Joseph Lai
November 22, 2015

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?

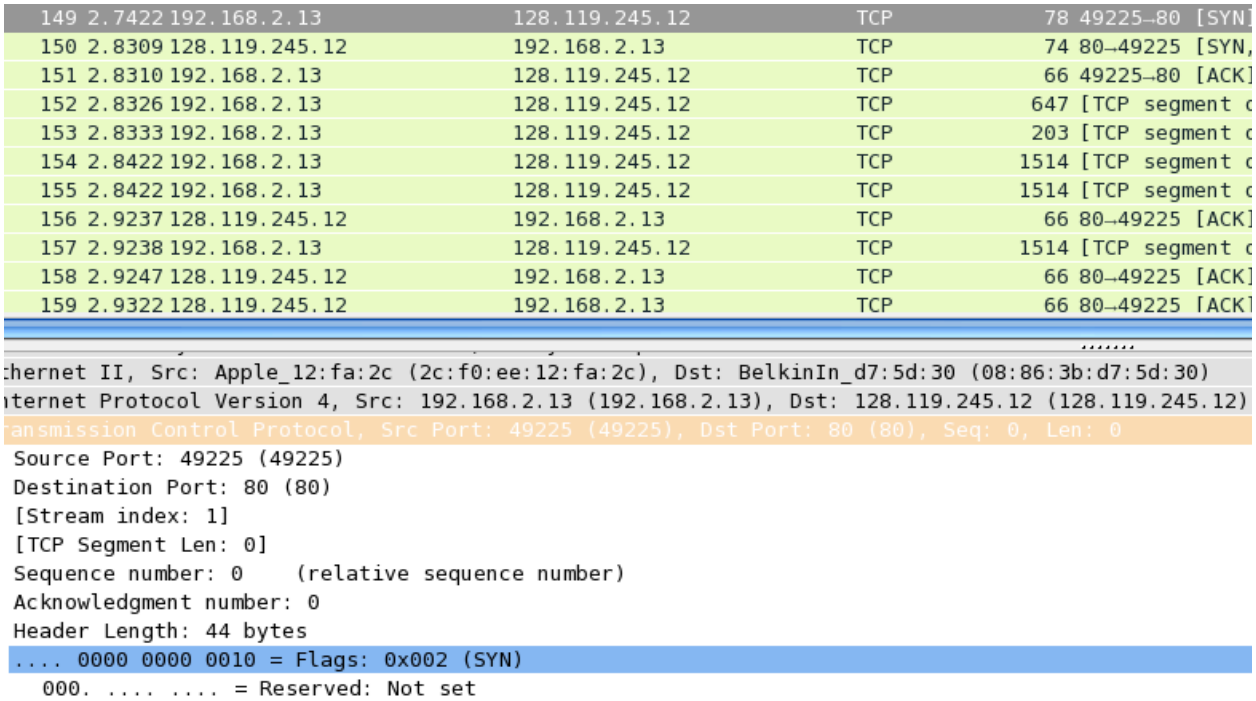
- IP and port of my client computer are 192.168.2.13 and 49203.

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?

- IP address of gaia.cs.umass.edu is 173.194.202.125 on port 5222.

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?

- IP and port of my client computer are still 192.168.2.13 and 49203.



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?

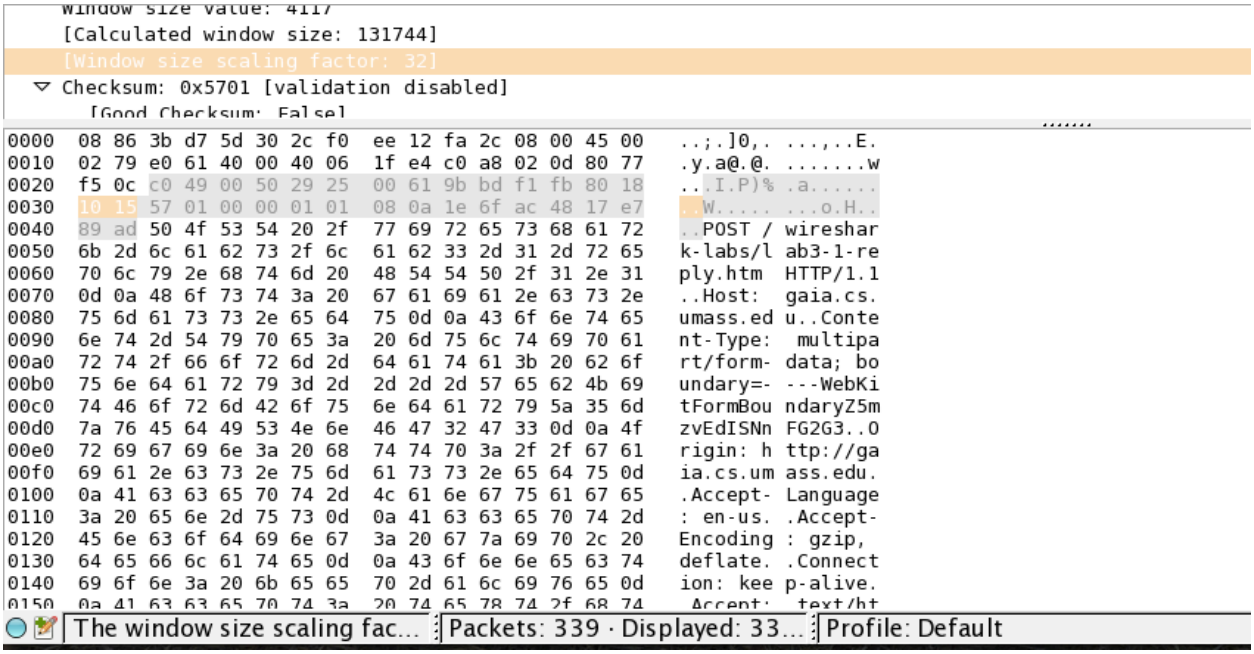
- The SYN segment is buried in packet 149, but the sequence number was still 1. There is the

SYN flag in the header.

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?

- Relative Sequence Number was 1. The Acknowledgement number is 1. The server determined that value by receiving the previous SYN packet. The header identified the the SYN ACK segment.

6. What is the sequence number of the TCP segment containing the HTTP POST command?



- POST had a sequence number of 1.

7. Consider the TCP segment containing the HTTP POST as the first segment in the TCP connection.

- The sequence numbers of the first six segments are 1, 582, 719, 2167, 582, and 3615

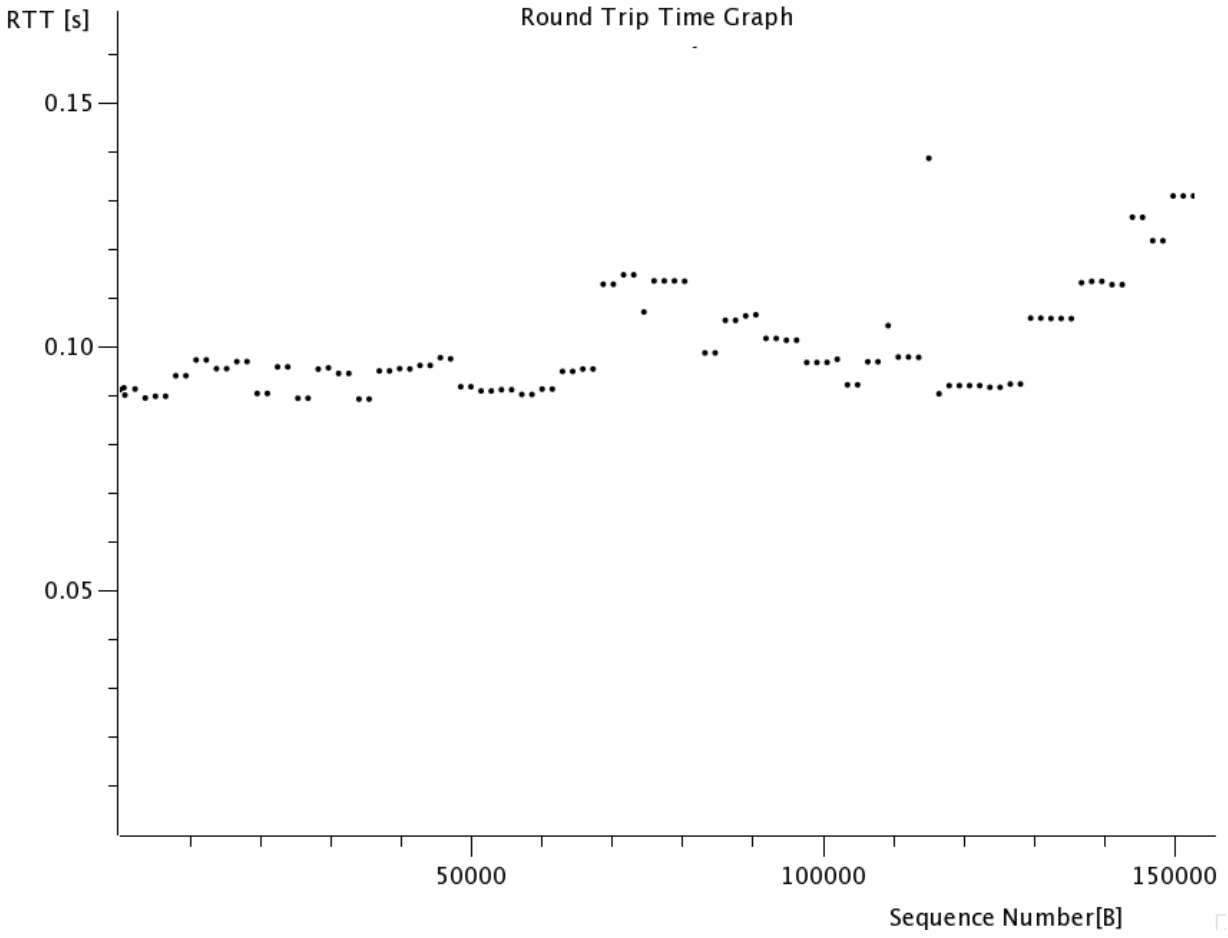
- At what time was each segment sent? Segments were sent at 2.7306, 2.8333, 2.8422, 2.9238, 2.2323, and 2.9323

- When was the ACK for each segment received? ACKs were received at 2.8310, 2.9237, 2.9247, 2.9322, 2.9334 and 3.0133.

- 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?

- The RTT times were 0.1004, .0904, 0.825, 0.840, 0.7011, and 0.081

- What is the `EstimatedRTT` value 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 239 for all subsequent segments.

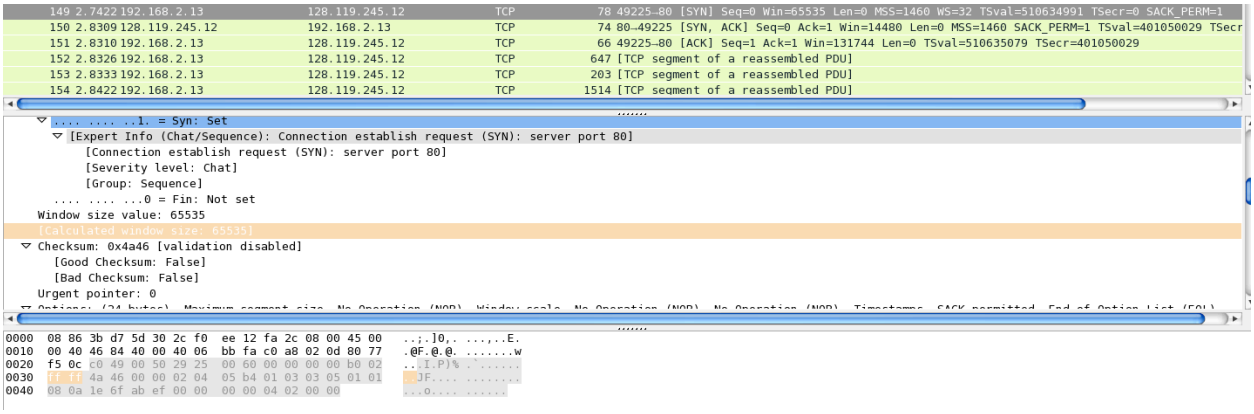


8. What is the length of each of the first six TCP segments?

- The lengths of the first six TCP segments are 647, 203, 1514, 1514, 1514, and 1514.

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?

- Minimum available buffer space was 65535. Lack of buffer space did not throttle the sender because our sizes did not exceed 1514.



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

- No segments were retransmitted because none of the ACK messages were duplicated.

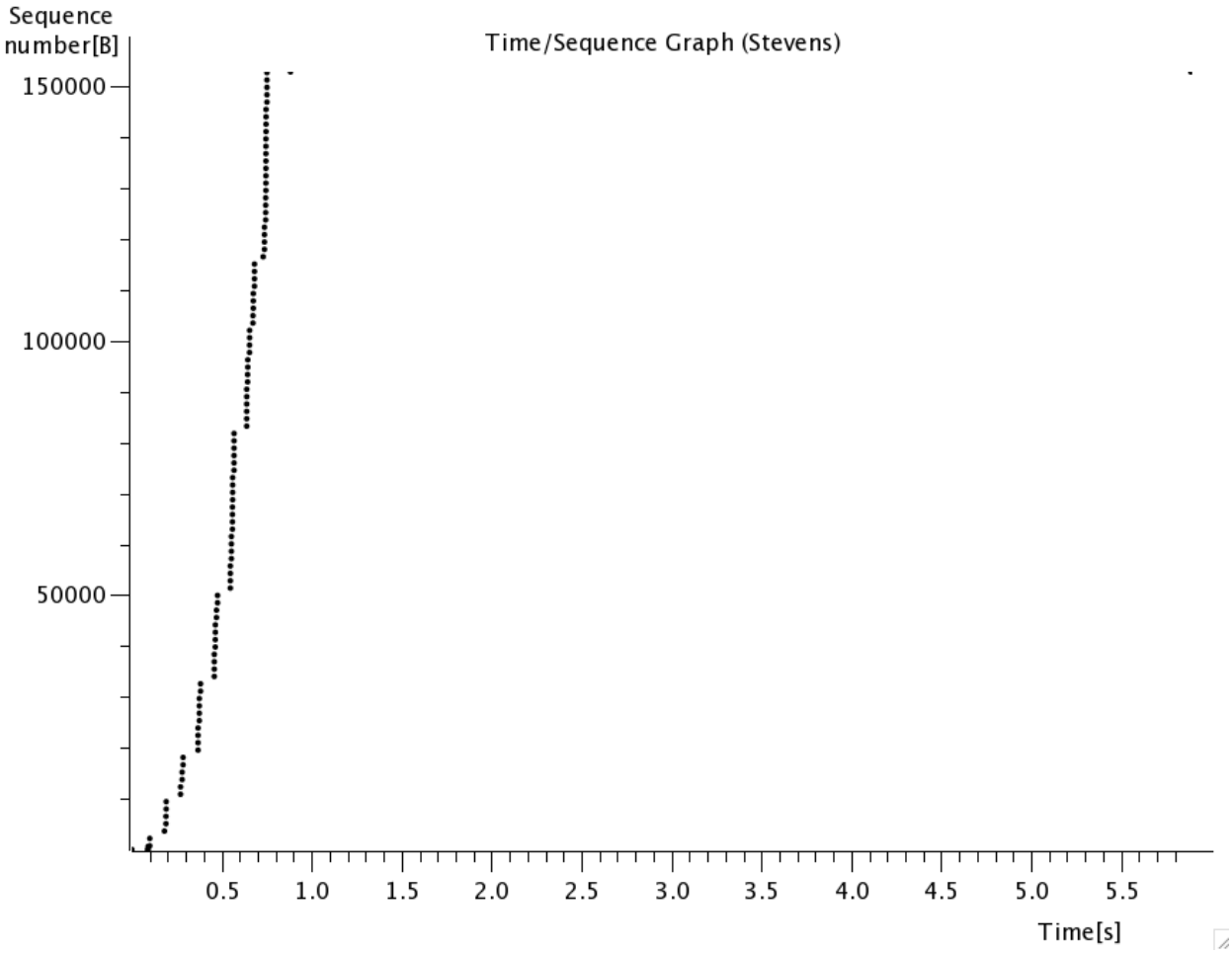
11. How much data does the receiver typically acknowledge in an ACK? Can you identify cases where the receiver is ACKing every other received segment (see Table 3.2 on page 247 in the text).

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

- Total throughput of the transfer is 534,710/(8.6301-2.742). That's the file size divided by the end time minus the start time.

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 slow start 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.

    - Congestion control did no need to take over. We know this because we learned in class that the rate grows exponentially, as shown below, and if needed, it would drop. There are no sudden drops in our graph.


Sequence number[B]
Time/Sequence Graph (Stevens)
150000
100000
50000
0.5
1.0
1.5
2.0
2.5
3.0
3.5
4.0
4.5
5.0
5.5
Time[s]