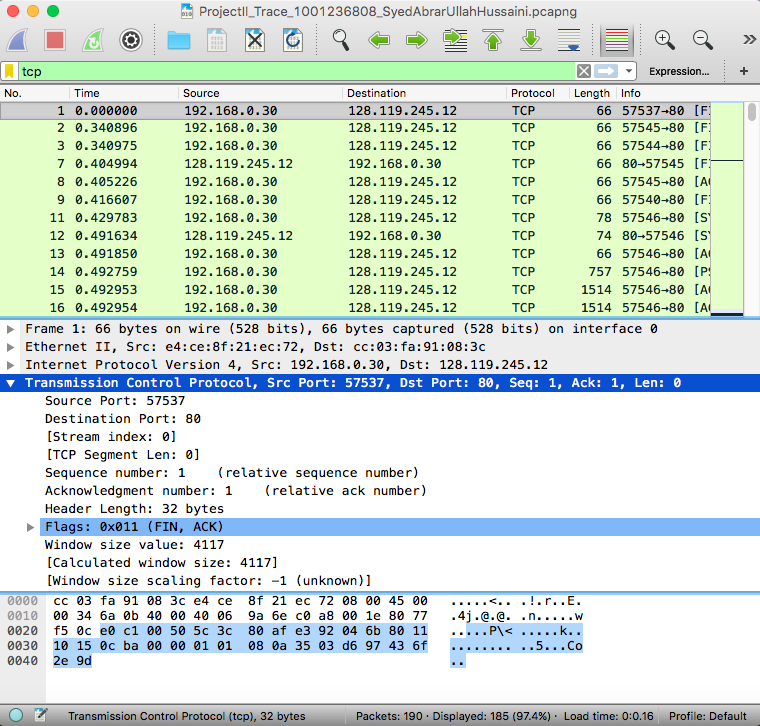
**COMPUTER NETWORKS PROJECT II**

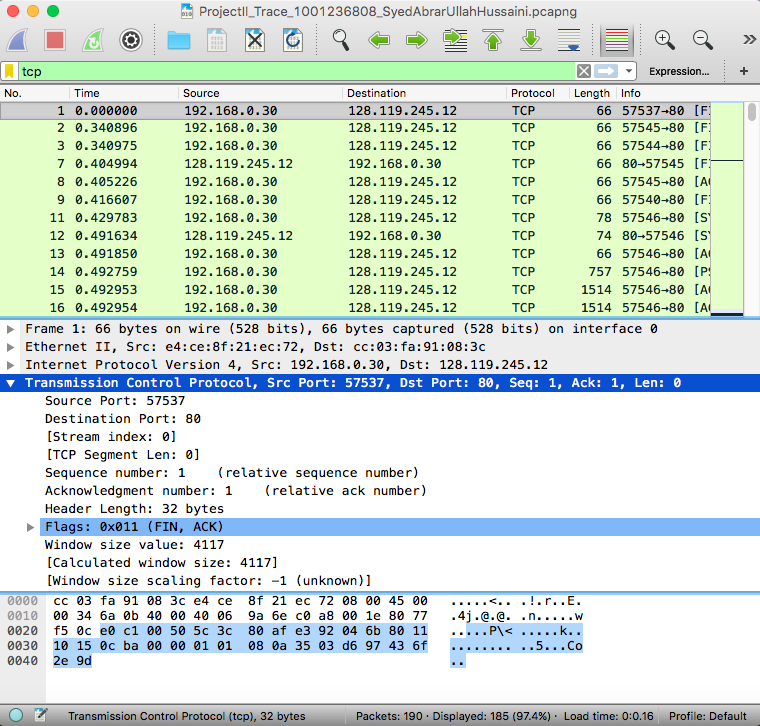
1. What is the IP address and TCP port number used by the client computer (source) that is transferring the file to gaia.cs.umass.edu? To answer this question, it’s probably easiest to select an HTTP message and explore the details of the TCP packet used to carry this HTTP message, using the “details 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.

A) My IP Address is 192.168.0.30 TCP source port number is 57537. See screenshot below



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?

A) destination IP Address is 128.119.245.12 and destination Port Number is 80. See the screenshot below

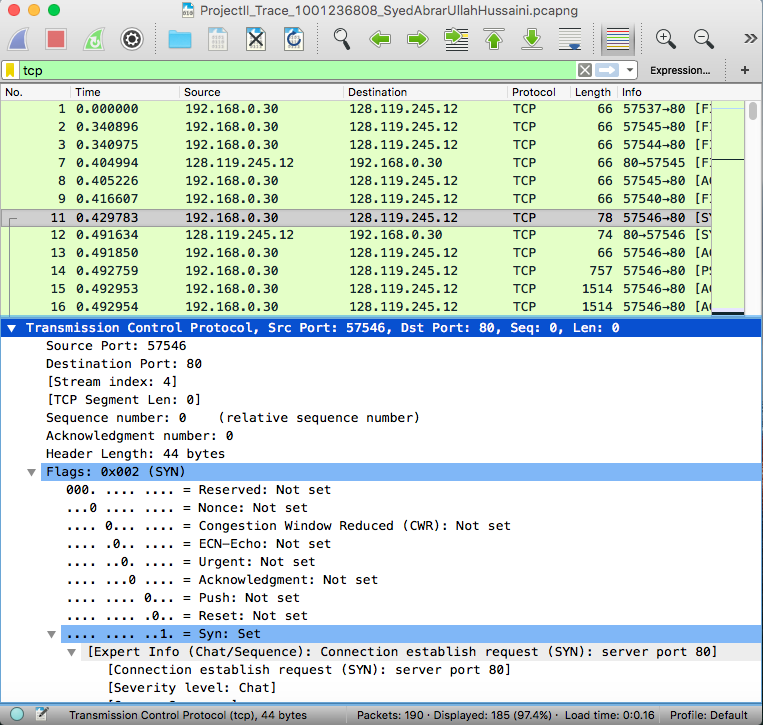


3. What is the IP address and TCP port number used by your client computer (source) to transfer the file to gaia.cs.umass.edu?

A) Same as Question 1

4. 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? What is it in the segment that identifies the segment as a SYN segment?

A) The Sequence Number is 0. Segment that identifies SYN segment is flags where SYN is SET(See below Screenshot)



5. 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? What is it in the segment that identifies the segment as a SYNACK segment?

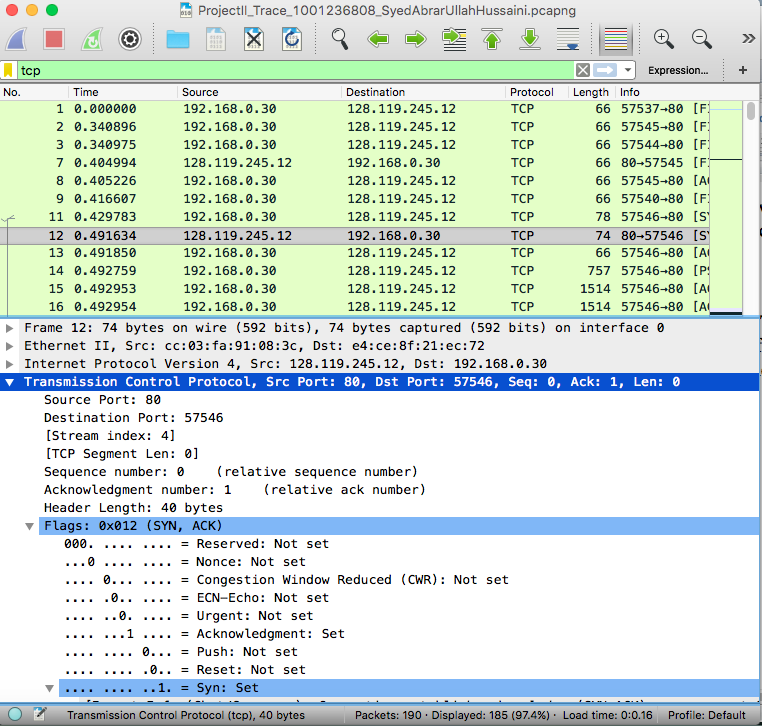
A) The Sequence Number of SYNACK segment is (relative sequence number=0).

Value of Acknowledgement field is (Ack=1).

This value is determined using TCP mechanism that when SYN message is sent with sequence

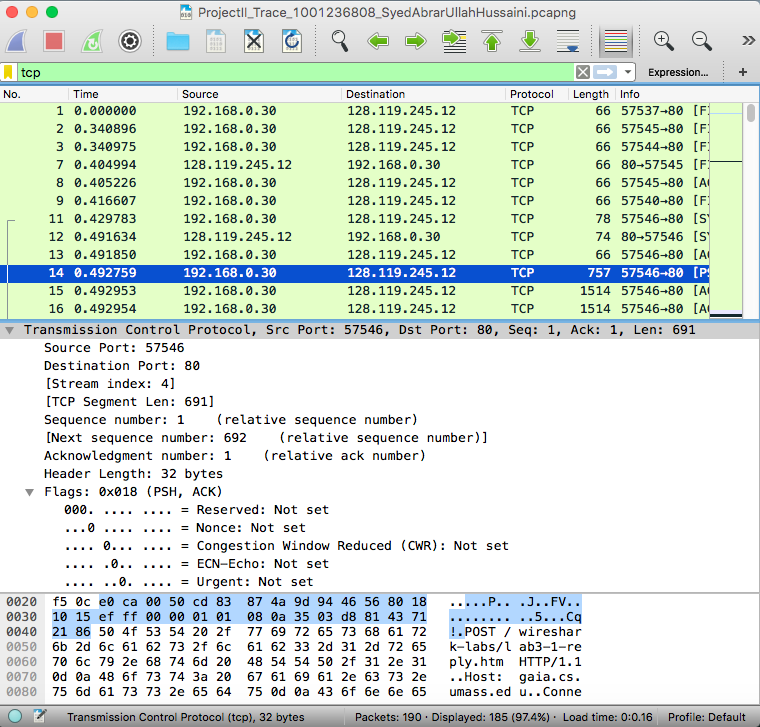
Number 0 then SYNACK is received with ACK number(sequence number value+1) 0+1=1.

Flags identifies SYNACK segment with SYN and ACK SET.(Screenshot below)



6. What is the 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 a “POST” within its DATA field.

A) Sequence Number containing HTTP POST is 1.(Screenshot below)



7. Consider the TCP segment containing the HTTP POST as the first segment in the TCP connection. What are the sequence numbers of the first six segments in the TCP connection (including the segment containing the HTTP POST)? 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? What is the EstimatedRTT value (see Section 3.5.3, page 239 in text) after the receipt of each ACK? Assume that the value of the EstimatedRTT is equal to the measured RTT for the first segment, and then is computed using the EstimatedRTT equation on page 239 for all subsequent segments.

A) 1) Seq = 1 sent at 0.492759; ACK received at 0.555671; RTT = 0.0629141

2) Seq = 692 sent at 0.492953; ACK received at 0.560275; RTT = 0.067322

3) Seq = 2140 sent at 0.492954; ACK received at 0.560282; RTT = 0.067328

4) Seq = 3588 sent at 0.555772; ACK received at 0.620151; RTT = 0.064431

5) Seq = 5036 sent at 0.560404; ACK received at 0.620932; RTT = 0.060528

6) Seq = 6484 sent at 0.560405; ACK received at 0.625339; RTT = 0.064934

The HTTP POST segment is considered as the first segment. Segments 1 – 6 are No. 14, 15, 16, 18, 21, and 22 in this trace respectively. The ACKs of segments 1 – 6 are No. 17, 19, 20, 25, 28, and 31 in this trace.

EstimatedRTT = 0.875 \* EstimatedRTT + 0.125 \* SampleRTT

EstimatedRTT after the receipt of the ACK of segment 1: EstimatedRTT = RTT for Segment 1 = 0.0629141 second

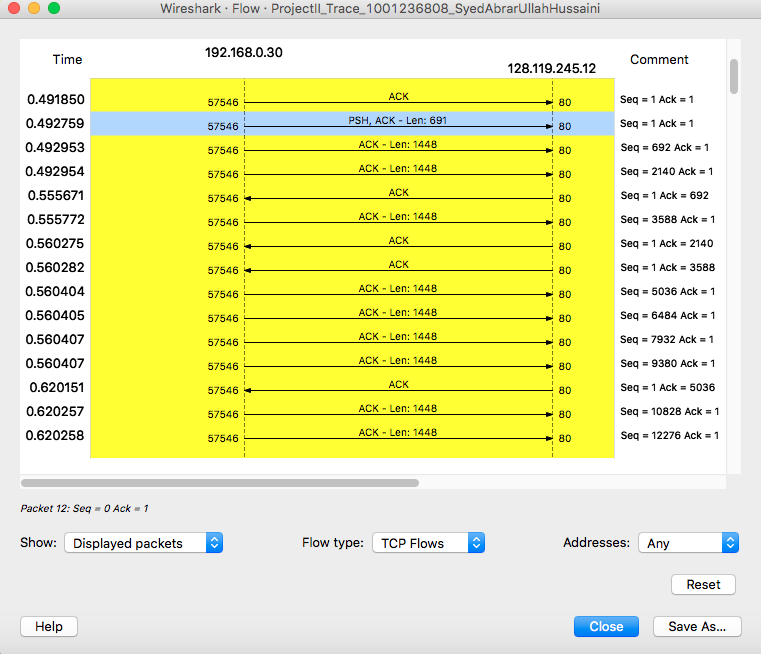
EstimatedRTT after the receipt of the ACK of segment 2: EstimatedRTT = 0.875 \* 0.0629141 + 0.125 \* 0.067322 = 0.0634650

EstimatedRTT after the receipt of the ACK of segment 3: EstimatedRTT = 0.875 \* 0.0634650 + 0.125 \* 0.067328 = 0.063947

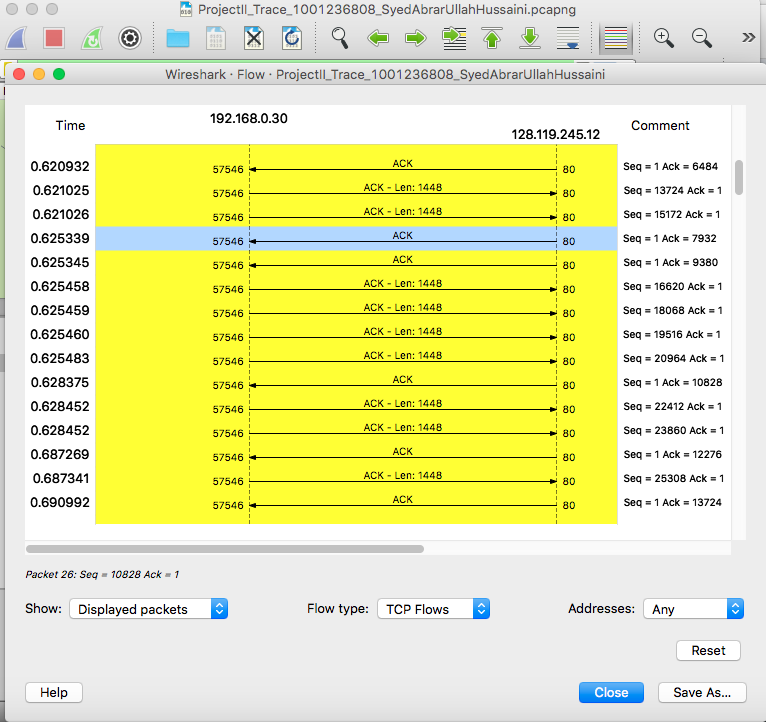
EstimatedRTT after the receipt of the ACK of segment 4: EstimatedRTT = 0.875 \* 0.063947+ 0.125 \* 0.064431 = 0.0640075

EstimatedRTT after the receipt of the ACK of segment 5: EstimatedRTT = 0.875 \* 0.0640075 + 0.125 \* 0.060528 = 0.063631

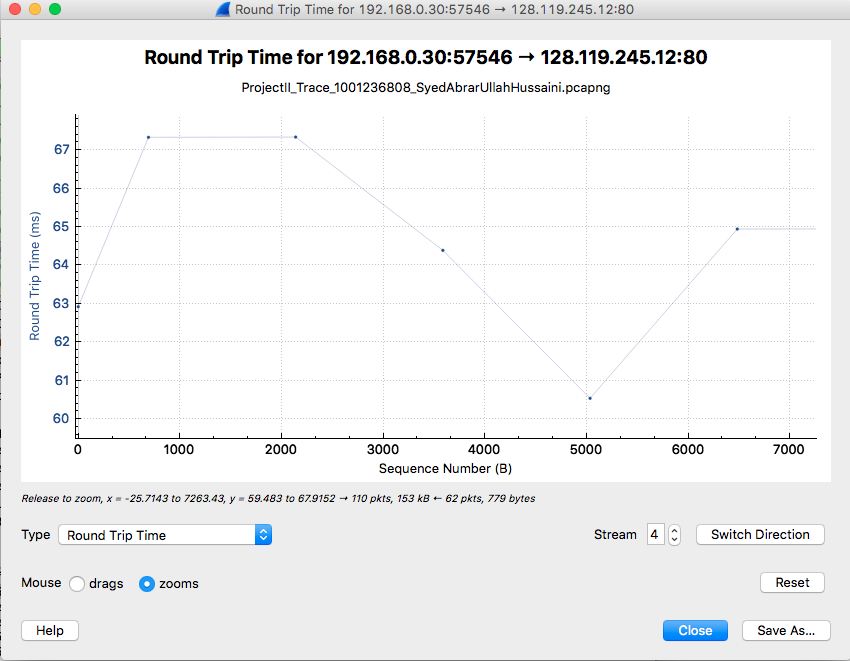
EstimatedRTT after the receipt of the ACK of segment 6: EstimatedRTT = 0.875 \* 0.063631 + 0.125 \* 0.064934 = 0.063793 second



Flow Graph of 1st 6 segments(Start highlighted)



Flow Graph of 1st 6 segments(End)



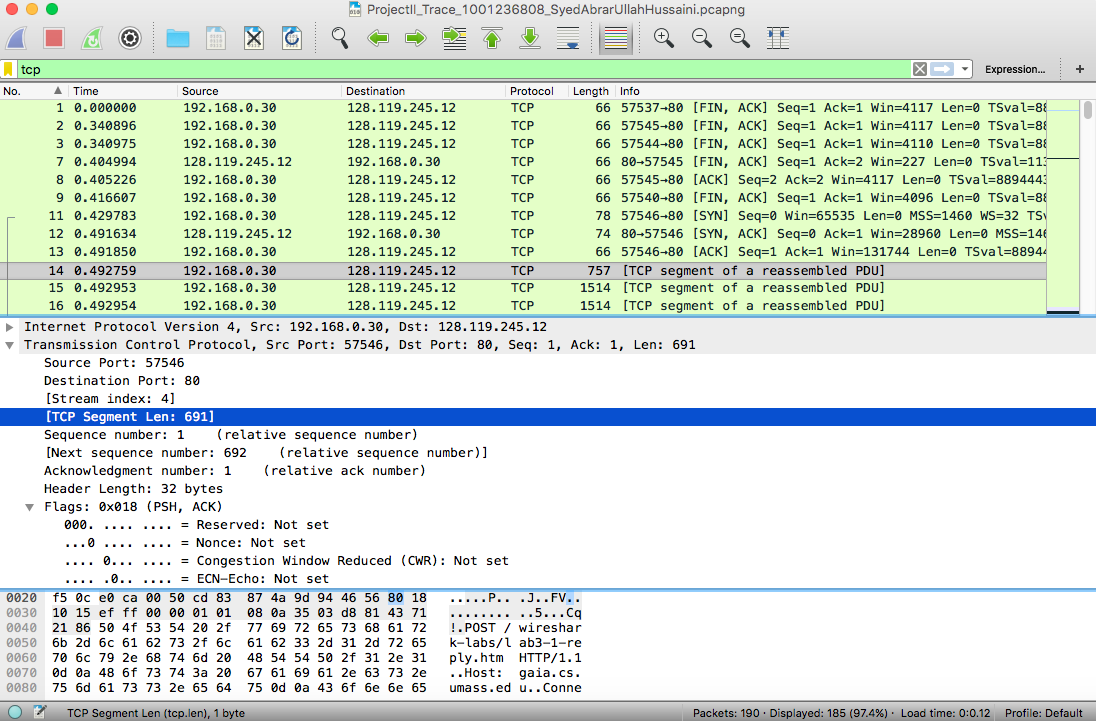
Round Trip Time Graph of first 6 Segments

8. What is the length of each of the first six TCP segments?

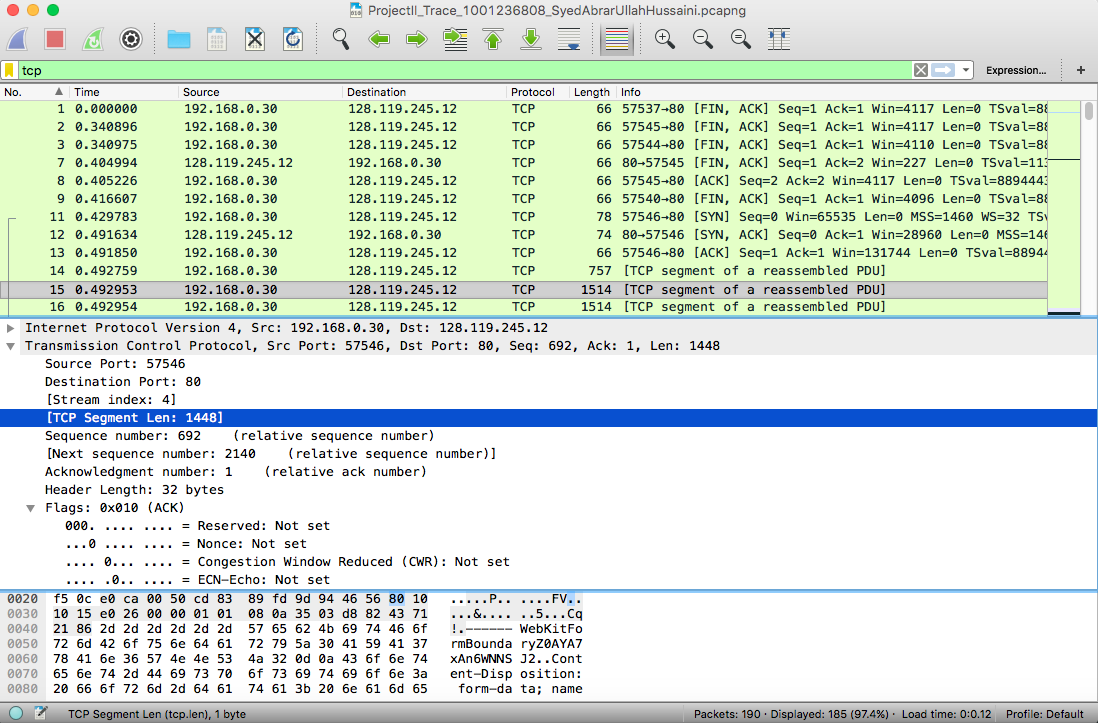
A) Segments 1 – 6 are No. 14, 15, 16, 18, 21, and 22 in this trace respectively.

Length of first TCP segment (containing the HTTP POST): 691 bytes

Length of other 5 TCP segments : 1448 bytes (Screenshots below)



TCP Segment Length of 1st segment



TCP Segment Length of other 5 segments

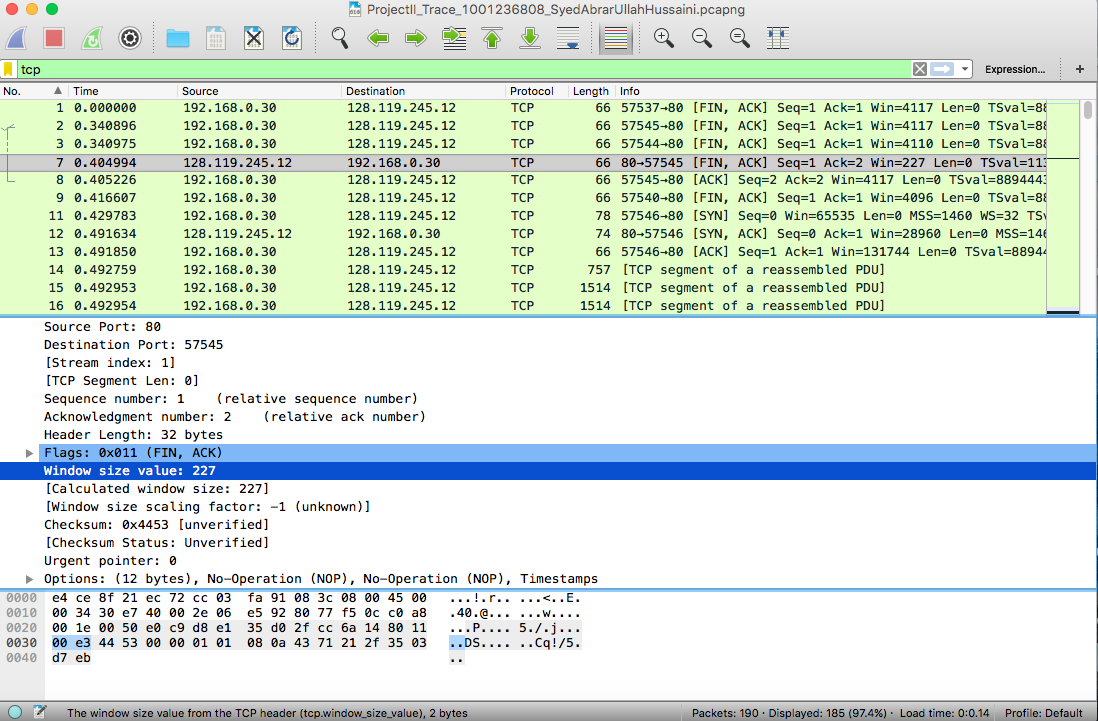
9. What is the minimum amount of available buffer space advertised at the received for the entire trace? Does the lack of receiver buffer space ever throttle the sender?

A) The minimum amount of buffer space (receiver window) advertised at gaia.cs.umass.edu for the trace is 227 bytes which is at no 7 in this trace.

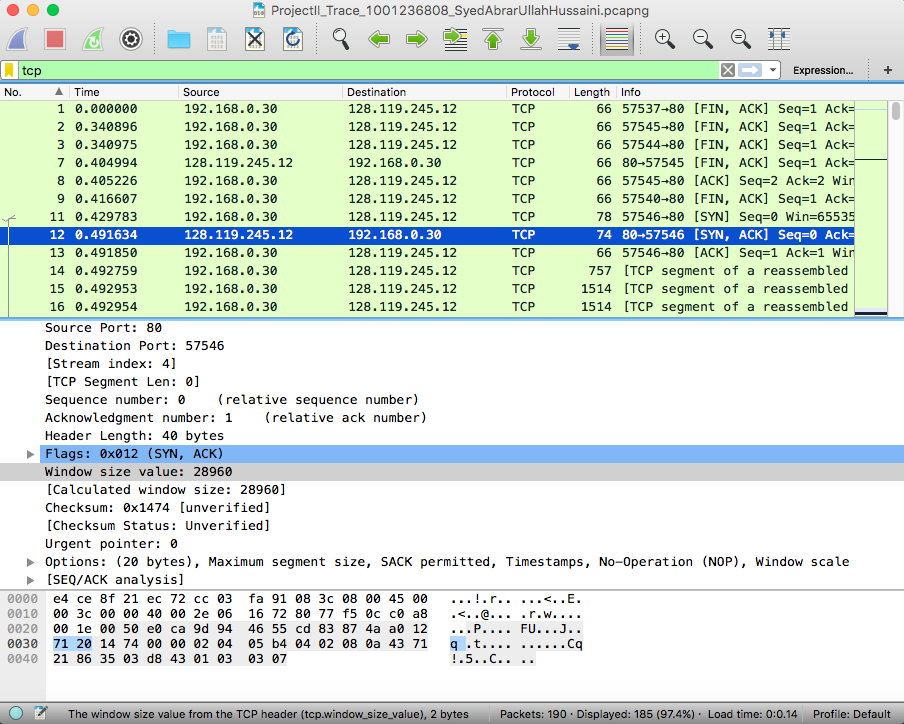
On SYNACK the receiver window size is 28930 bytes which is at number 12 in this trace.

This receiver window grows steadily until a maximum receiver buffer size of 183296 bytes.

The sender is never throttled due to lacking of receiver buffer space by inspecting this trace.



Minimum Receiver Window Size



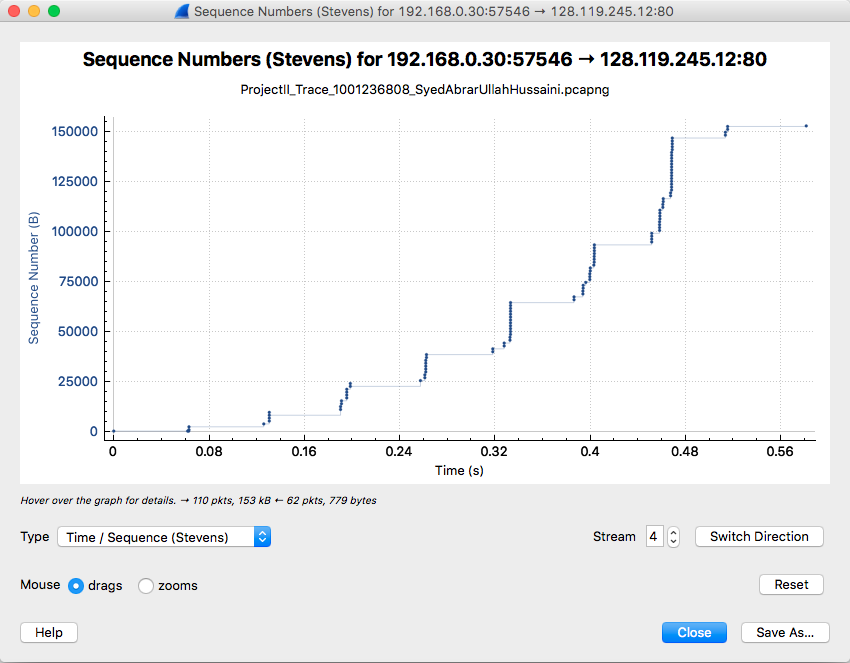
SYNACK Window size

10. Are there any retransmitted segments in the trace file? What did you check for (in the trace) in order to answer this question?

A) There are no retransmitted segments in the trace file for large data(Alice in Wonderland). We can verify this by checking the sequence numbers of the TCP segments in the trace file.

In the TimeSequence-Graph (Stevens) of this trace, all sequence numbers from the source (192.168.0.30) to the destination (128.119.245.12) are increasing monotonically with respect to time. If there is a retransmitted segment, the sequence number of this retransmitted segment should be smaller than those of its neighboring segments.

However in the end the FIN,ACK packets are retransmitted which are at number 56,188,189,190 in this trace. In the trace it shows TCP Retransmission.



Sequence numbers of the segments from the source (192.168.0.30) to the destination (128.119.245.12)

11. How much data does the receiver typically acknowledge in an ACK? Can you identify cases where the receiver is ACKing every other received segment (see Table 3.2 on page 247 in the text).

A) The acknowledged sequence numbers of the ACKs are listed as follows

Where ACK data is ACK sequence number – previous ACK Sequence number

ACK=1; ACK Sequence Number = 692; ACK data = 692

ACK=2; ACK Sequence Number = 2140; ACK data = 1448

ACK=3; ACK Sequence Number = 3588; ACK data = 1448

ACK=4; ACK Sequence Number = 5036; ACK data = 1448

ACK=5; ACK Sequence Number = 6484; ACK data = 1448

ACK=6; ACK Sequence Number = 7932; ACK data = 1448

ACK=7; ACK Sequence Number = 9380; ACK data = 1448

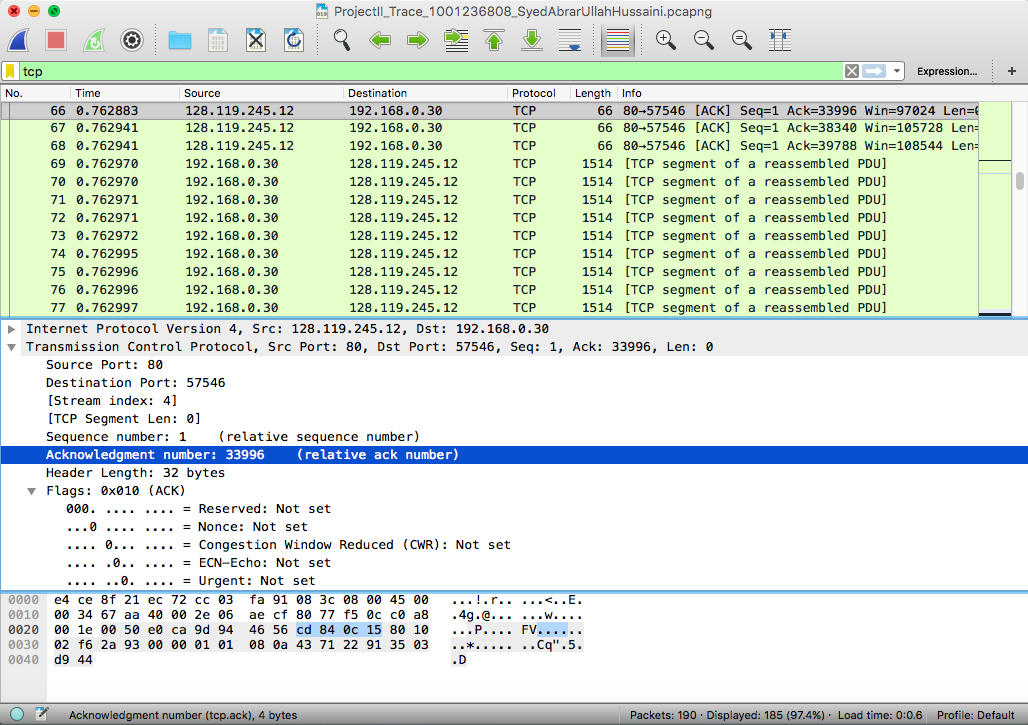
ACK=8; ACK Sequence Number = 10828; ACK data = 1448

ACK=9; ACK Sequence Number = 12276; ACK data = 1448 and so on

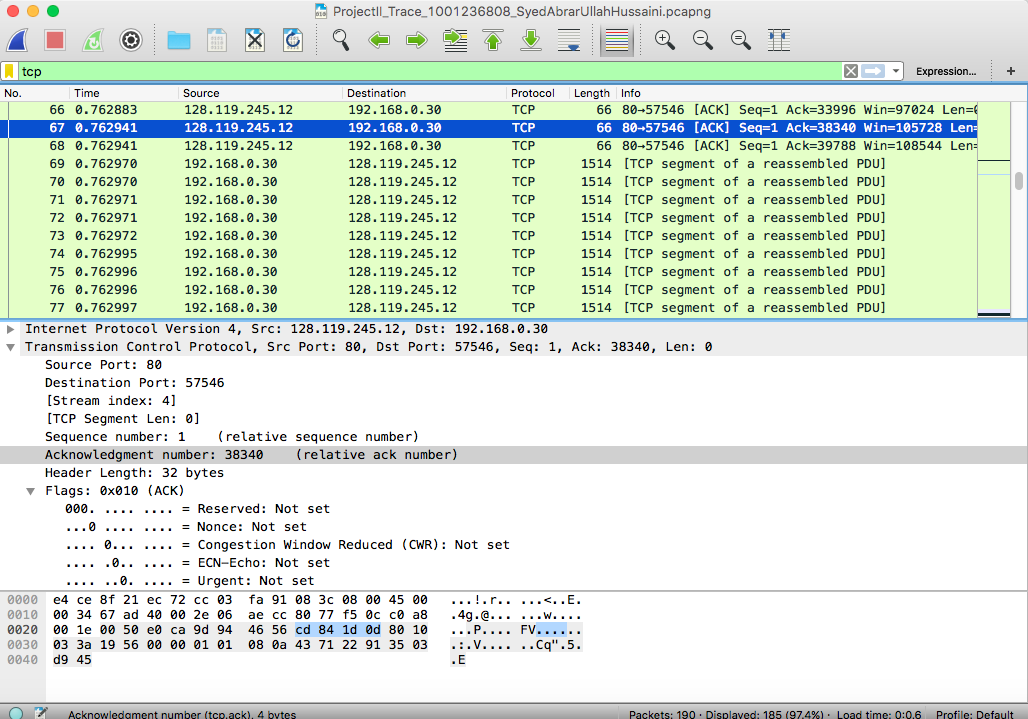
The difference between the acknowledged sequence numbers of two consecutive ACKs

indicates the data received by the server between these two ACKs. By inspecting the

amount of acknowledged data by each ACK, there are cases where the receiver is ACKing every other segment. For example, segment of No. 66,67 acknowledged data with 4344 bytes = 1448\*3 bytes.



Number 66 that ACKed data 4344 bytes

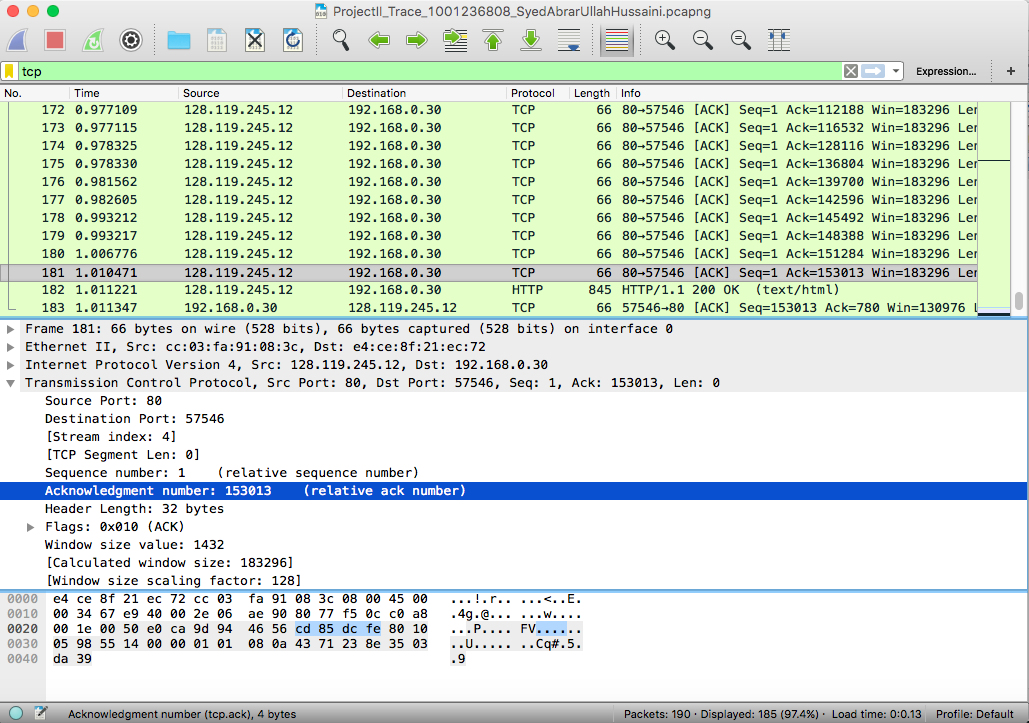


Number 67 that ACKed data 4344 bytes

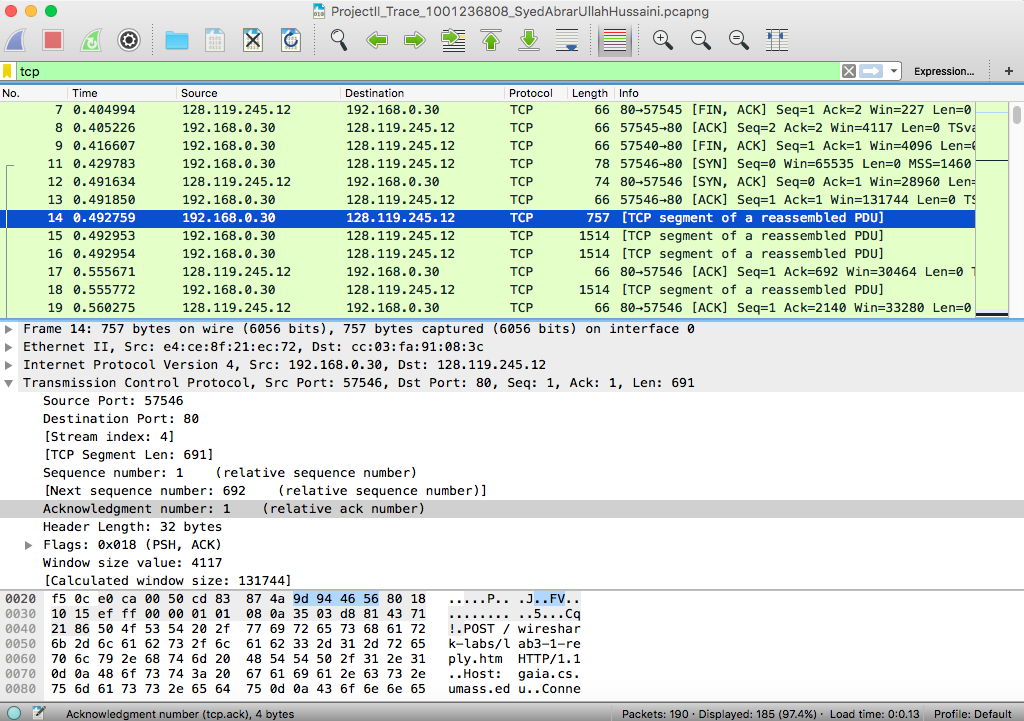
12. What is the throughput (bytes transferred per unit time) for the TCP connection? Explain how you calculated this value.

A) The average throughput for this TCP connection is computed as the ratio between the total amount of data and the total transmission time. The total amount data transmitted can be computed by the difference between the sequence number of the first TCP segment (1 byte for No. 14 in trace) and the acknowledged sequence number of the last ACK (153013 bytes for No. 181 in trace). Therefore, the total data are 153013 - 1 = 153012 bytes. The whole transmission time is the difference of the time instant of the first TCP segment (0.492759 seconds for No. 14 in trace) and the time instant of the last ACK (1.010471 second for No. 181 in trace). Therefore, the total transmission time is 1.010471 - 0.492759 = 0.517712 seconds. Hence, the throughput for the TCP connection is computed as 153012/0.517712 = 295554.28 bytes/sec or 288.62 KByte/sec considering 1kb=1024bytes.

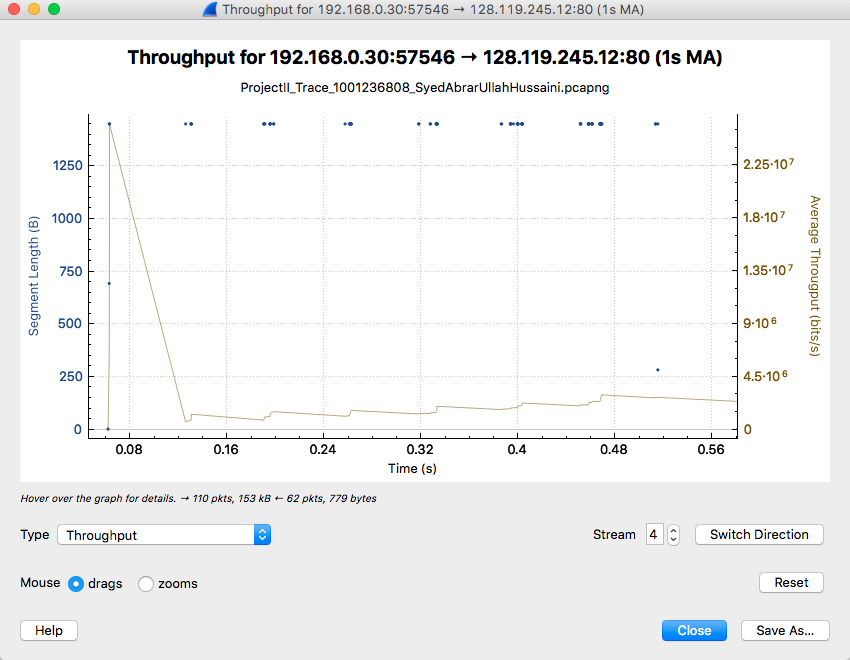
There is also Average Throughput graph in wireshark which shows Average throughpur 2.365 x 1000000 = 2365000 bits/sec = 288.69kbyte/sec



Last Sequence Number



First Sequence Number



Average Throughput Graph