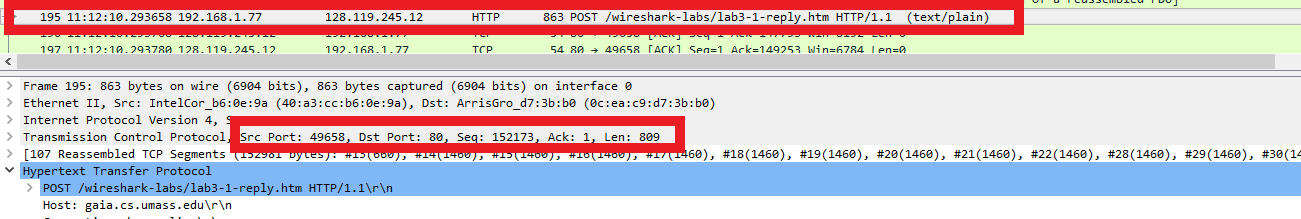
2d Lt James Marvin

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 client IP address is 192.168.1.77. You can see this listed in the source IP field for the top box around the HTTP POST packets. The TCP port number is 49658. You can see this in the TCP description (bottom box) of the post packet.**

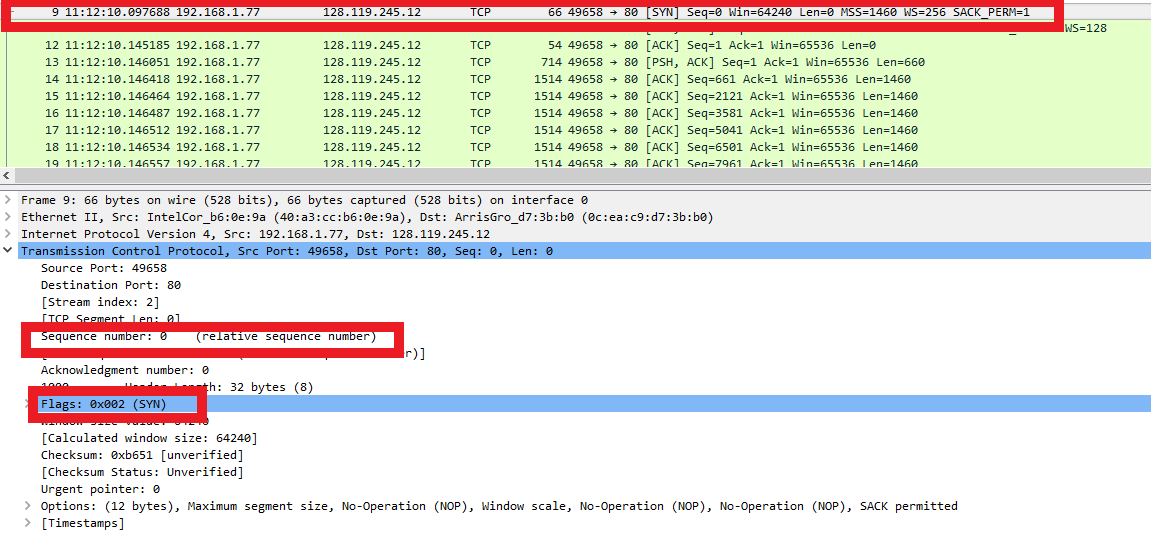
1. What is the IP address of gaia.cs.umass.edu? On what port number is it sending and receiving TCP segments for this connection?

**The IP address of gaia.cs.umass.edu is 128.119.245.12. You can see this listed in the destination IP field for the top box in the image used in the solution for problem 1. The port number it is send and receiving TCP segments is port 80. You can see this in the “Dst port:” filed in the bottom box of the image used in problem 1.**

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

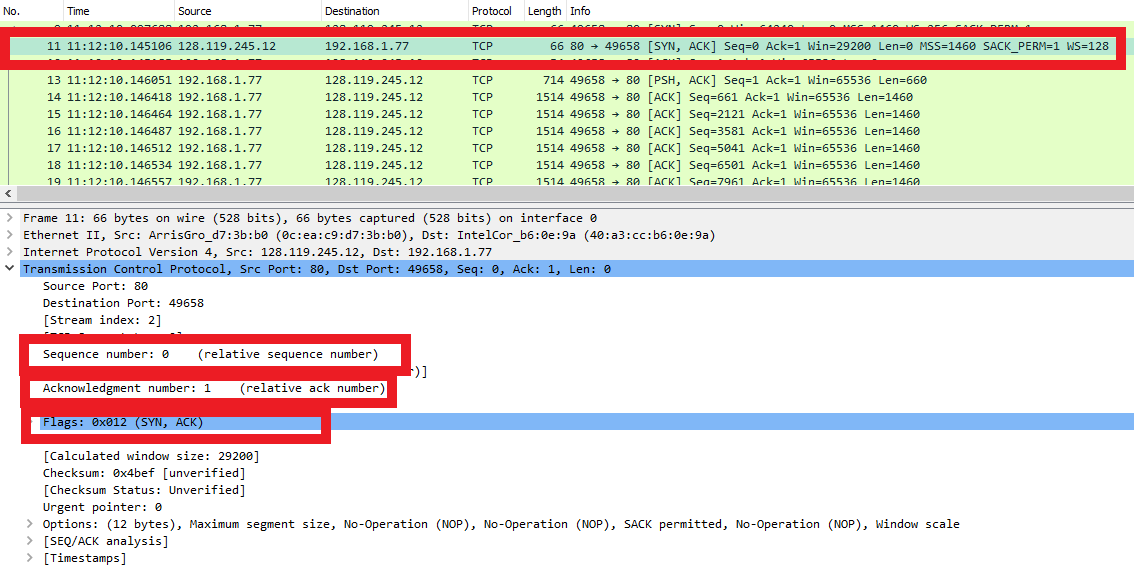
**The IP address is 192.168.1.77 and the TCP port number is 49658. This is the same question as number 1 and the picture for number 1 provides the same justification.**

1. 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?



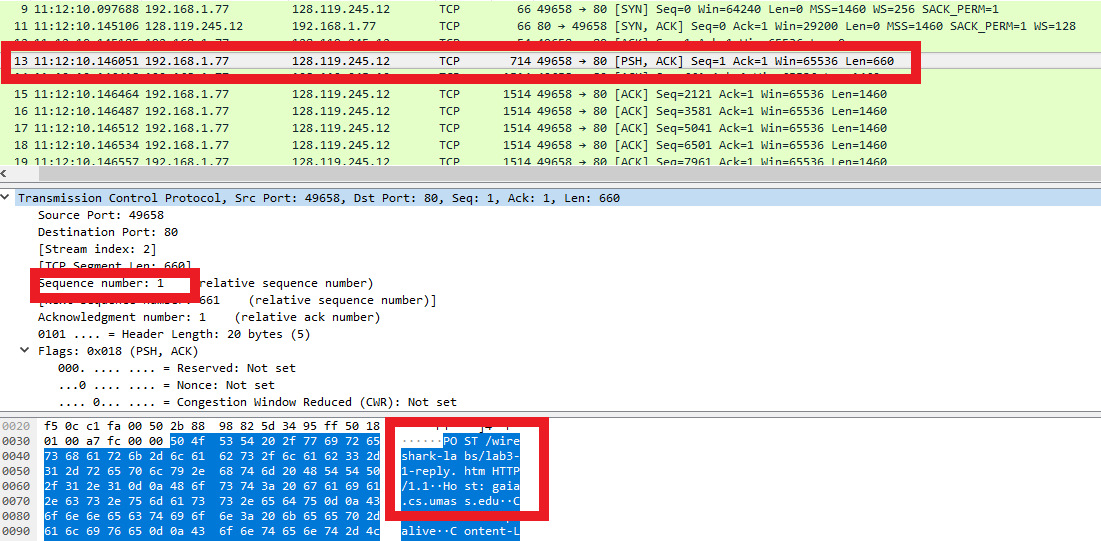
**The sequence number that is used to initiate the TCP connection between the client computer and gaia.cs.umass.edu is 0. You can find that information in the second box in the picture above. The SYN flag (0x002) in the packet indicates that it is a SYN packet. You can find this information within the third/bottom box in the picture above.**

1. 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 is 0 and is seen in the second box from the top. The value of the acknowledgement field is 1 and is seen in the third box from the top. The value was determined by the number in the byte stream it expects from the computer in the next packet. Since no bytes of data were sent in the SYN packet from the previous problem it expects to receive the first byte in the next packet. The part of the segment that indicates that this is a SYNACK is the Flags: header seen in the bottom box.**

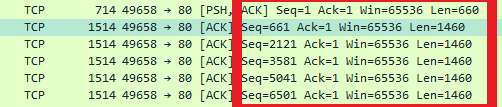
1. What is the sequence number of the TCP segment containing the HTTP POST command? Note that in order to find the POST command, you’ll need to dig into the packet content field at the bottom of the Wireshark window, looking for a segment with a “POST” within its DATA field.



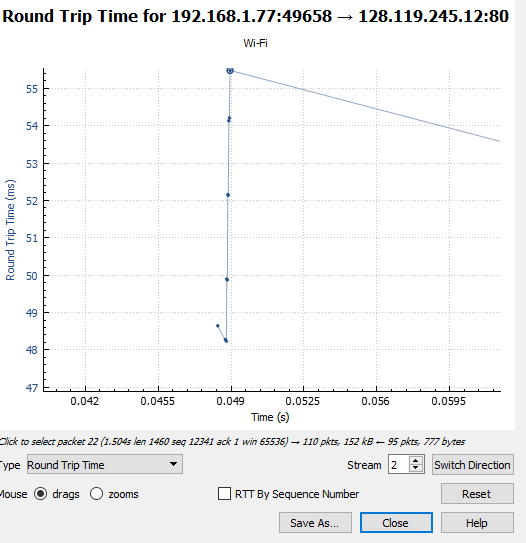
**The sequence number for the TCP segment containing the HTTP POST command was 1. I found the TCP packet by examining the data field (seen in bottom box containing POST command). Then I examine the Wireshark data in the middle box to determine the sequence number was 1.**

1. Combine your answers to questions 7 and 8 in a table similar to the following.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Sequence # | Time sent(seconds) | Time ACK rcd (seconds) | RTT  (seconds) | Est RTT (seconds) | Seg Length (bytes) |
| 1 | .04831 | .09691 | .0486 | .0486 | 660 |
| 661 | .048651 | .096951 | .0483 | .0485 | 1460 |
| 2121 | .048657 | .096857 | .0482 | .0484 | 1460 |
| 3581 | .048657 | .098557 | .0499 | .0485 | 1460 |
| 5041 | .048658 | .098458 | .0498 | .0486 | 1460 |
| 6501 | .04866 | .10076 | .0521 | .0490 | 1460 |



I used this information to gather the sequence numbers, and segment lengths for the packets.

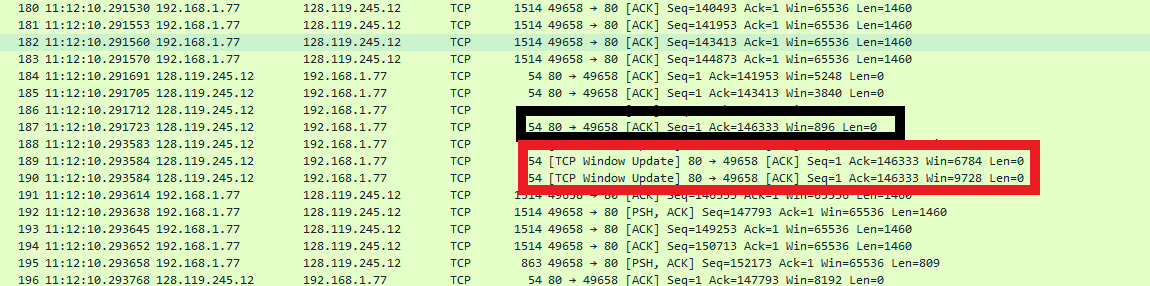


I used the information from this graph to get the RTTs, time sent, and time received.

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

**Refer to the chart in problem 7.**

1. 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 advertised widow size is 896 bytes. I did this by scrolling through and examining the window fields of the TCP packets (example in black box). You can see that it throttles the sender when it sends window update packets to the sender (seen in red box).**

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

**No, there were no retransmitted segments. I checked the sequence numbers and there were no repeats.**

1. 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 typically acknowledges 1460 bytes. This can be seen by the difference between acks in a few of the previous screenshots. The trace did not indicate that the receiver was acking every other received segment.**

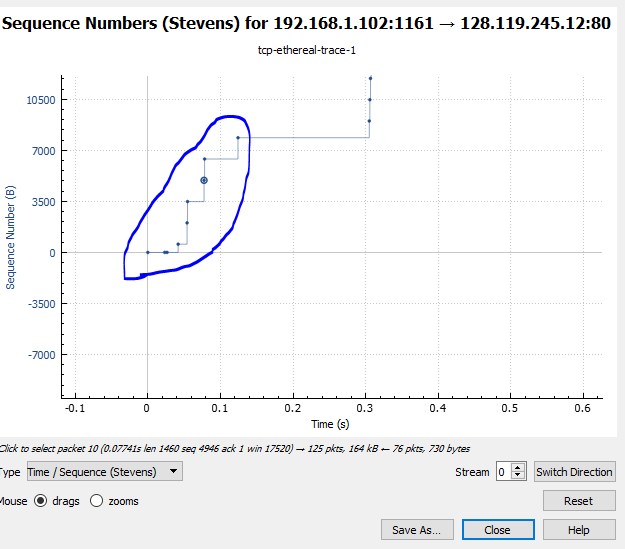
1. What is the throughput (bytes transferred per unit time) for the TCP connection? Explain how you calculated this value.



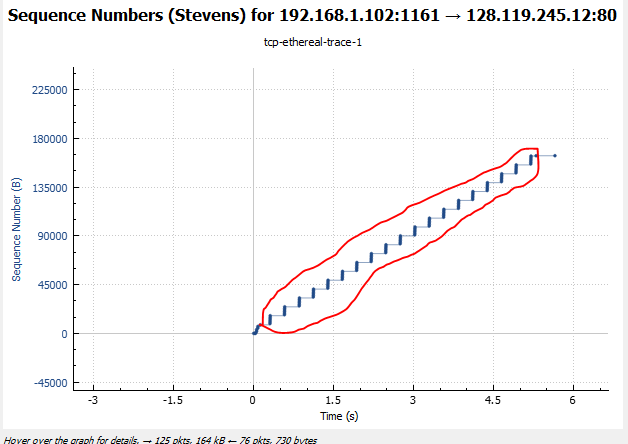


**You can see the number of bytes given in the sequence number of the second picture. And the time stamps give me the time from the first byte being sent to the last byte being received.**

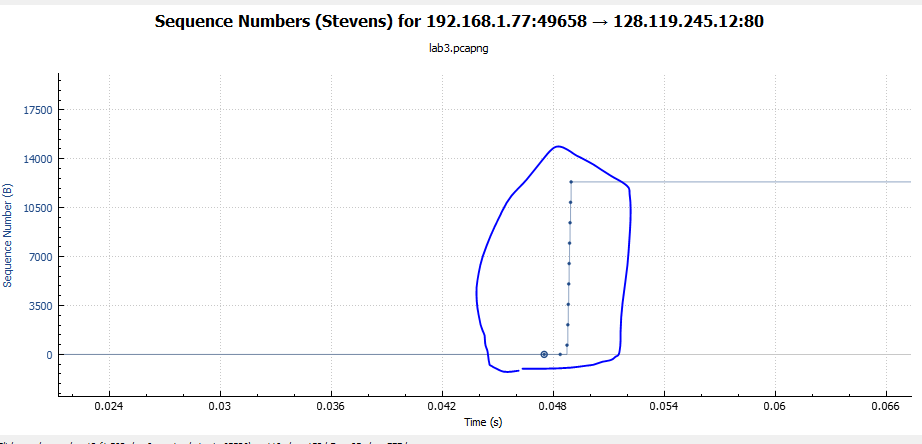
**Throughput= 152982 bytes/(.414596-.146051 seconds) = .56966 MBps**

1. 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.

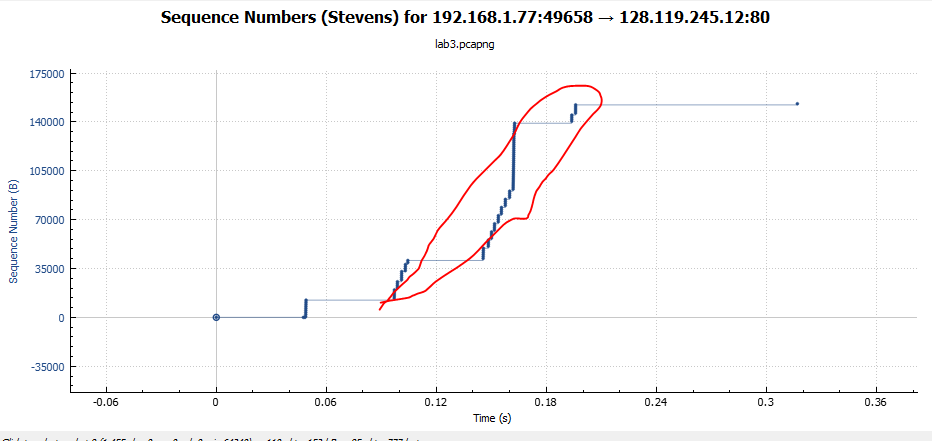
**The slow start phase is within the blue circle above.**



**In the red circle is where it enters congestion avoidance. It is different from the behavior we studied in class because the ranges between sets of sequence numbers were very consistent.**

1. 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

**This is where my results were entering slow start.**



**In the red circle is where my results were entering congestion avoidance. The ranges for sequence numbers before waiting on acks were not consistent.**