**CSCE 560**

**Wireshark Lab 3**

**14 Nov 18**

**Marvin Newlin**

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?

**Sol’n:** The red boxes in the picture below show the IP address and TCP port from my computer as 192.168.1.9:55503.

2. What is the IP address of gaia.cs.umass.edu? On what port number is it sending

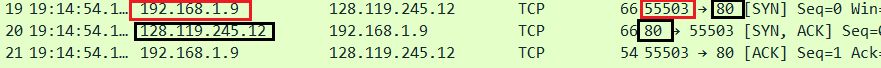
and receiving TCP segments for this connection?

**Sol’n:** The Black boxes show that their IP address is 128.119.245.12 and they are receiving on port 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?

**Sol’n:** Same as answer for problem 1.

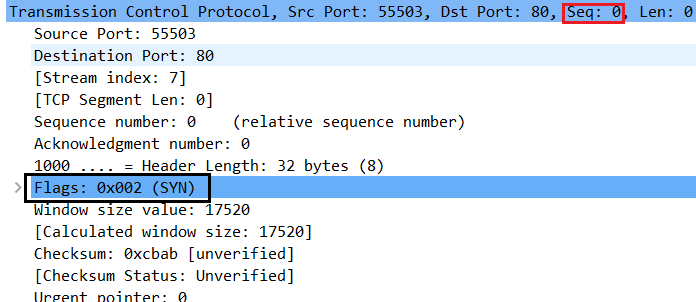


Picture for probs 1-3

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?



**Sol’n:** Here we see that the sequence number is 0 (red box) and in the black box we see that the SYN flag is set.

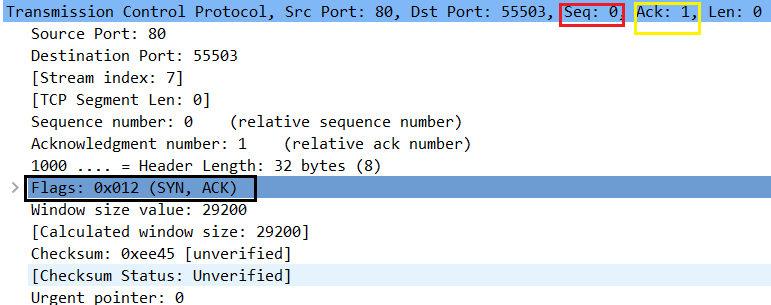
5. What is the sequence number of the SYNACK segment sent by gaia.cs.umass.edu

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?



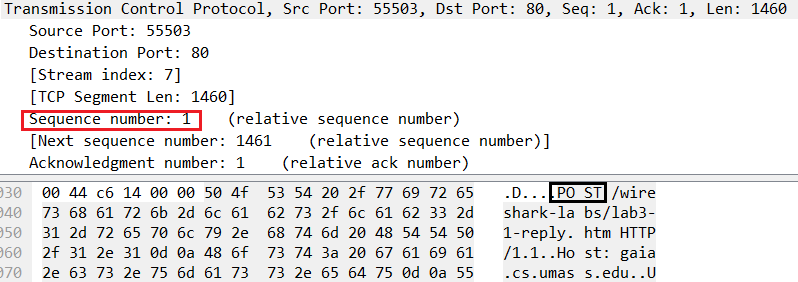
**Sol’n:** Here we see that the sequence number is 0 (red box) and the ACK value is 1 (yellow box). Both the SYN and ACK flags are set (black box).

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.



**Sol’n:** The red box shows us the seq number is 1 and the black box shows us that this segment contains the HTTP POST.

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

TCP connection.

1. What are the sequence numbers of the first six segments in the TCP connection (including the segment containing the HTTP POST)?
2. At what time was each segment sent?
3. When was the ACK for each segment received?
4. 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?

1. 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.*

**Sol’n:**

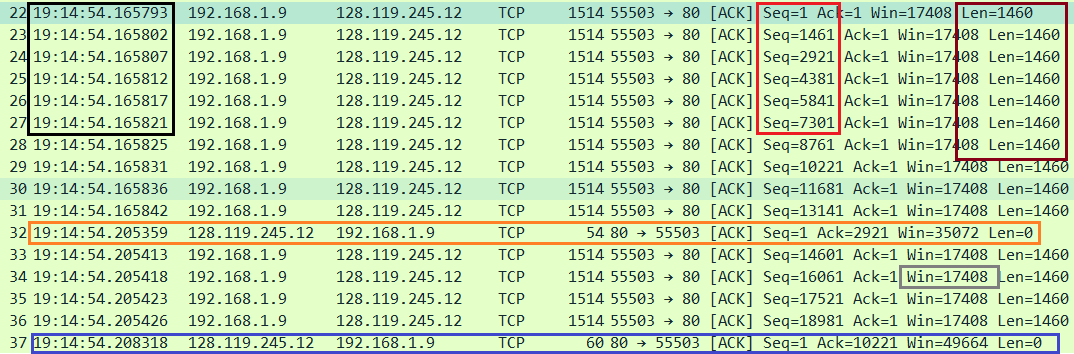
Answers to 7 & 8 in table below with notes below the table.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Sequence # | Time sent 19:14:54 | Time ACK rcd (19:14:54) | RTT  (seconds) | Est RTT (seconds) | Seg Length (bytes) |
| 1 | . 165793 | .205359 (cumulative 1-2) | 0.039566 | 0.039566 | 1460 |
| 1461 | .165802 | .205359 (cumulative 1-2) | 0.039557 | 0.039565 | 1460 |
| 2921 | .165807 | .208318  (cumulative 3-7) | 0.042511 | 0.039933 | 1460 |
| 4381 | .165812 | .208318  (cumulative 3-7) | 0.042506 | 0.040255 | 1460 |
| 5841 | .165817 | .208318  (cumulative 3-7) | 0.042501 | 0.040536 | 1460 |
| 7301 | .165821 | .208318  (cumulative 3-7) | 0.042497 | 0.040781 | 1460 |

Assuming alpha = 0.125 then,

EstimatedRTT = 0.875 • EstimatedRTT + 0.125 • SampleRTT

First EstRTT = SampleRTT0 = 0.039566



Screenshot for prob. 7, 8, 9. Black box contains send times, red box is sequence numbers, orange and blue boxes are ACKs, burgundy box is segment lengths

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

In the burgundy box (and table above) we see that the length of each segment is 1460 bytes.

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?

**Sol’n:** Highlighted in the grey box is the window size of 17408. This is the smallest window size sent through the entire trace, so this is the minimum amount of buffer space advertised because the window size is how much room the sender has in their buffer.

10. Are there any retransmitted segments in the trace file? What did you check for (in

the trace) in order to answer this question?

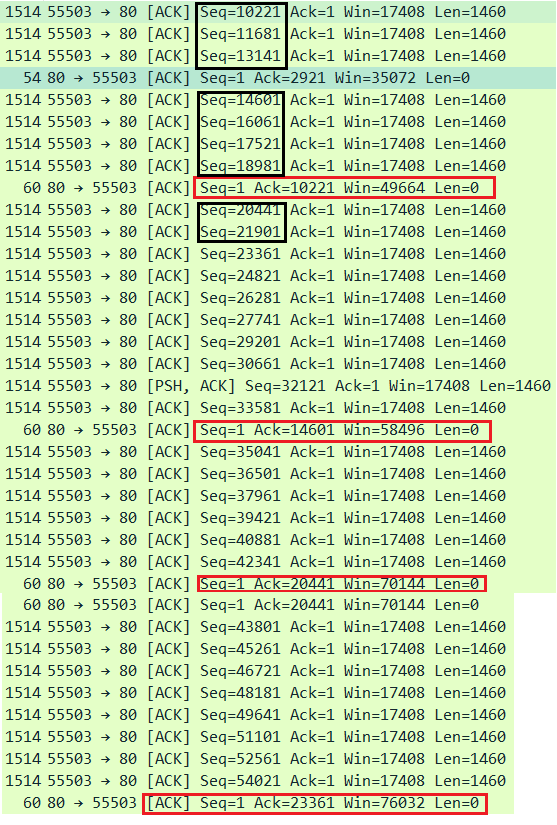
**Sol’n:** There were no retransmitted segments. I was looking for 3 duplicate ACKs in the trace and there were no duplicate ACKs

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).

**Sol’n:**In the picture below, we can see an example of some ACKs. The red boxes indicate the ACK messages and the black boxes show the segments sent between ACKs. From the black boxes we see that it typically ACKs about 2-4 segments per ACK, (2920-5840 bytes)



12. What is the throughput (bytes transferred per unit time) for the TCP connection?

Explain how you calculated this value.

**Sol’n:**

First byte sent at 19:14.54.165793, last captured packet entry listed below.

170 19:14:54.345395 128.119.245.12 192.168.1.9 TCP 60 80 → 55503 [ACK] Seq=1 Ack=150381 Win=242560 Len=0

This frame entry indicates that the last byte was ACKed at 19:14:54.345395 giving us a total time of transfer of 19:14:54.345395 - 19:14.54.165793 = 0.179602sec

Size of file = 152,138 bytes (based on windows file size properties)

Throughput = size/transfer time = 152138\*8 bits/0.179602 sec = 6776672.8 bps

Throughput = 6.78 Mbps

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

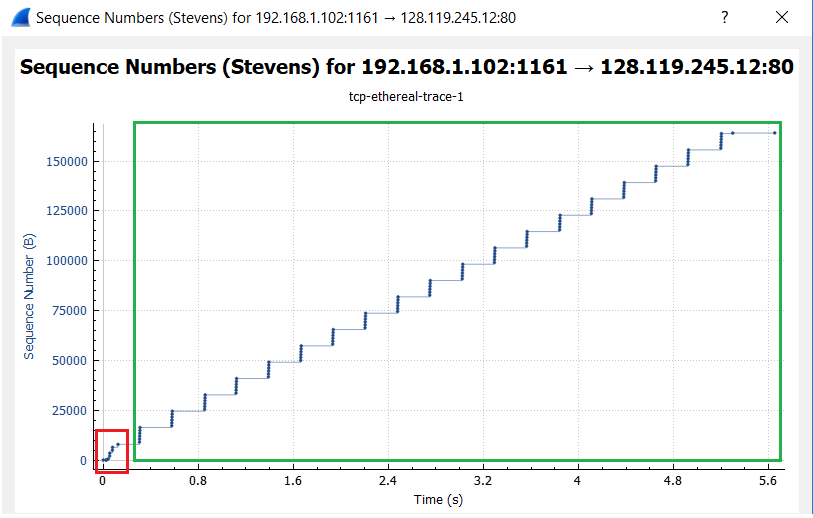
number versus time plot of segments being sent from the client to the

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.



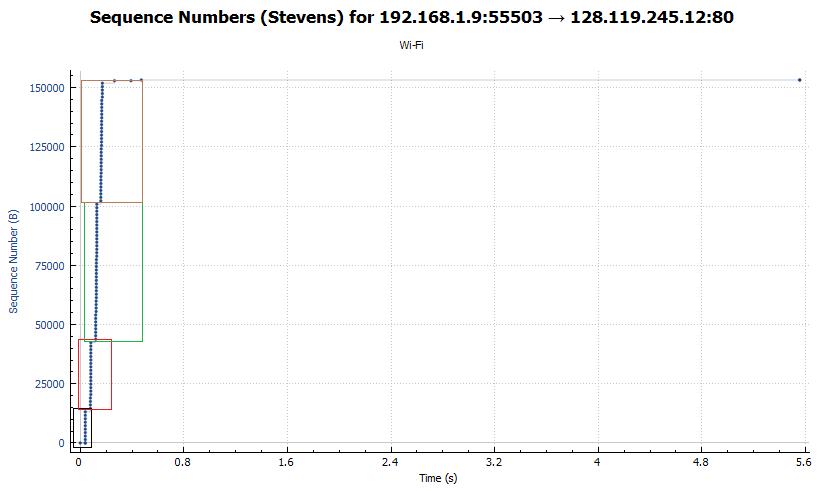
**Sol’n:**

In the red box we can see that the number of segments sent back to back starts low and then increases so this is the slow start phase. Then from there in the green box the number of segments sent back to back seems very constant (green box) each time so it’s probably the congestion avoidance even though it behaves differently.

This behaves differently from ideal TCP in that the slow start is very small, and the congestion avoidance doesn’t continually do additive increase for the number of segments it sends.

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



**Sol’n:**

Above we can see that the slow start begins with a certain number of segments sent back to back in the black (first) box. Then the next set of segments sent is roughly double the number of segments (red box). Then the next set is roughly double that size (green box) so this is the slow start phase. The brown box though is about the same size as green so it shows we have reached the threshold and are at the additive increase, i.e. the congestion avoidance section.

It seems to be different in that were still only sending one segment at a time but were sending them back to back whereas in the idealized behavior we treat it as if all segments are transmitted simultaneously. Additionally, in the measured data there are no timeouts or duplicate ACKs so no threshold reductions!