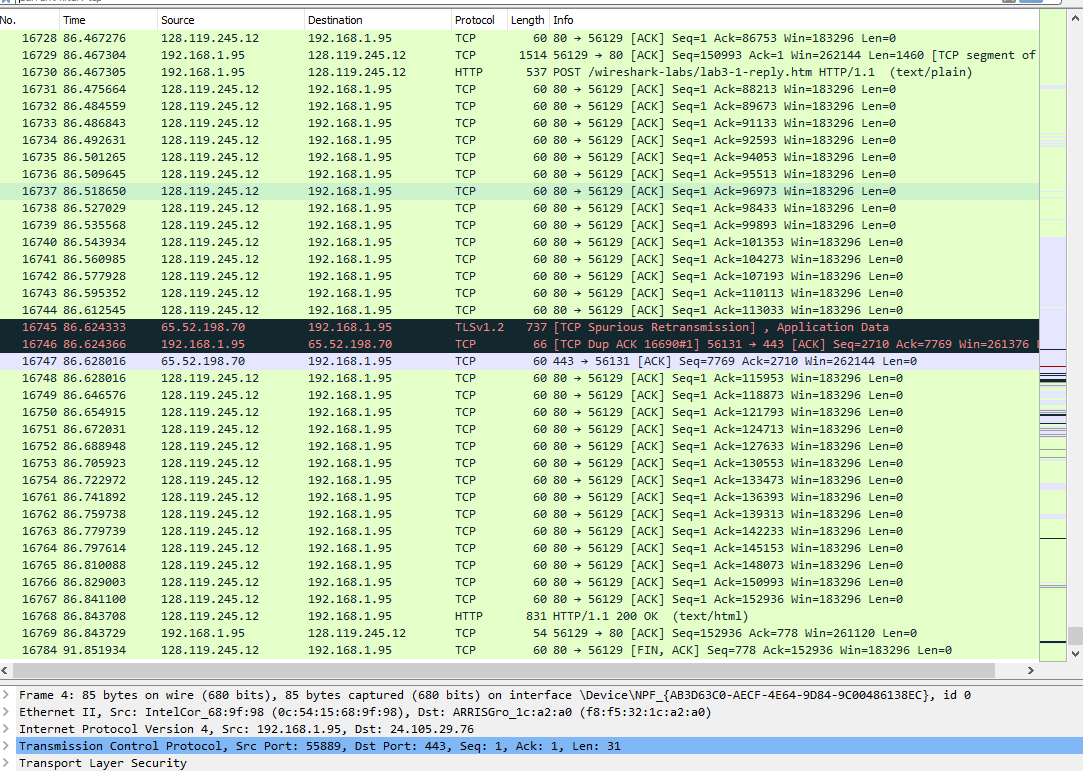
Wireshark Lab #3

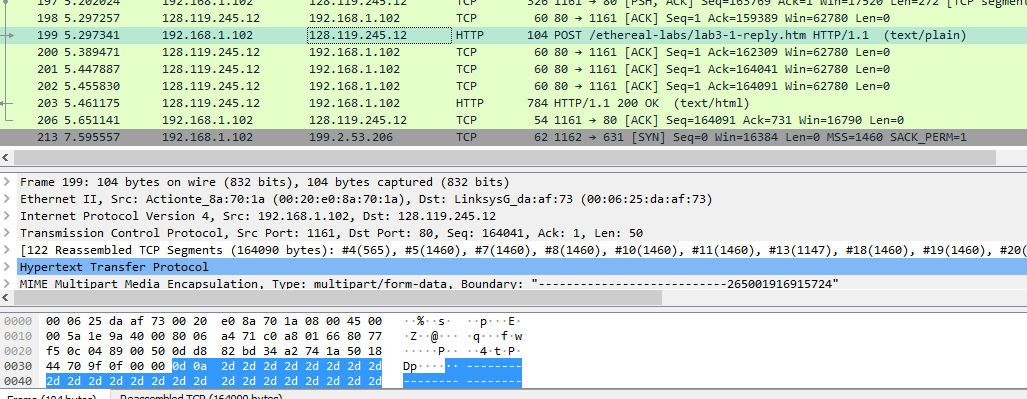
1. What is the IP address and TCP port number used by the client computer (source) that is transferring the file to gaia.cs.umass.edu? To answer this question, it’s probably easiest to select an HTTP message and explore the details of the TCP packet used to carry this HTTP message, using the “details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

The IP address and TCP port number used by the client computer (source) that is transferring the file to gaia.cs.umass.edu is 192.168.1.102 and 1161



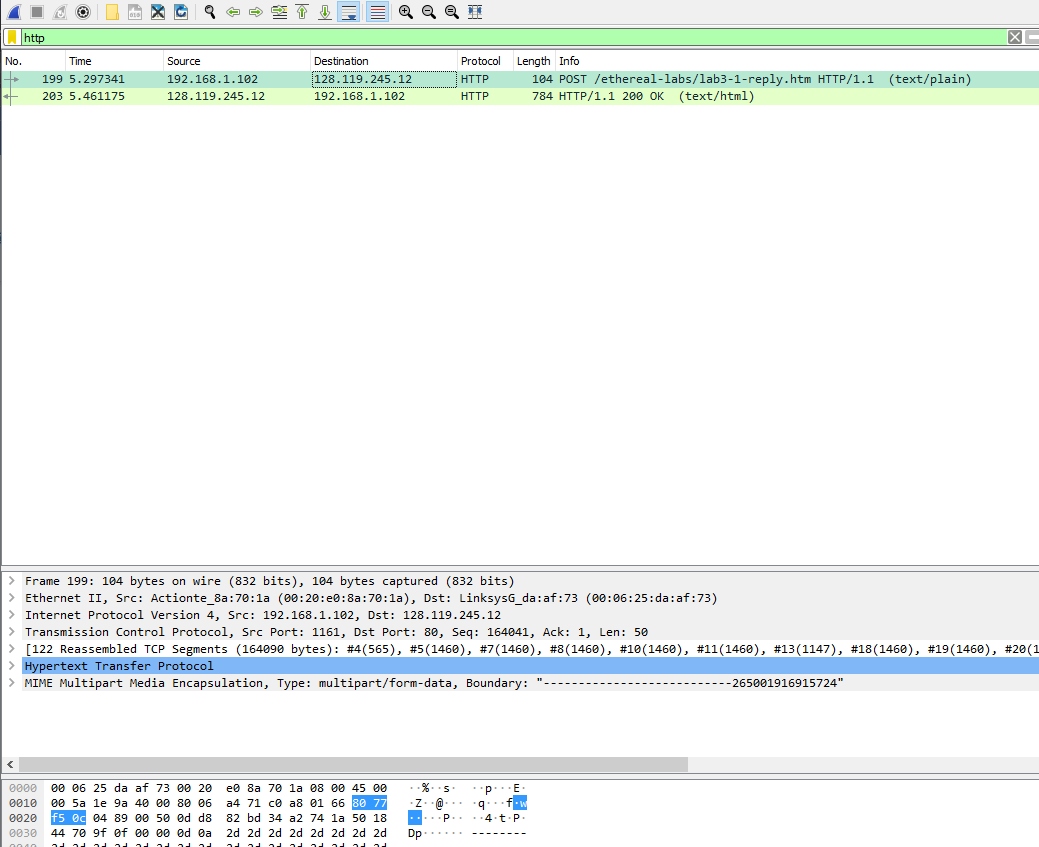
2. What is the IP address of gaia.cs.umass.edu? On what port number is it sending and receiving TCP segments for this connection?

The IP address of gaia.cs.umass,edu is 128.119.245.12 and the port number that it is sending and receiving TCP segments for this connection is 80.



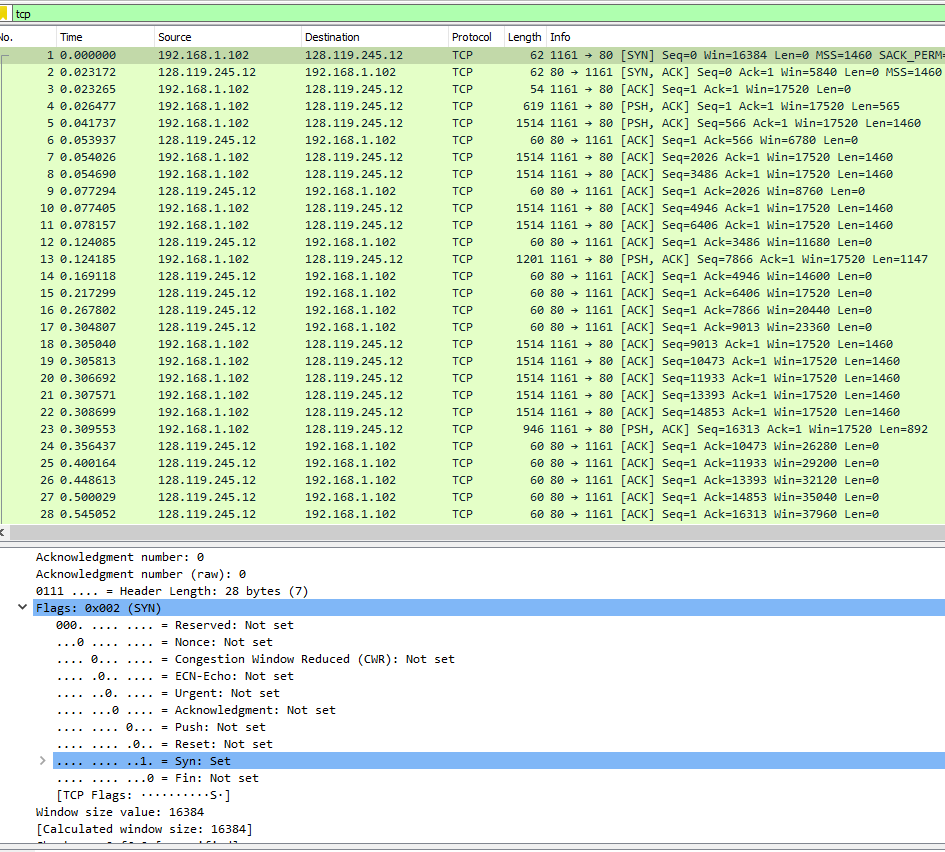
3. What is the IP address and TCP port number used by your client computer (source) to transfer the file to gaia.cs.umass.edu?

The IP address and TCP port number used by my client computer to transfer the file to gaia.cs.umass.edu is 192.168.1.102 and 1161.



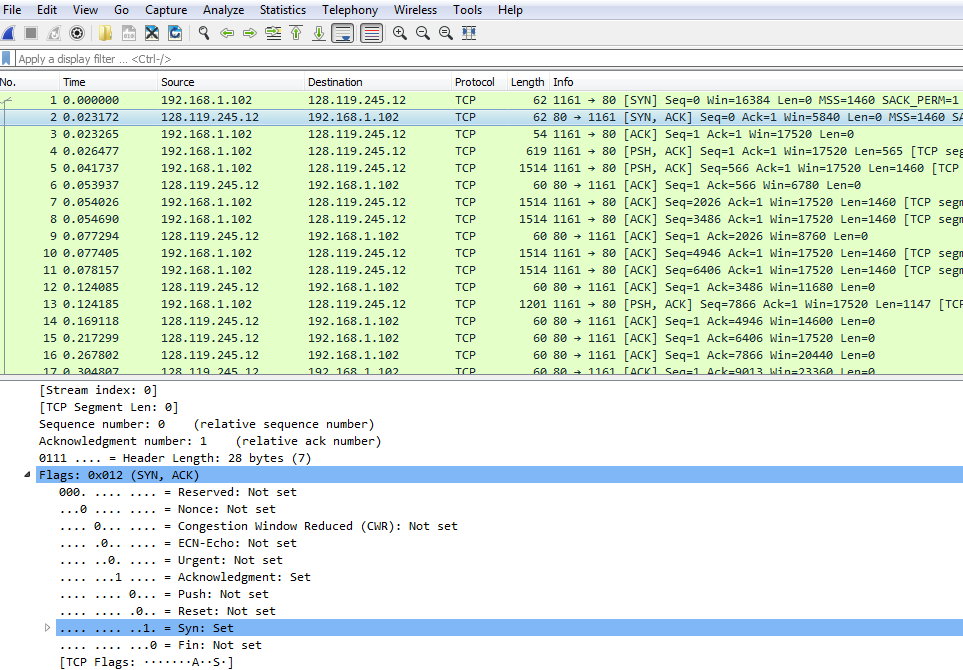
4. What is the sequence number of the TCP SYN segment that is used to initiate the TCP connection between the client computer and gaia.cs.umass.edu? What is it in the segment that identifies the segment as a SYN segment?

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 is 0. The SYN flag identifies the segment as a SYN segment in the segment. The flag is set to 1.



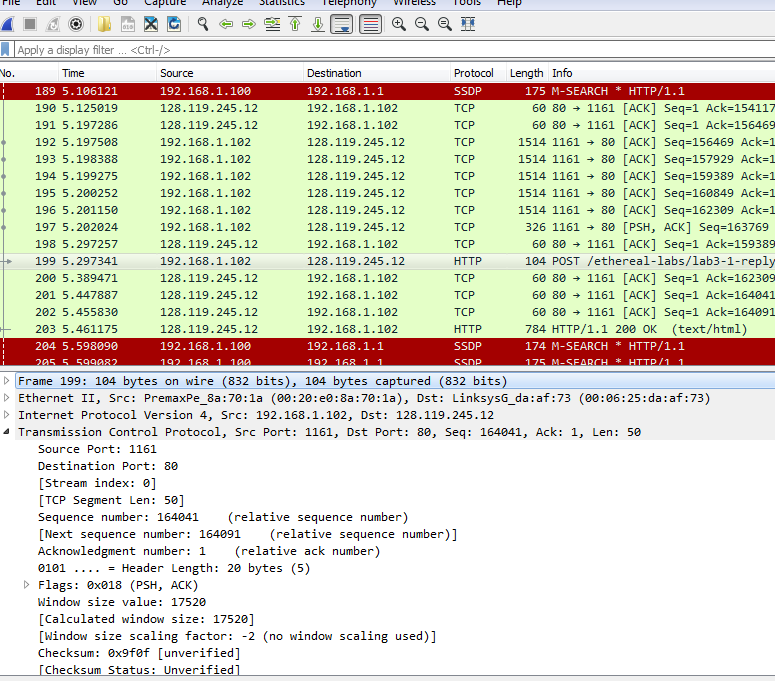
5. What is the sequence number of the SYNACK segment sent by gaia.cs.umass.edu to the client computer in reply to the SYN? What is the value of the Acknowledgement field in the SYNACK segment? How did gaia.cs.umass.edu determine that value? What is it in the segment that identifies the segment as a SYNACK segment?

The sequence number of the SYNACK segment sent by gaia.cs.umass.edu to the client computer in reply to the SYN is 0. The value of the acknowledgement field in the SYNACK segment is 1. Gaia.cs.umass.edu determine that value by adding 1 to the initial sequence number of SYN segment from the client computer. The SYN and acknowledgement flag in the segment, which is set to 1, indicate that this segment is a SYNACK segment.

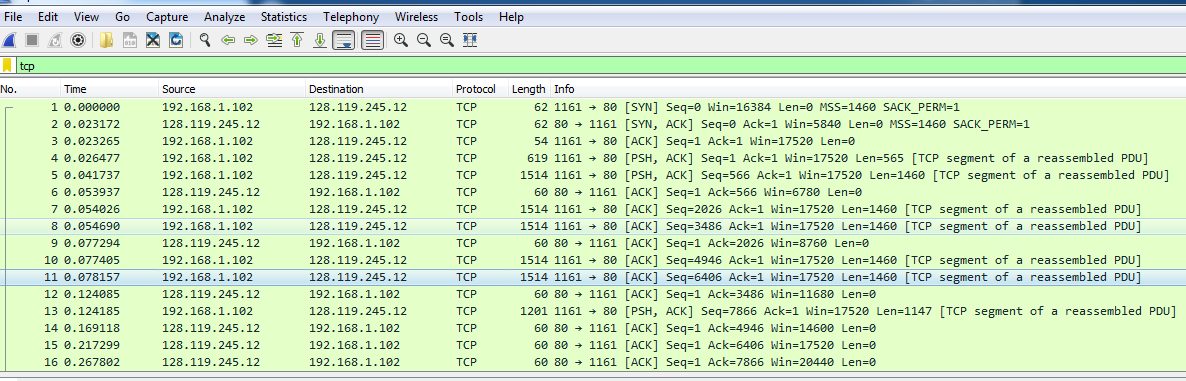


6. What is the sequence number of the TCP segment containing the HTTP POST command? Note that in order to find the POST command, you’ll need to dig into the packet content field at the bottom of the Wireshark window, looking for a segment with a “POST” within its DATA field.

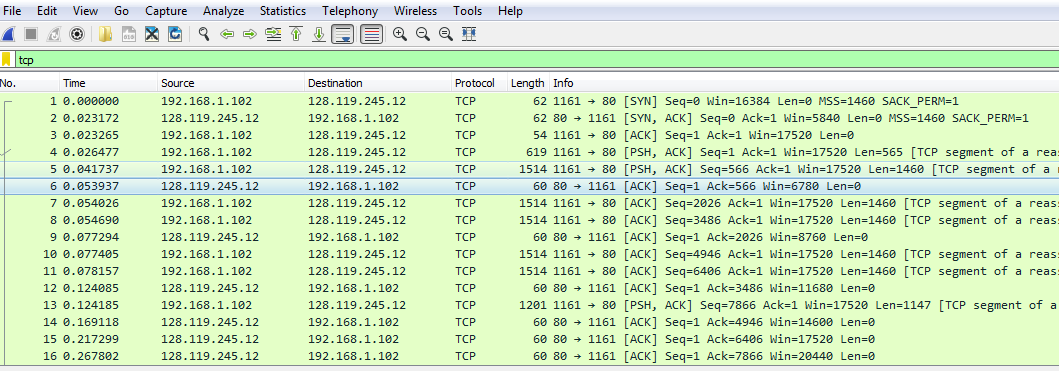
The sequence number of the TCP segment containing the HTTP POST command is 164041.

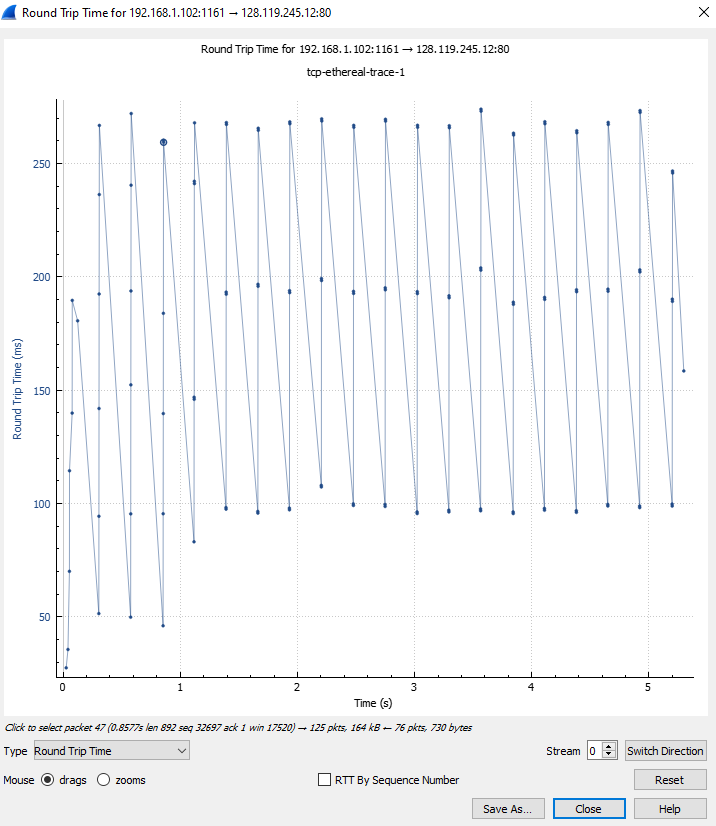


7. Consider the TCP segment containing the HTTP POST as the first segment in the TCP connection. What are the sequence numbers of the first six segments in the TCP connection (including the segment containing the HTTP POST)? At what time was each segment sent? When was the ACK for each segment received? 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 242 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 242 for all subsequent segments. Note: Wireshark has a nice feature that allows you to plot the RTT for each of the TCP segments sent. Select a TCP segment in the “listing of captured packets” window that is being sent from the client to the gaia.cs.umass.edu server. Then select: Statistics->TCP Stream Graph- >Round Trip Time Graph.



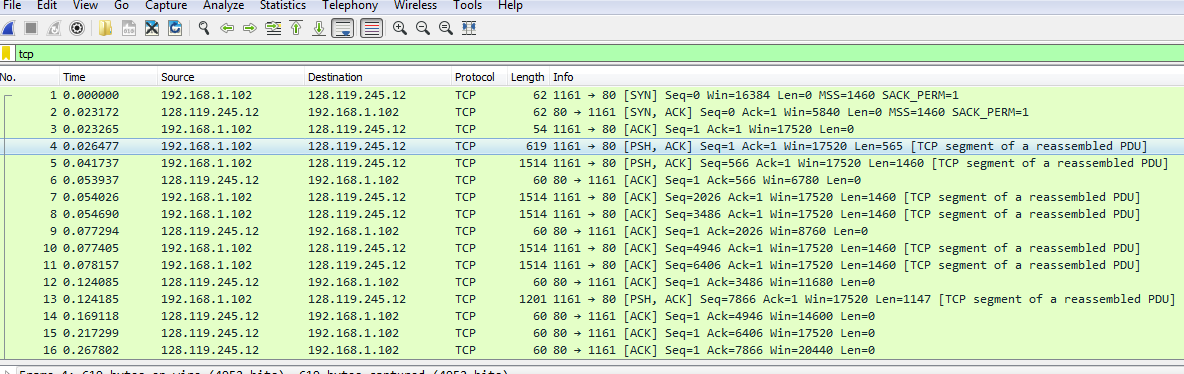
|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Number | Sequence number | Sent time | ACK received time | RTT | EstimatedRTT  (ACK Receive Time – Time Sent) |
| 4 | 1 | .026477 | .053937 | .02746 | .02746 |
| 5 | 566 | .041737 | .077294 | .035557 | .0285 |
| 7 | 2026 | .054026 | .124085 | .070059 | .0337 |
| 8 | 3486 | .054690 | .169118 | .11443 | .0438 |
| 10 | 4946 | .077294 | .217299 | .13989 | .0558 |
| 11 | 6406 | .078157 | .267802 | .18964 | .0725 |
|  |  |  |  |  |  |





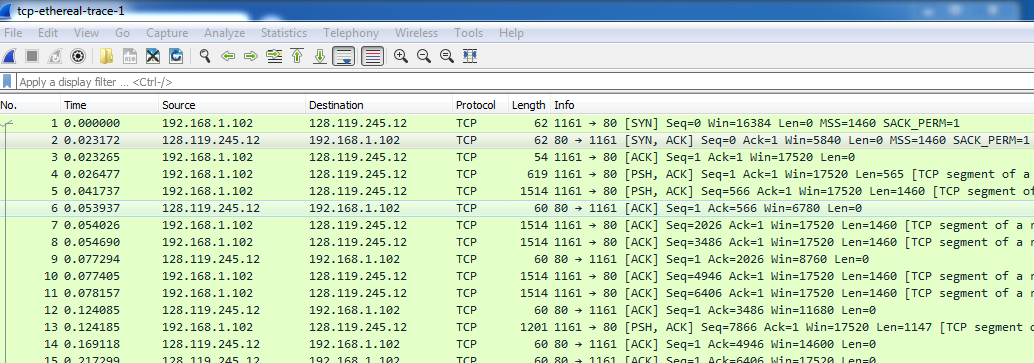
8. What is the length of each of the first six TCP segments?

The length of each of the first six TCP segments are 565 and 1460 for the next five TCP segments.



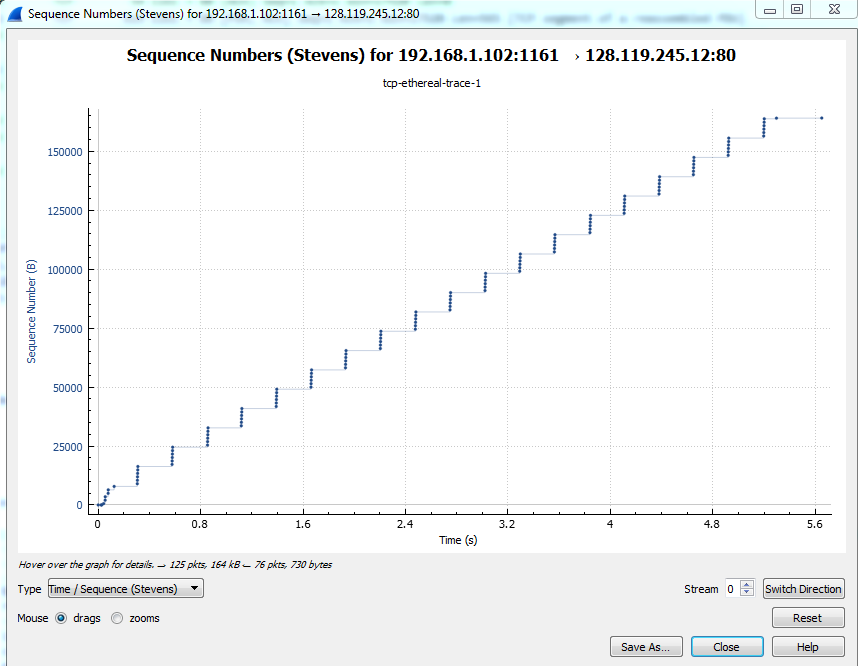
9. What is the minimum amount of available buffer space advertised at the received for the entire trace? Does the lack of receiver buffer space ever throttle the sender?

The minimum amount of available buffer space advertised at the received for the entire trace is 5480 bytes. The lack of receiver buffer space is never throttled due to lacking of receiver buffer space by inspecting this trace and the number increases over time.



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

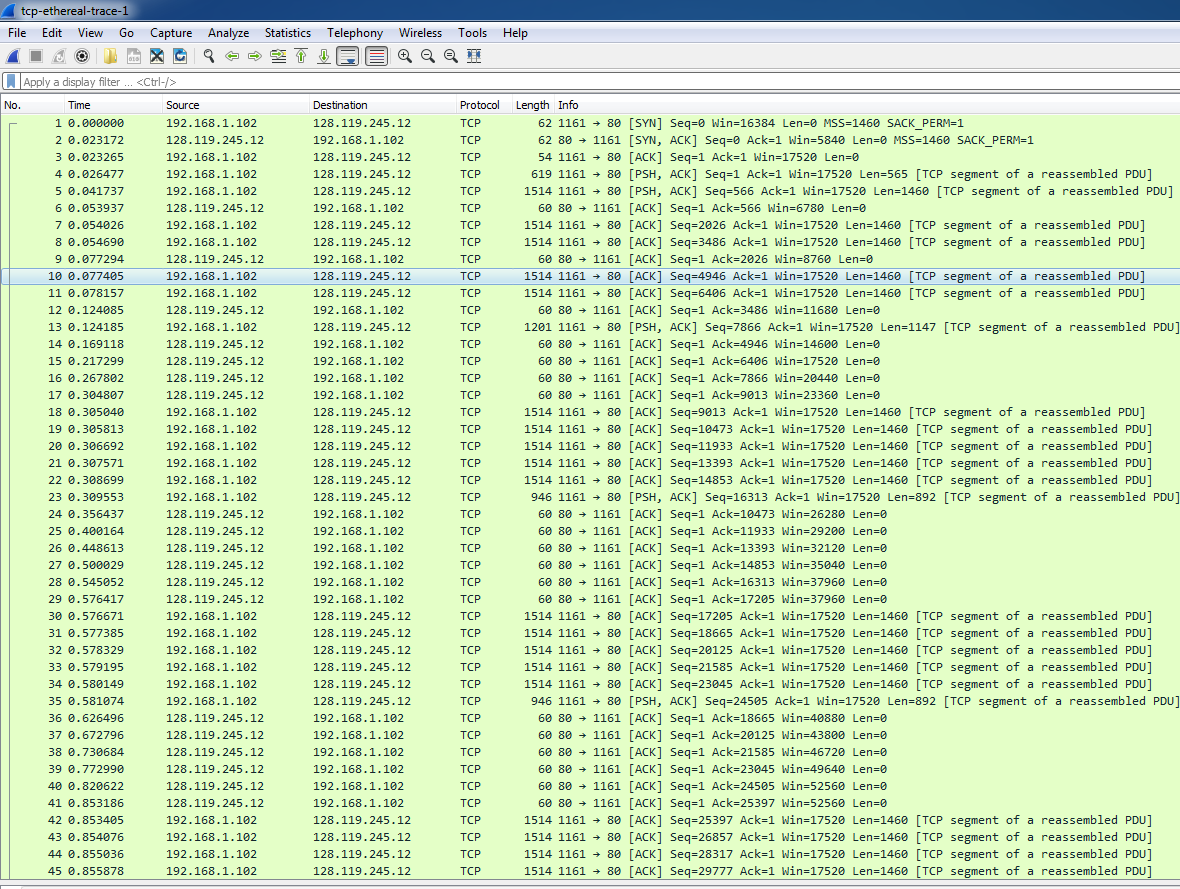
There are no retransmitted segments in the trace file. In order to answer this question I checked the sequence numbers of the TCP segments in the trace file. All sequence numbers from the source to the destination are increasing monotonically with respect to time. If there is a retransmitted segment, the sequence number of this retransmitted segment should be smaller than those of its neighboring segments.



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 250 in the text).

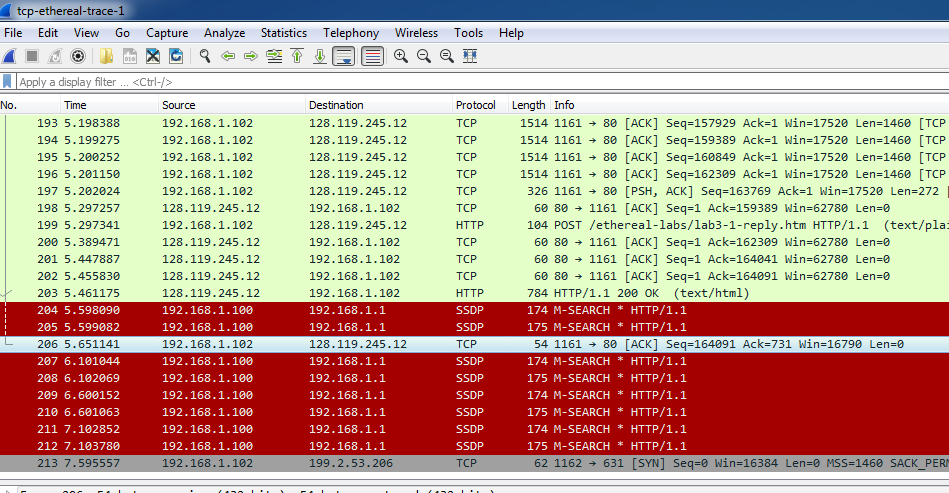
The receiver is typically acking 432 bits of data in an ACK. There are cases where the receiver acks every other segment. This is shown when more than one ack occurs in a row.

|  |  |  |
| --- | --- | --- |
| ACK Number | ACK Sequence Number | Acknowledged Data |
| 1 | 566 | 566 |
| 2 | 2026 | 1460 |
| 3 | 3486 | 1460 |
| 4 | 4946 | 1460 |
| 5 | 6406 | 1460 |
| 6 | 7866 | 1460 |
| 7 | 9013 | 1147 |
| 8 | 10473 | 1460 |
| 9 | 11933 | 1460 |
| 10 | 13393 | 1460 |
| 11 | 14853 | 1460 |
| 12 | 16313 | 1460 |



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

164090/5.4294 = 30.222 KByte/sec



13. Use the *Time-Sequence-Graph(Stevens*) plotting tool to view the sequence

number versus time plot of segments being sent from the client to the

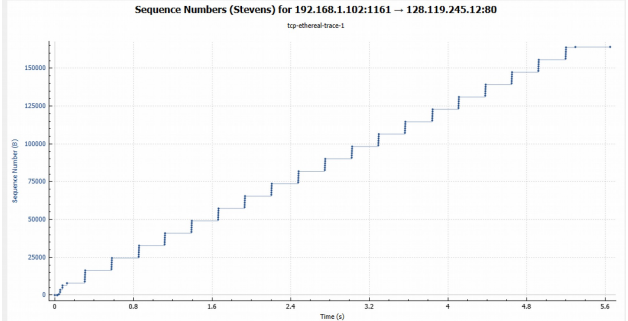
gaia.cs.umass.edu server. Can you identify where TCP’s slowstart phase begins

and ends, and where congestion avoidance takes over? Comment on ways in

which the measured data differs from the idealized behavior of TCP that we’ve

studied in the text.

Slow start begins at the beginning of the connection. Congestion avoidance never takes over, as the sender never overwhelms the receiver. In the idealized behavior of TCP, it is assumed that at a certain point, the sender will overwhelm the receiver. However, in the measured data, this never happens.



14. Answer each of two questions above for the trace that you have gathered when

you transferred a file from your computer to gaia.cs.umass.edu.

Slow start begins at the beginning of the connection. Congestion avoidance never takes over, as the sender never overwhelms the receiver. In the idealized behavior of TCP, it is assumed that at a certain point, the sender will overwhelm the receiver. However, in the measured data, this never happens

