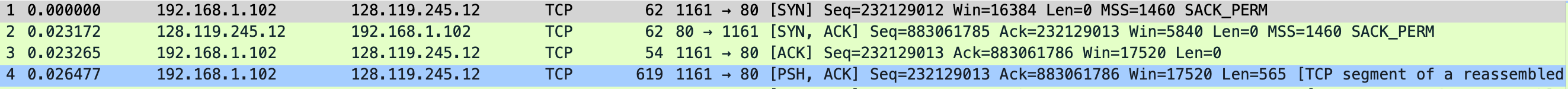
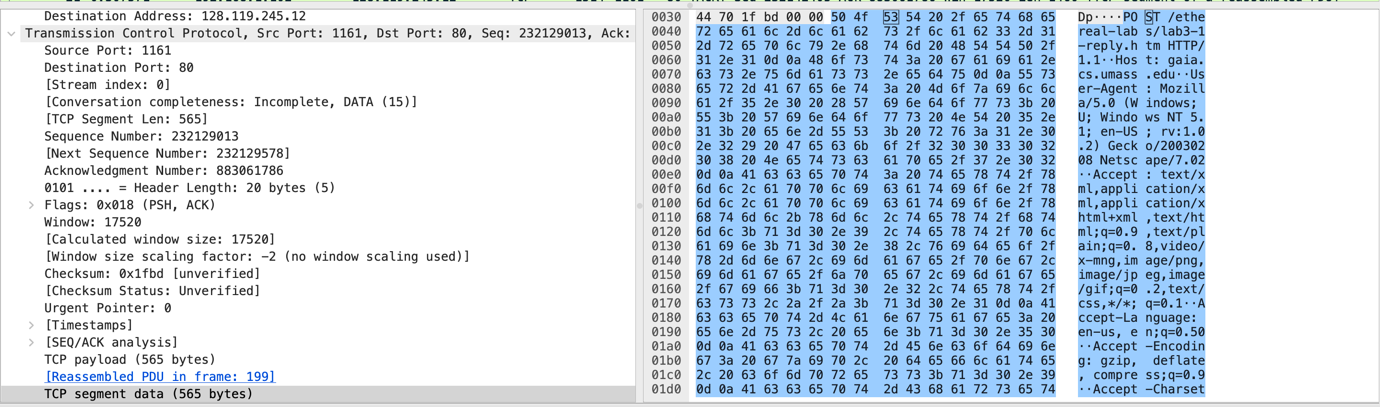
**Exercise 1**

**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 are the IP address and TCP port numbers used by the client computer (source) that is transferring the file to gaia.cs.umass.edu?**



The IP address of gaia.cs.umass.edu is 128.119.245.12. It is sending and receiving on the port 80. The IP address of the source is 192.168.1.102 and the port number is 1161.

**Question 2. What is the sequence number of the TCP segment containing the HTTP POST command?**

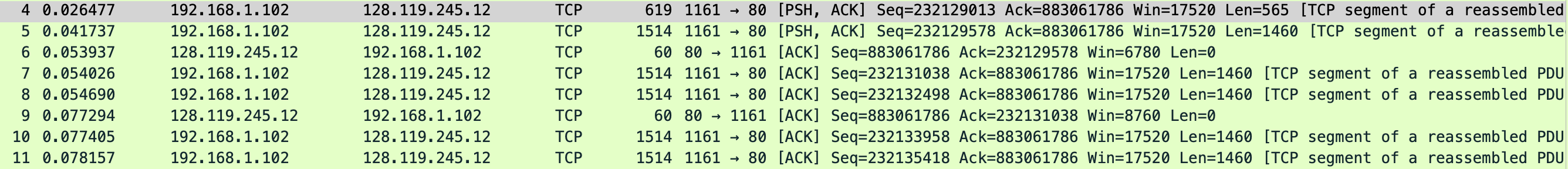
The sequence number is 232129013.

**Question 3. Consider the TCP segment containing the HTTP POST as the first segment in the TCP connection.**

**(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.**

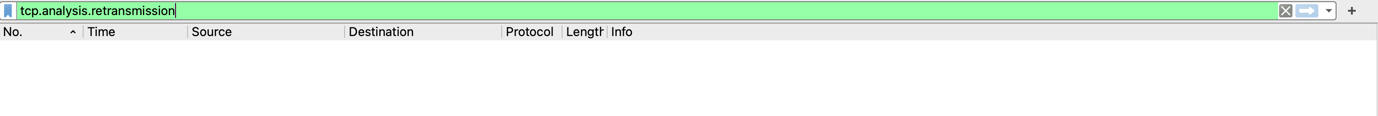
**(d) What is the length of each of the first six TCP segments?**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Segment No. | Sequence No. | Time | ACK Received | RTT | Estimated RTT | Length |
| 1(4) | 232129013 | 0.026477 | 0.053937 | 0.053937 – 0.026477 = 0.02746 | 0.02746 | 565 |
| 2(5) | 232129578 | 0.041737 | 0.077294 | 0.077294 – 0.041737 = 0.035557 | (1-0.125)\*0.02746 + 0.125\*0.035557 = 0.02487 | 1460 |
| 3(7) | 232131038 | 0.054026 | 0.124085 | 0.124085 – 0.054026  = 0.070059 | (1-0.125)\*0.02487 + 0.125\*0.070059 = 0.03367 | 1460 |
| 4(8) | 232132498 | 0.054690 | 0.169118 | 0.169118 – 0.054690 = 0.114428 | (1-0.125)\* 0.03367 + 0.125\*0.114428= 0.04376 | 1460 |
| 5(10) | 232133958 | 0.077405 | 0.217299 | 0.217299 – 0.077405 = 0.139894 | (1-0.125)\* 0.04376+ 0.125\*0.139894= 0.05578 | 1460 |
| 6(11) | 232135418 | 0.078157 | 0.267802 | 0.267802 – 0.078157  = 0.189645 | (1-0.125)\* 0.05578+ 0.125\*0.189645= 0.07251 | 1460 |

**Question 4. 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 amount of buffer space is 5840(line 2). The lack of receiver buffer space does not throttle the sender as the window grows larger to a maximum of 62780 bytes.

**Question 5. Are there any retransmitted segments in the trace file? To answer this question, what did you check for (in the trace)?**

****There are not any retransmitted files in the tracefile. This was checked by using ‘tcp.analysis.retransmission’ in the filter field. No segments appeared, therefore there was no retransmissions

**Question 6. 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)?**

The receiver typically acknowledges 1460 bytes in an ACK.

**Question 7. What is the TCP connection's throughput (bytes transferred per unit of time during the connection)? Explain how you calculated this value.**

Looking at frame 199 we can see that all the reassembled TCP segments is 164090 bytes. This can also be shown by the last ACK – first Seq no. 232293103 – 232129013 = 164090 bytes.

Total transmission time = Last ACK time – first Seq time = 5.455830 – 0.026477 = 5.429353

Therefore, throughput = 164090/5.429353 = 30222.75398 bytes/sec.

**Exercise 2**

**A screenshot of a computer

Description automatically generated**

**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?**

The sequence number is 28148463618.

**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?**

The sequence number is 1247095790. The ACK value is 2818463619. This ACK value is determined by the sequence number from the client + 1.

**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?**

The sequence number is 2818463619. The value of the ACK is 1247095791. This segment does not contain any data.

**Question 4*.* Who has done the active close? Is it the client or the server? How have you determined this? What type of closure has been performed? 3 Segment (FIN/FINACK/ACK), 4 Segment (FIN/ACK/FIN/ACK) or Simultaneous close?**

Both the client and the server have done active close. This can be determined as the client and server have both sent FINACK to each other. The type of closure used is simultaneous close. The sequence number of 304 is equal to the ACK of 305.

**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?**

Data bytes from the client to the server are 2818463653 – 28148463618 – 2(SYN,FIN) = 33bytes.

Data bytes from the server to the client are 1247095832 – 1247095790 – 2(SYN,FIN) = 40 bytes.

The relationship between initial sequence number and the final ACK is the amount of bytes sent.