequence number of this TCP session.). What is it in the segment that identifies the segment as a SYN segment?

Sequence number (raw): 4236649188
The flags field should contain SYN.

4. What is the sequence number of the SYNACK segment sent by gaia.cs.umass.edu to the client computer in reply to the SYN? What is the value of the ACKnowledgement field in the SYNACK segment? How did gaia.cs.umass.edu determine that value?

Sequence Number (raw): 1068969752

Acknowledgment number (raw): 4236649188
The acknowledgement number is the initial sequence number + 1.

5. What is the raw (not relative!) sequence number of the TCP segment containing the HTTP POST command? Note that in order to find the POST command, you'll need to dig into the packet content field at the bottom of the Wireshark window, looking for a segment with the ASCII text "POST" within its DATA field.

Sequence Number (raw): 4236649188

6. Consider the TCP segment containing the HTTP POST as the first segment in the TCP connection.

• At what time was the first segment (the one containing the HTTP POST) in the data-transfer part of the TCP connection sent?
0.024047

• At what time was the ACK for this first data-containing segment received?
0.052671

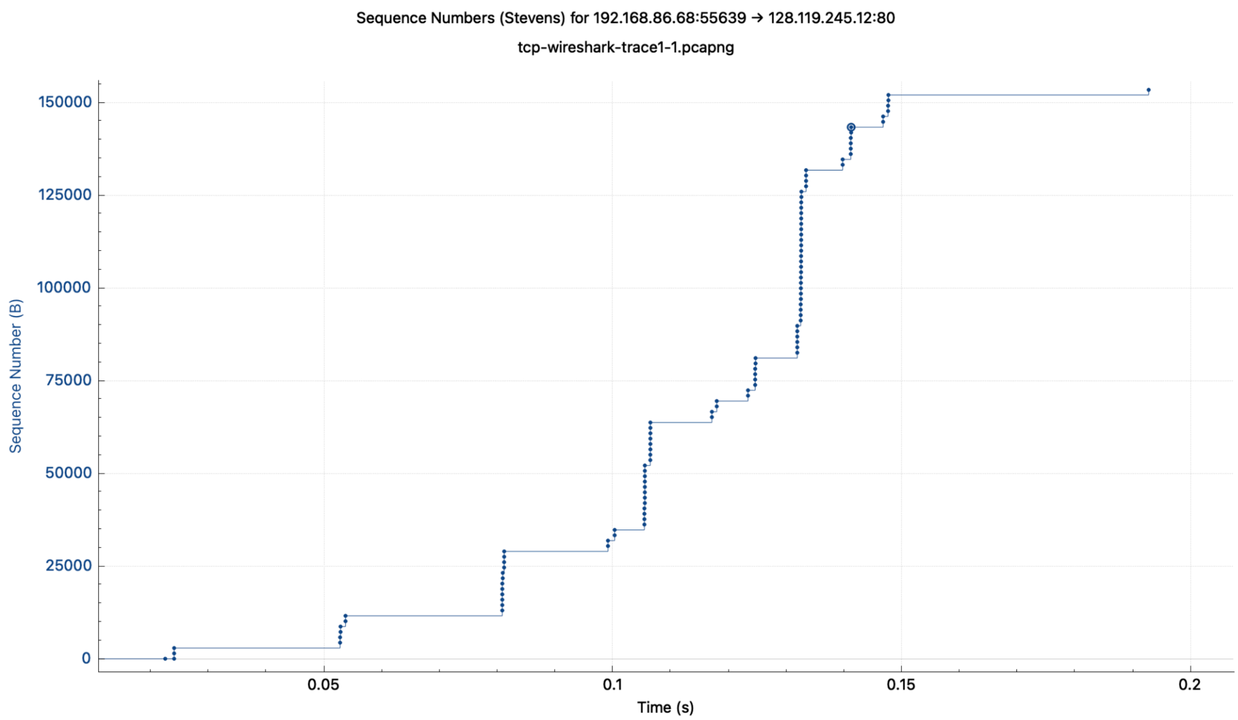
• What is the RTT for this first data-containing segment?
0.052671 - 0.024047 = 0.028624

• What is the RTT value the second data-carrying TCP segment and its ACK?
0.052676 - 0.024048 = 0.028628

• What is the EstimatedRTT value (see Section 3.5.3, in the text) after the ACK for the second data-carrying segment is received? Assume that in making this calculation after receiving the ACK for the second segment, that the initial value of EstimatedRTT is equal to the measured RTT for the first segment, and then is computed using the EstimatedRTT equation on page 242, and a value of a = 0.125.

$(1 - 0.125) \times 0.028624 + 0.125 \times 0.028628 = 0.0286245$

7. Use the Time-Sequence-Graph(Stevens) plotting tool to view the sequence number versus time plot of segments being sent from the client to the gaia.cs.umass.edu server. Consider the “fleets” of packets sent around t = 0.025, t = 0.053, t = 0.082 and t = 0.1. Comment on whether this looks as if TCP is in its slow start phase, congestion avoidance phase or some other phase. The figure below shows a slightly different view of this data.



It looks as if it's in the slow start phase. It may be going to congestion avoidance phase after 0.1s.