E1:

Q1*:* What is the IP address of gaia.cs.umass.edu? On what port number is it sending and receiving TCP segments for this connection? What are the IP address and TCP port numbers used by the client computer (source) that is transferring the file to gaia.cs.umass.edu?

A: IP address of gaia.cs.umass.edu: 128.119.245.12
Port number is 80.
IP address for client computer: 192.168.1.102
Port number is 1161.

Q2: What is the sequence number of the TCP segment containing the HTTP POST command? Note that 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 a "POST" within its DATA field.

A: The sequence number is 232129013

Q3: Consider the TCP segment containing the HTTP POST as the first segment in the TCP connection.

(a) What are the sequence numbers of the first six segments in the TCP connection (including the segment containing the HTTP POST) sent from the client to the webserver (Do not consider the ACKs received from the server as part of these six segments)?

(b) At what time was each segment sent? When was the ACK for each segment received? Given the difference between when each TCP segment was sent and when its acknowledgement was received, what is the RTT value for each of the six segments?

(c) What is the *EstimatedRTT* value (see relevant parts of Section 3.5 or lecture slides) after receiving each ACK? Assume that the initial value of *EstimatedRTT* is equal to the measured RTT (*SampleRTT*) for the first segment and then is computed using the *EstimatedRTT* equation for all subsequent segments. Set alpha to 0.125.

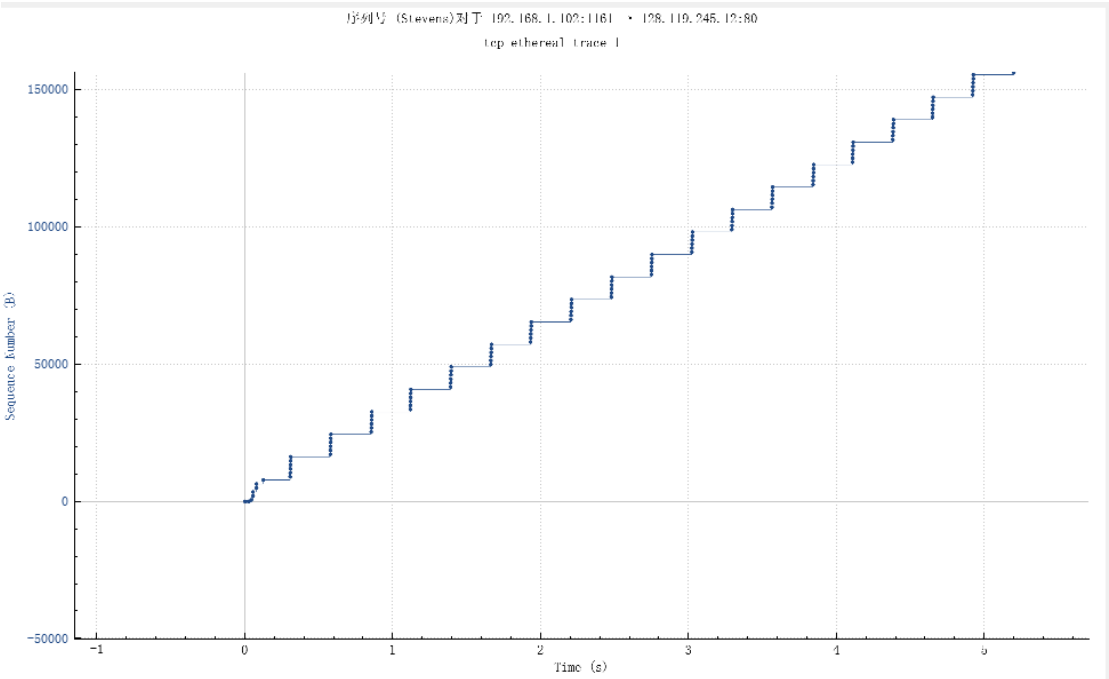
Segment	Seq NO.	length	Sent time	Received time	SampleRTT	EstimatedRTT
1	232129013	565	0.026477	0.053937	0.02746	0.02746
2	232129578	1460	0.041737	0.077294	0.035557	0.028472125
3	232131038	1460	0.054026	0.124085	0.070059	0.033670484
4	232132498	1460	0.05469	0.169118	0.114428	0.043765174
5	232133958	1460	0.077405	0.217299	0.139894	0.055781277
6	232135418	1460	0.078157	0.167802	0.089645	0.060014242

Q4: What is the length of each of the first six TCP segments?

A: In the chart above

Q5: What is the minimum amount of available buffer space advertised at the receiver for the entire trace? Does the lack of receiver buffer space ever throttle the sender?
A: 5840. When the receiver's buffer is full, it can send a "window size" signal to the sender, indicating that it cannot receive any more data until some of the existing data is consumed. So, the answer is yes.

Q6: Are there any retransmitted segments in the trace file? To answer this question, what did you check for (in the trace)?
A: NO, there are no retransmitted segments.



Q7: How much data does the receiver typically acknowledge in an ACK? Can you identify cases where the receiver is ACKing every other received segment (recall the discussion about delayed acks from the lecture notes or Section 3.5 of the text)?
A: The receiver is acknowledging the receipt of 1460 bytes in segments 53-57 and then 892 bytes in segment 58. Through the data in track list, segment 54,55,56, used one ack number. To identify it, can check if the client has recent the package or not.

Q8: What is the TCP connection's throughput (bytes transferred per unit of time)? Explain how you calculated this value.
A: (232293103-232129013-1)/(5.455830-0.026477)=
164089/5.429353=30,222.56979791146

E2:

Q1: What is the sequence number of the TCP SYN segment that is used to initiate the TCP connection between the client computer and server?
A: seq=2818463618

Q2: What is the sequence number of the SYNACK segment sent by the server to the client computer in reply to the SYN? What is the value of the Acknowledgement field in the SYNACK segment? How did the server determine that value?
A: seq=1247095790 Ack=2818463619

Q3: What is the sequence number of the ACK segment sent by the client computer in response to the SYNACK? What is the value of the Acknowledgment field in this ACK segment? Does this segment contain any data?
A: seq=2818463619 Ack=1247095791

Q4: Who has done the active close? Is it the client or the server? How you have determined this? What type of closure has been performed? 3 Segment (FIN/FINACK/ACK), 4 Segment (FIN/ACK/FIN/ACK) or Simultaneous close?
A: The client initiates the close by sending a TCP FIN segment (packet number 304) to the server to indicate that it has no more data to send. The server responds with a TCP FIN/ACK segment (packet number 305) to acknowledge receipt of the FIN segment and inform the client that it too has no more data to send. There is a 3-way handshake (FIN/FINACK/ACK) type of closure has been performed to close the TCP connection.

Q5: How many data bytes have been transferred from the client to the server and from the server to the client during the whole duration of the connection? What relationship does this have with the Initial Sequence Number and the final ACK received from the other side?
A: the number of data bytes transferred from the client to the server is:
2818463652 - 2818463619 + 1 = 34 bytes.
And the number of data bytes transferred from the server to the client is:
1247095831 - 1247095791 + 1 = 41 bytes.
The relationship between the Initial Sequence Number and the final ACK received from the other side is that the ISN is used to initiate the sequence numbers in subsequent packets, while the final ACK indicates that the other side has received and acknowledged all the data sent up to that point.