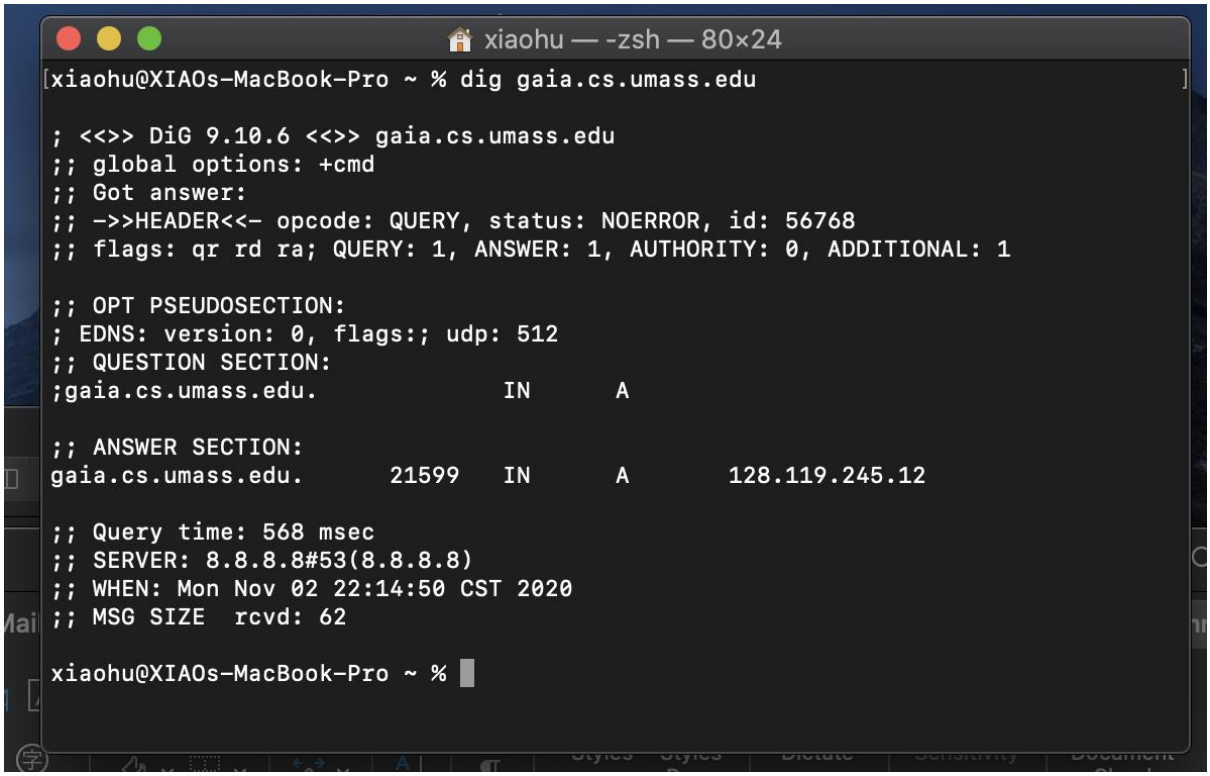
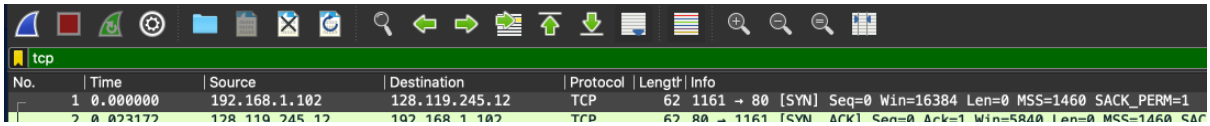
Written by XiaoHu z5223731
Exercise 1 Answer:

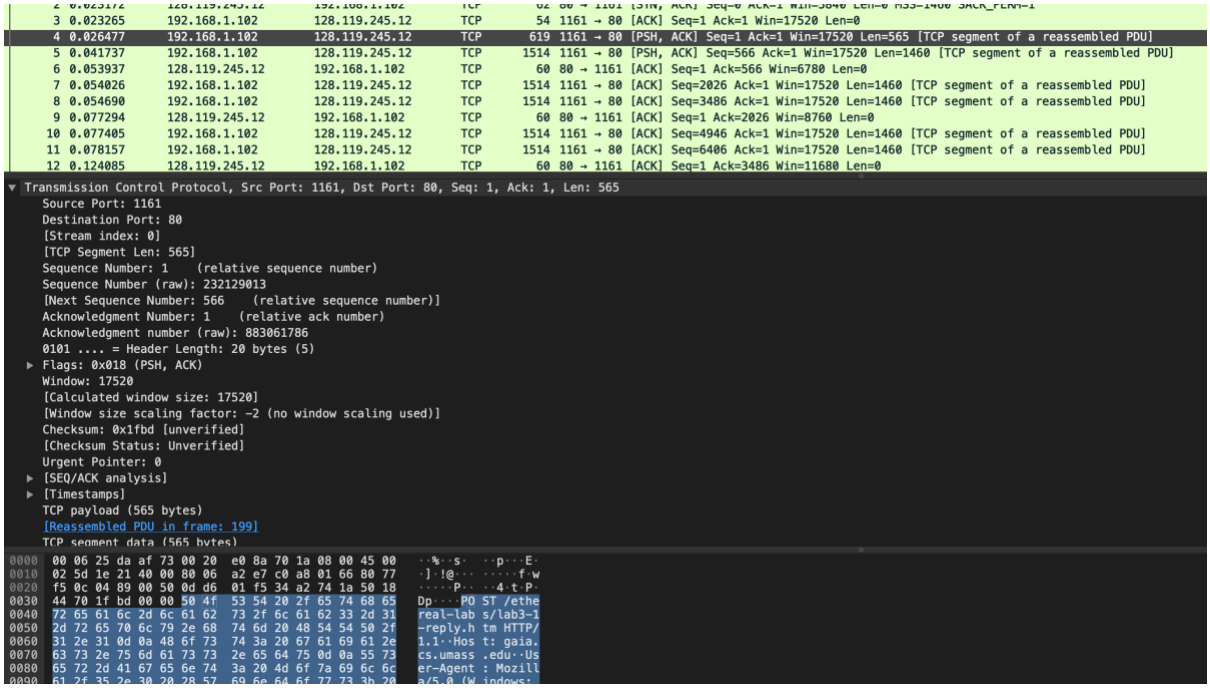
Question 1. What is the IP address of gaia.cs.umass.edu? On what port number is it sending and receiving TCP segments for this connection? What is the IP address and TCP port number used by the client computer (source) that is transferring the file to gaia.cs.umass.edu?
Answer:





As the screen shortcut shown above, the IP address of gaia.cs.umass.edu is 128.119.245.12 and the port is 80. The IP address of client is 192.168.1.102 and port is 1161.

Question 2. 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.
Answer:
The segment 4 contains the HTTP POST command.
The sequence number is 232129013.



Question 3. 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) sent from the client to the web server (Do not consider the ACKs received from the server as part of these six segments)? 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 relevant parts of Section 3.5 or lecture slides) after the receipt of 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.

Answer:

The first 6 segments sent from the client to web server are 4 5 7 8 10 11.

Segment	Sequence num	Sent time(s)	ACK	RTT
1	232129013	0.026477000	0.053937000	0.02746
2	232129578	0.041737000	0.077294000	0.035557
3	232131038	0.054026000	0.124085000	0.070059
4	232132498	0.054690000	0.169118000	0.114428
5	232133958	0.077405000	0.217299000	0.139894
6	232135418	0.078157000	0.267802000	0.189645

Estimate RTT = (1 - α) * Estimate RTT + α * sample RTT

= 0.875 * Estimate RTT + 0.125 * sample RTT

Segment 1 = 0.02746000.

Segment 2 = 0.857 * 0.02746000 + 0.125 * 0.035557 = 0.02847.

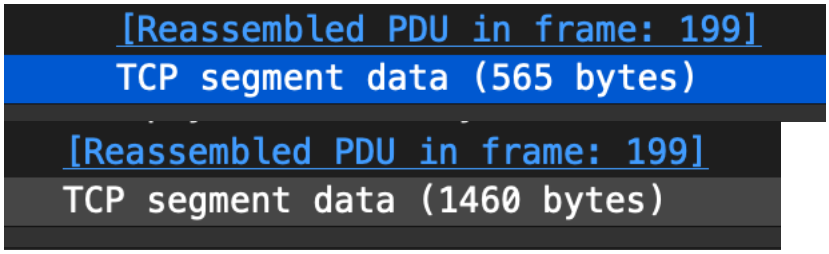
Segment 3 = 0.857 * 02847 + 0.125 * 0.070059 = 0.03367.

Segment 4 = 0.857 * 0.03367 + 0.125 * 0.114428 = 0.04376.

Segment 5 = 0.857 * 0.04376 + 0.125 * 0.139894 = 0.05578.

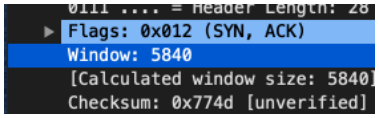
Segment 6 = 0.875 * 0.05578 + 0.125 * 0.189645 = 0.07251.

Question 4. What is the length of each of the first six TCP segments? (same six segments as Q3)
Answer:



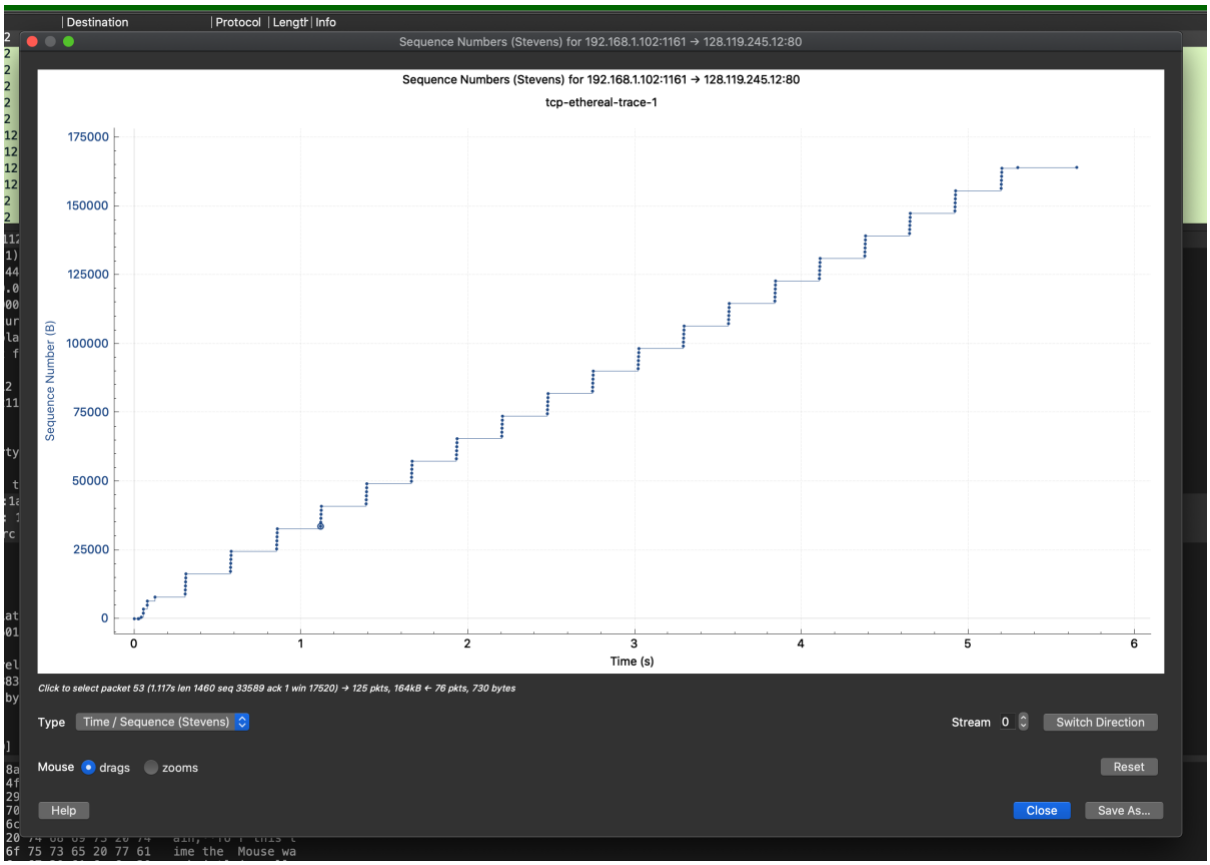
The first one is 565 but others are 1460.

Question 5. 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?



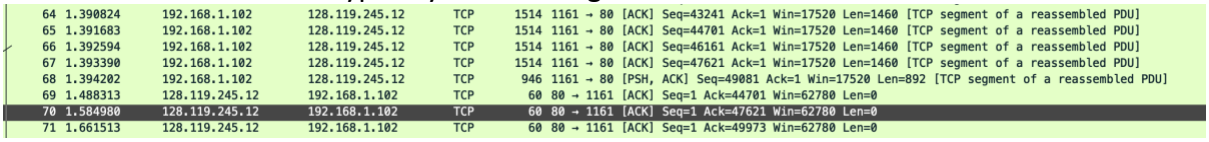
The minimum buffer space is 5840.
No, it doesn't.

Question 6. Are there any retransmitted segments in the trace file? What did you check for (in the trace) in order to answer this question?
Answer:



No, there aren't any retransmitted segments in the trace file because if it does, then there will be some segments that the sequence number of them is smaller than their neighbour's sequence number somewhere in the graph.

Question 7. 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).
Answer:
1460 should the receiver typically acknowledge in an ACK.



64	1.390824	192.168.1.102	128.119.245.12	TCP	1514 1161 → 80 [ACK] Seq=43241 Ack=1 Win=17520 Len=1460 [TCP segment of a reassembled PDU]
65	1.391683	192.168.1.102	128.119.245.12	TCP	1514 1161 → 80 [ACK] Seq=44701 Ack=1 Win=17520 Len=1460 [TCP segment of a reassembled PDU]
66	1.392594	192.168.1.102	128.119.245.12	TCP	1514 1161 → 80 [ACK] Seq=46161 Ack=1 Win=17520 Len=1460 [TCP segment of a reassembled PDU]
67	1.393390	192.168.1.102	128.119.245.12	TCP	1514 1161 → 80 [ACK] Seq=47621 Ack=1 Win=17520 Len=1460 [TCP segment of a reassembled PDU]
68	1.394202	192.168.1.102	128.119.245.12	TCP	946 1161 → 80 [PSH, ACK] Seq=49081 Ack=1 Win=17520 Len=892 [TCP segment of a reassembled PDU]
69	1.488313	128.119.245.12	192.168.1.102	TCP	60 80 → 1161 [ACK] Seq=1 Ack=44701 Win=62780 Len=0
70	1.584980	128.119.245.12	192.168.1.102	TCP	60 80 → 1161 [ACK] Seq=1 Ack=47621 Win=62780 Len=0
71	1.661513	128.119.245.12	192.168.1.102	TCP	60 80 → 1161 [ACK] Seq=1 Ack=49973 Win=62780 Len=0

Like in segment 70, the ACK 47621 is ACKing two segments with seq=46161 and seq=47621. This is due to the TCP use delayed ACK and cumulative ack for received segments.

Question 8. What is the throughput (bytes transferred per unit time) for the TCP connection? Explain how you calculated this value.
Answer:
Total data transmitted is the difference between the ACK of last segment and seq of first which is
164091 – 1 = 164090.
Transmission time is the difference between the last ACK and first which is
5.455830000 - 0.026477000 = 5.429353.
Throughput = Total data transmitted / Transmission time = 30.233 Kbyte/s

Exercise 2 Answer:

Question 1 . What is the sequence number of the TCP SYN segment that is used to initiate the TCP connection between the client computer and server?
Answer:
2818463618

Question 2. 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?
Answer:
Seq = 1247095790
Ack = 2818463619
The value is increase by one from client seq number.

Question 3 . 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?
Answer:
Seq = 2818463619
Ack = 1247095791

No, it doesn't.

Question 4 . Who has done the active close? 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?
Answer:
Client and server both have done the active close because the seq number in No 304 is equal to ACK number in No 305, it didn't increase by one.

Question 5 . 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?
For client to server:
There are total 2818463653 – 2818463618 – 2(which are FIN and SYN) = 33 Bytes.
For server to client:
There are total 1247095832 – 1247095790 – 2(which are FIN and SYN) = 40 Bytes.

Since the initial sequence number is sent, the both sides will expect the next segment.
Since the final ACK is received, the both sides will stop transmit data.