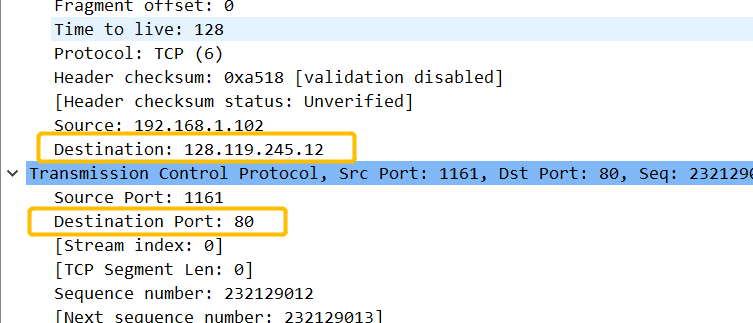
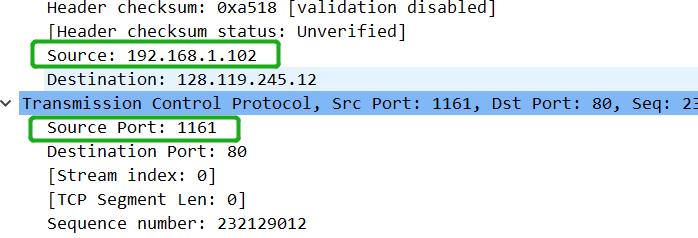
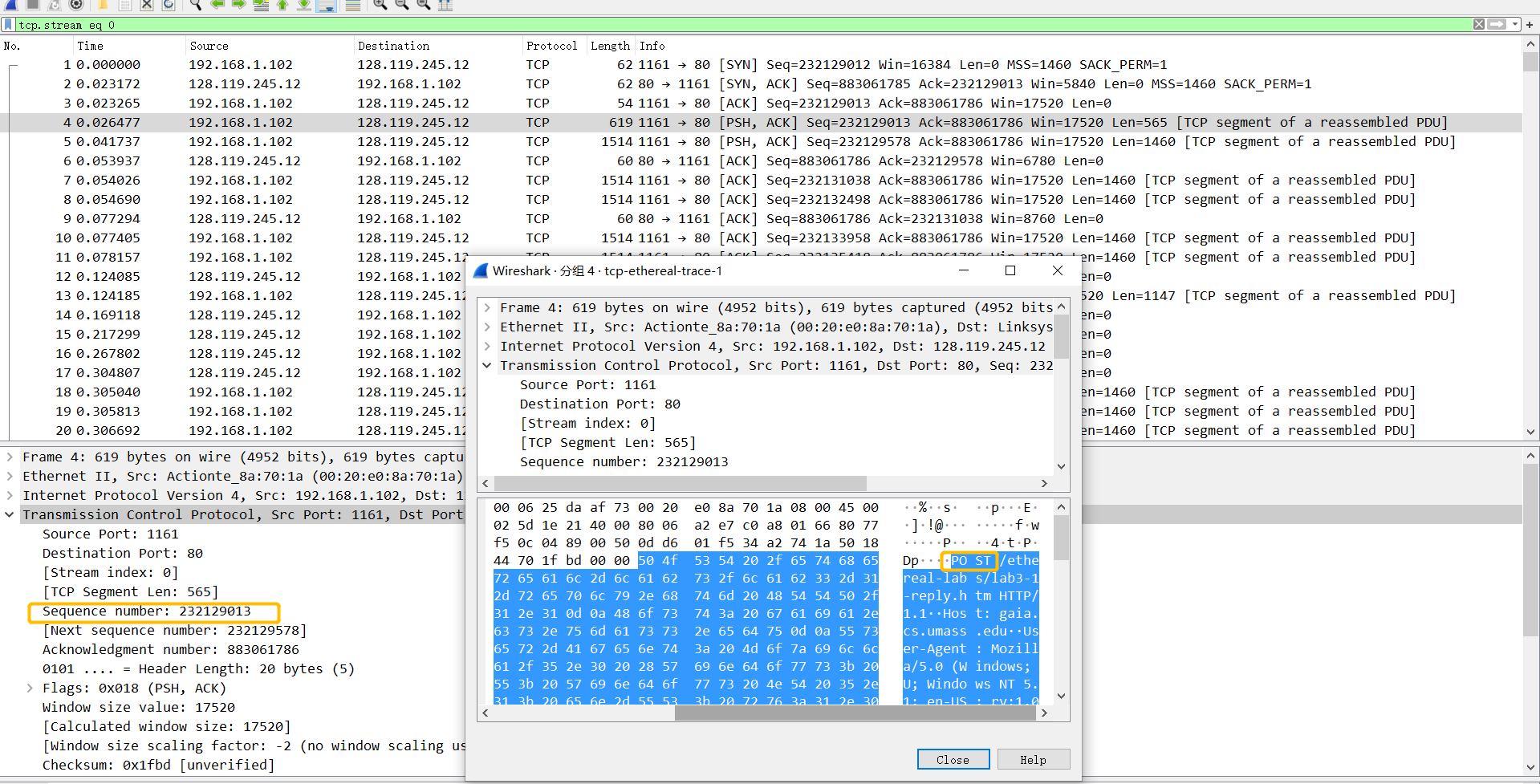
Exercise 1

*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 is the IP address and TCP port number 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 and port number is 80 for this connection.

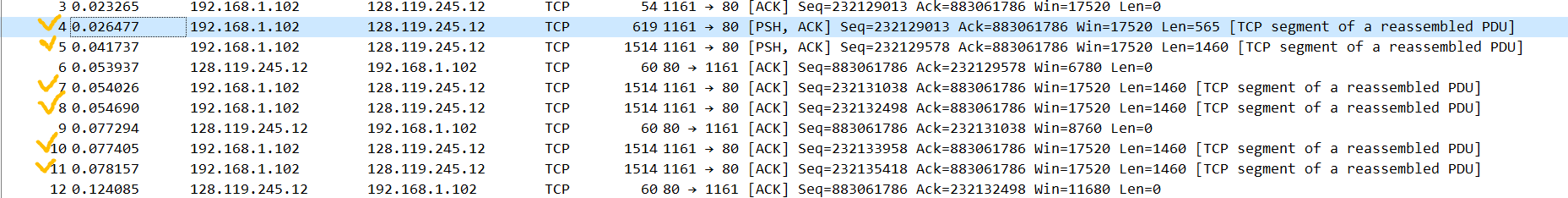


The IP address and TCP port number used by the client computer (source) is 192.168.1.102 and 1161.

*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.

The sequence number of the TCP segment containing the HTTP POST   
Command 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.

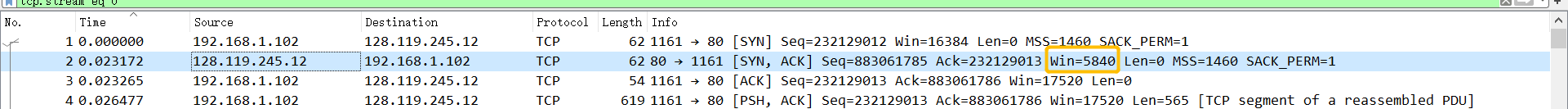
There are 4th ,5th ,7th ,8th ,10th and 11th .

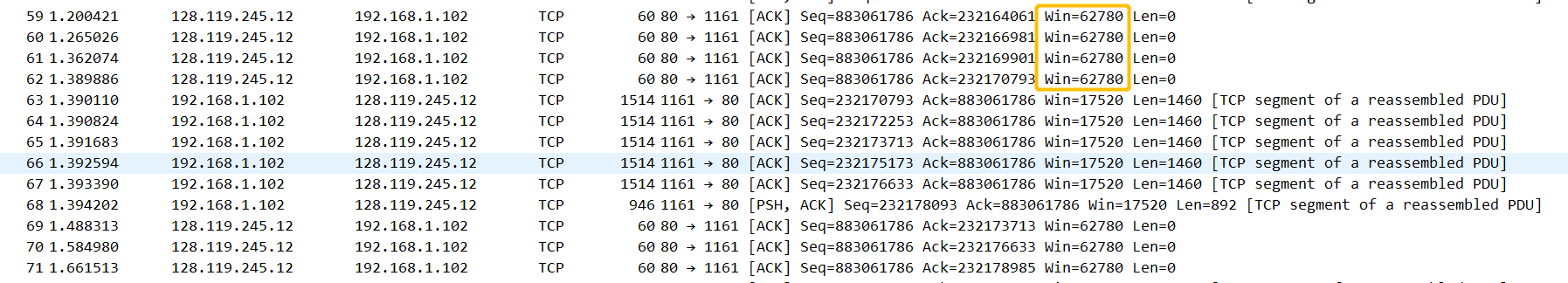
|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Sequence number | Sent time(s) | Time of Acks received(s) | RTT(s) | EstimatedRTT(s) |
| 232129013 | 0.026477 | 0.53937 | 0.027460 | 0.027460000 |
| 232129578 | 0.041737 | 0.77294 | 0.035557 | 0.028472125 |
| 232131038 | 0.054026 | 0.124085 | 0.070059 | 0.03367048438 |
| 232132498 | 0.054690 | 0.169118 | 0.114428 | 0.04376517383 |
| 232133958 | 0.077405 | 0.217299 | 0.139894 | 0.0557812771 |
| 232135418 | 0.078157 | 0.267802 | 0.189645 | 0.07251424246 |

*Question 4.*What is the length of each of the first six TCP segments?

|  |  |
| --- | --- |
| Sequence number | Length(bytes) |
| 232129013 | 565 |
| 232129578 | 1460 |
| 232131038 | 1460 |
| 232132498 | 1460 |
| 232133958 | 1460 |
| 232135418 | 1460 |

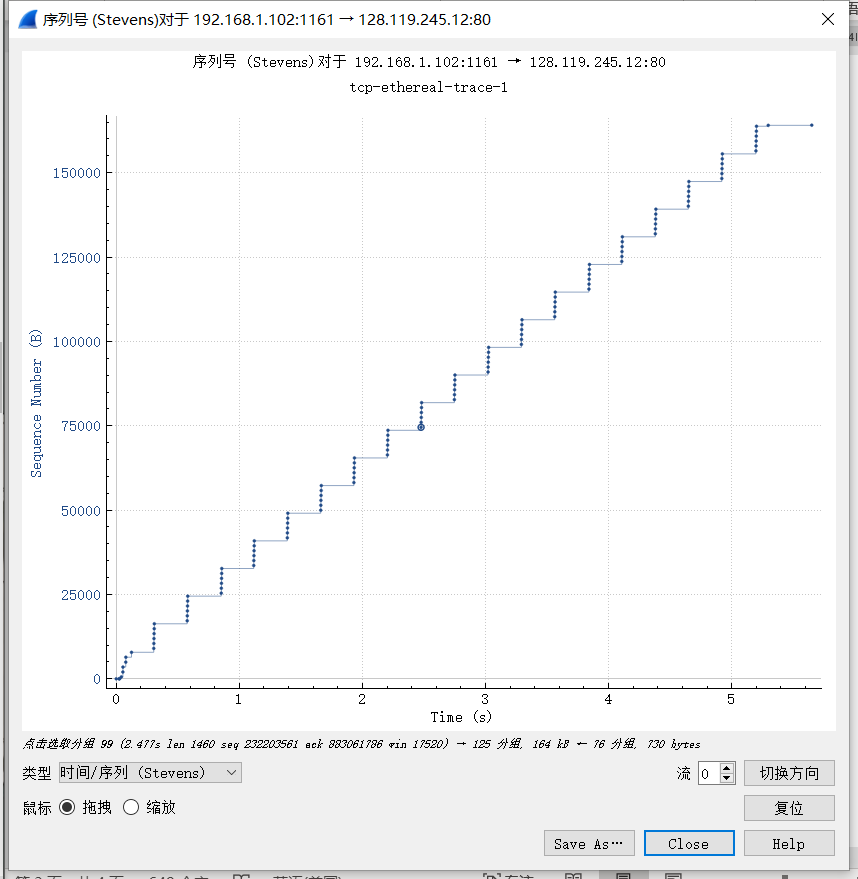
*Question 5.*What is the minimum amount of available buffer space advertised at the receiver for the entire trace? Does the lack of receiver buffer space ever throttle the sender?





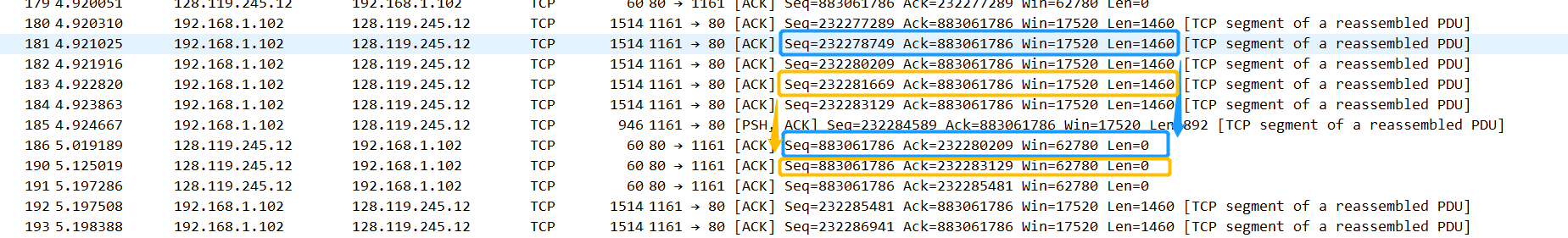
The minimum amount of available buffer space advertised at the receiver for the entire trace is 5840 bytes and the maximum of that buffer space is 62780 bytes, so it is not likely to throttle the sender, because the buffer space is always bigger that the segment size.

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?



According to this graph, we can see that there is no same sequence number with different time stamps, which means that no retransmit segments.

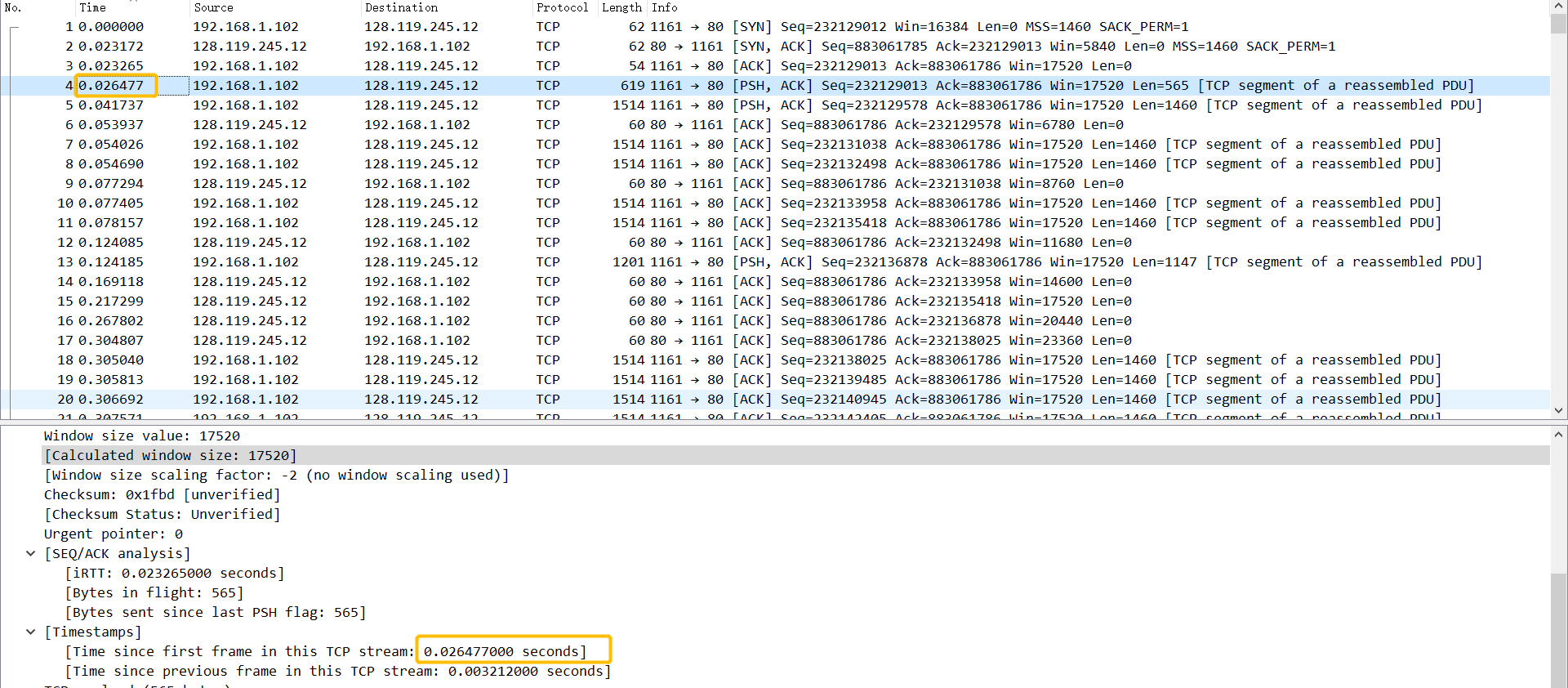
*Question 7.*How much data does the receiver typically acknowledge in an ACK? Can you identify cases where the receiver is ACKing every other received segment (recall the discussion about delayed acks from the lecture notes or Section 3.5 of the text).

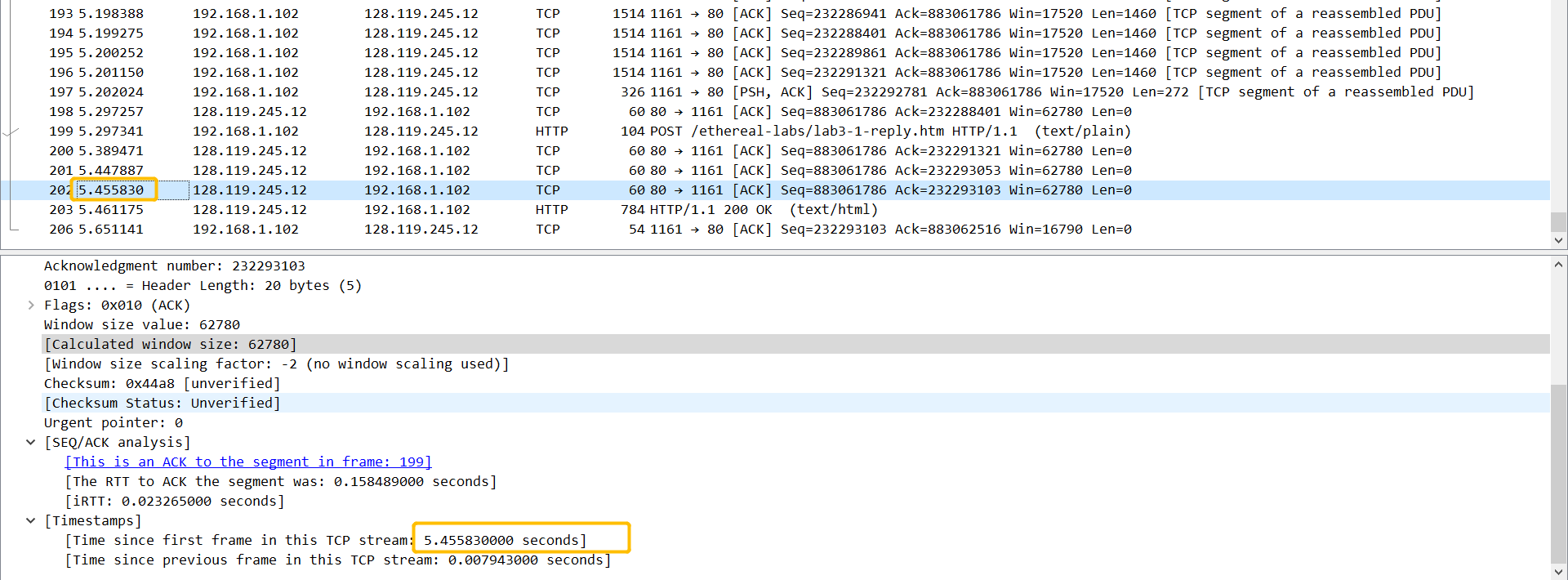


The data that the receiver typically acknowledge in an ACK is 1460 bytes.

I can identify cases where the receiver is ACKing every other received segment, for example, as the above picture, 181st is acknowledged by 186th and 183rd is acknowledged by 190th

*Question 8.*What is the throughput (bytes transferred per unit time) for the TCP connection? Explain how you calculated this value.



Due to 203rd shows OK, from 4th to 202nd, we can get:

hence,

therefore, the throughput for the TCP connection is computed as:

Exercise 2

*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 of the TCP SYN segment is 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?

The sequence number of the SYNACK segment is 1247095790 and is the value of the Acknowledgement field in the SYNACK segment is 2818463619. the server determine that value by adding 1 to the last sequence numbers(initial serial 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?

The sequence number of the ACK segment sent by the client computer in response to the SYNACK is 2818463619 and the value of the Acknowledgment field in this ACK segment is 1247095791.

Yes, this segment contain data because 301st ACK is 2818463652 and 298th Seq=2818463619, therefore it contains 2818463652-2818463619= 33 bytes data.

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?

Client and server both did the active close. I determined this by 304th and 305th, which both client and server generate the FIN segment before receiving it and the type of closure has been performed is Simultaneous close.

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?

The data bytes have been transferred from the client to the server:

The data bytes have been transferred from the server to the client:

The ACK number can track the length of the data being transferred, so the final ACK obtained from the other side minus the initial sequence number is the size of data that has been transmitted.