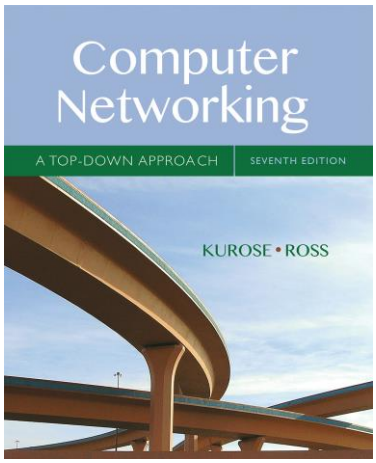
Name: Katie Schaumleffle

Wireshark Lab: TCP v7.0

Supplement to *Computer Networking: A Top-Down Approach, 7th ed.*, J.F. Kurose and K.W. Ross

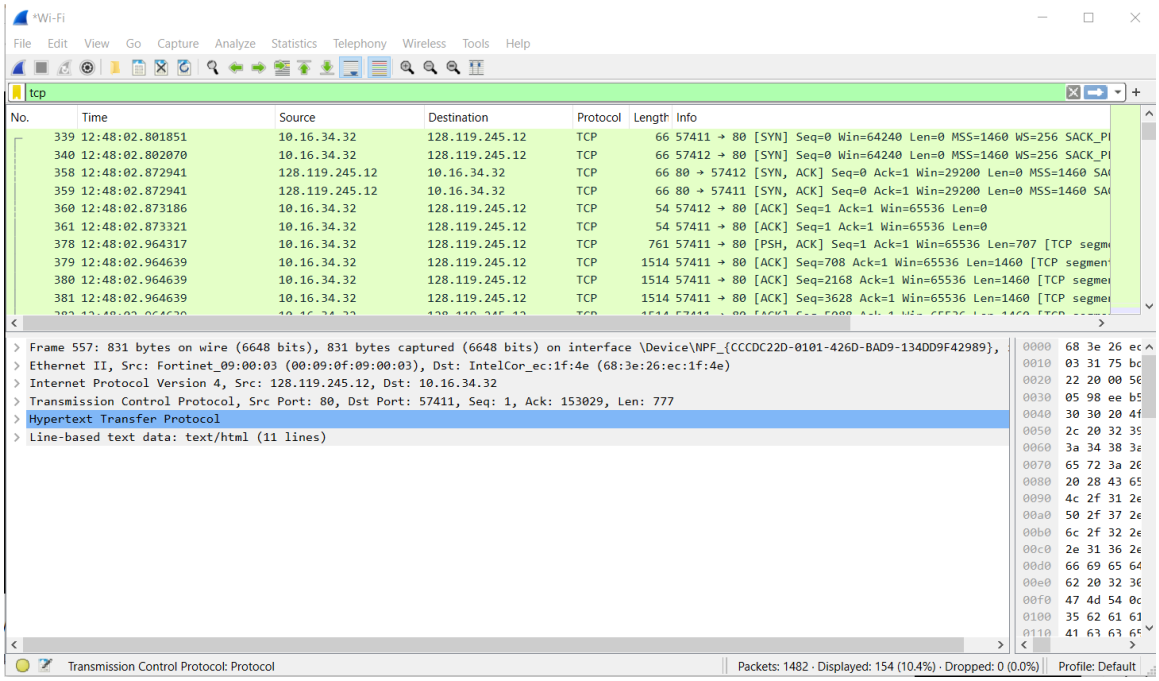


"Tell me and I forget. Show me and I remember. Involve me and I understand." Chinese proverb

1. Capturing a bulk TCP transfer from your computer to a remote server

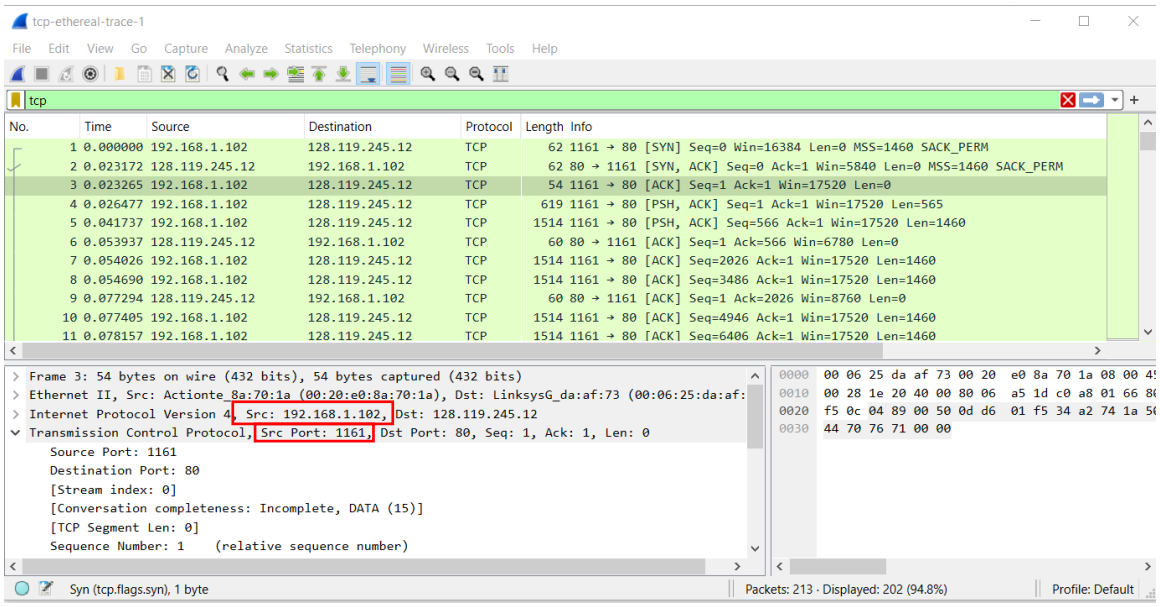
Do the following:

- Start up your web browser. Go the http://gaia.cs.umass.edu/wireshark-labs/alice.txt and retrieve an ASCII copy of *Alice in Wonderland.* Store this file somewhere on your computer.
- Next go to http://gaia.cs.umass.edu/wireshark-labs/TCP-wireshark-file1.html.
- You should see a screen that looks like:

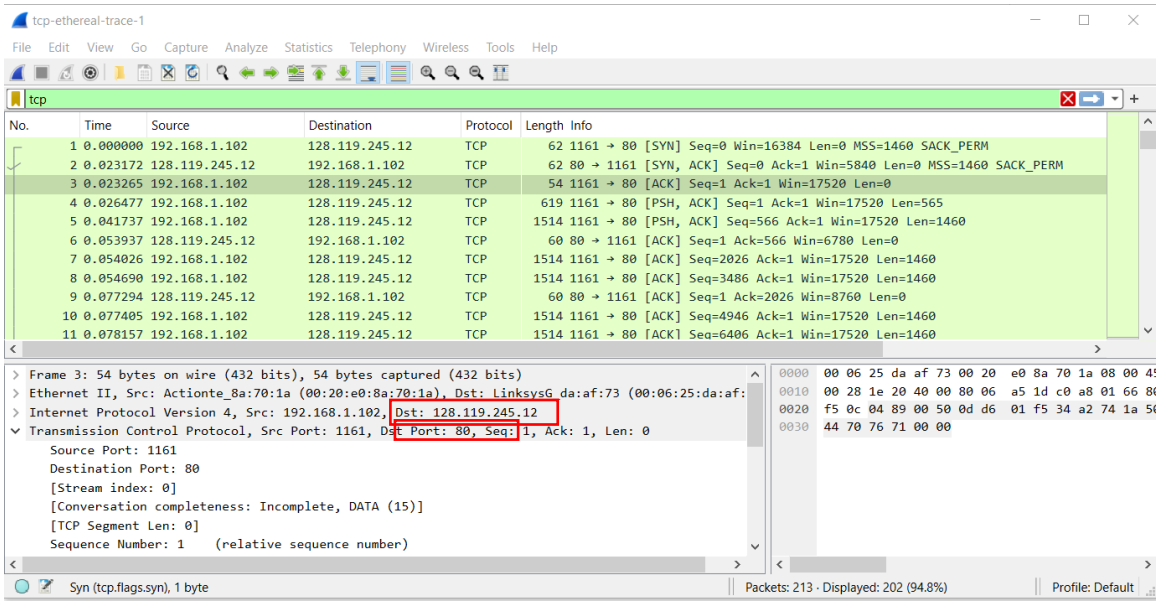


2. A first look at the captured trace

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?
   192.168.1.102:1161

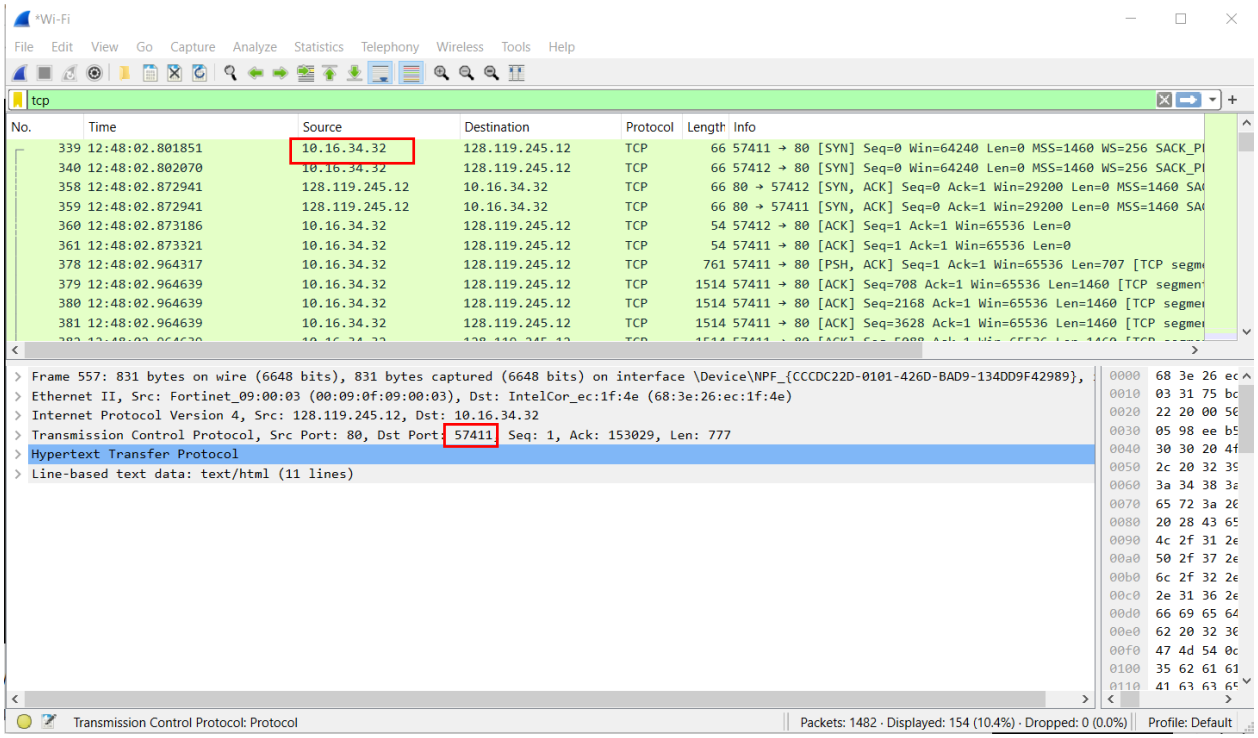


2. What is the IP address of gaia.cs.umass.edu? On what port number is it sending and receiving TCP segments for this connection?
   128.119.245.12:80



If you have been able to create your own trace, answer the following question:

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?
10.16.34.32:57411

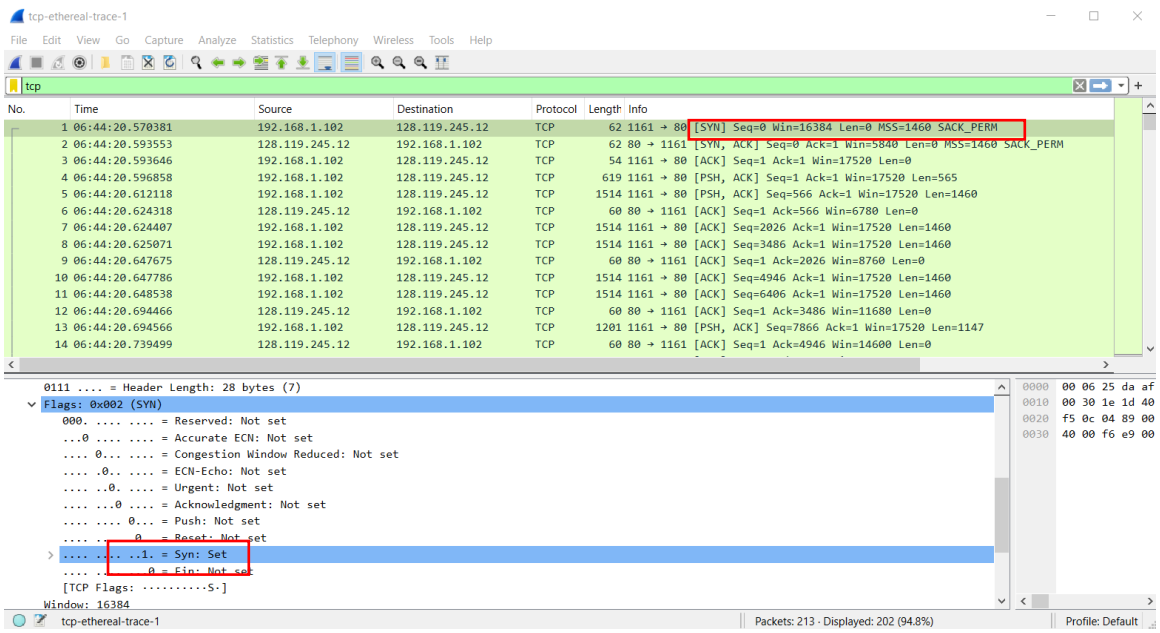


3. TCP Basics

Answer the following questions for the TCP segments:

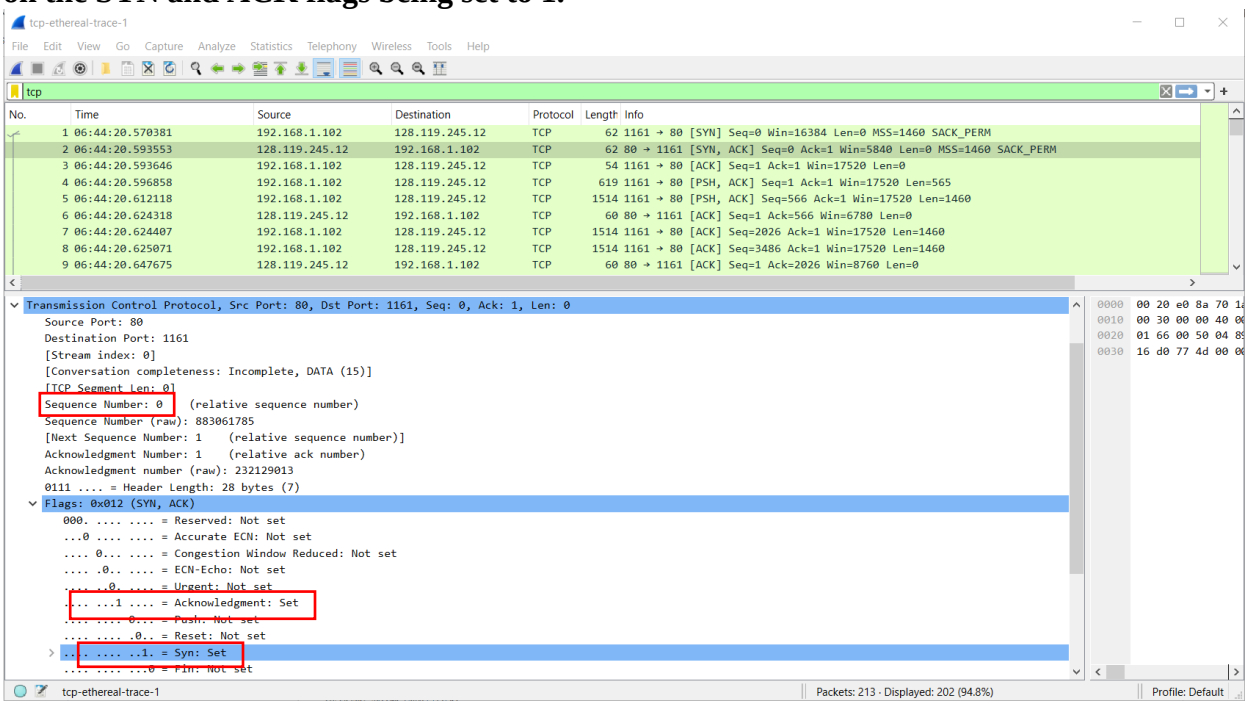
4. What is the sequence number of the TCP SYN segment that is used to initiate the TCP connection between the client computer and gaia.cs.umass.edu? What is it in the segment that identifies the segment as a SYN segment?

The sequence number is 0. The Syn flag is set to 1 which identifies it as a SYN.



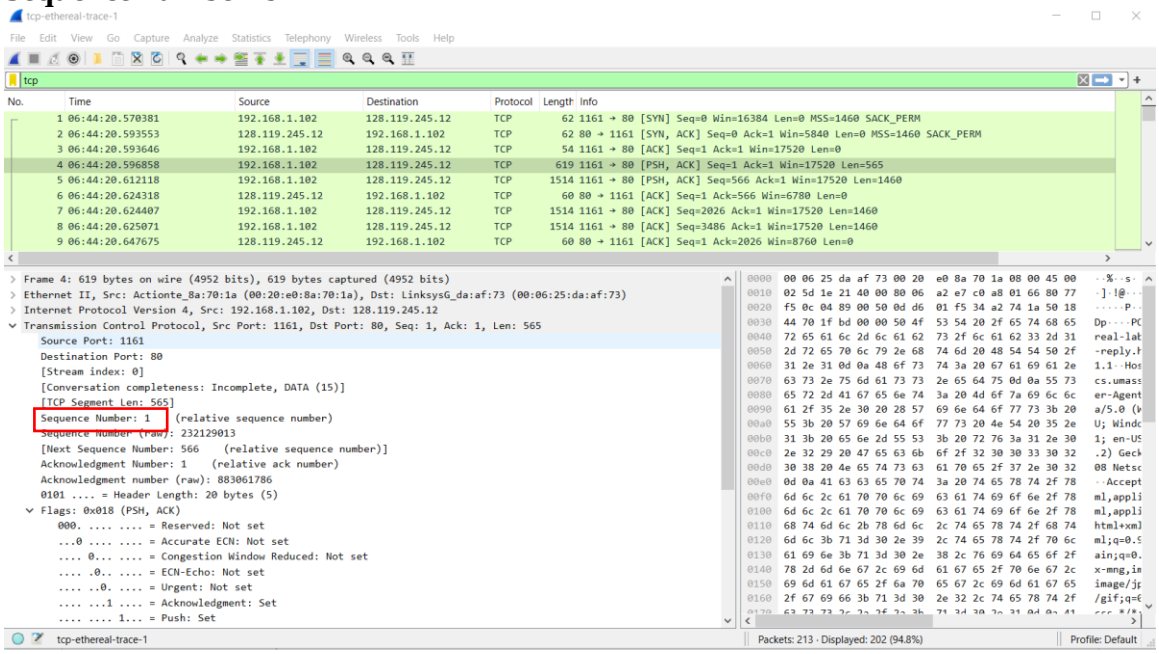
5. What is the sequence number of the SYNACK segment sent by gaia.cs.umass.edu to the client computer in reply to the SYN? What is the value of the Acknowledgement field in the SYNACK segment? How did gaia.cs.umass.edu determine that value? What is it in the segment that identifies the segment as a SYNACK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

The sequence number is 0, and the Acknowledgement number is 1. It was determined by adding 1 to the sequence number. It's identified as SYNACK based on the SYN and ACK flags being set to 1.



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.

Sequence number is 1



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.

Segment 1: sent time 0.026477; ACK = 0.053934, RTT = 0.02746 seconds
Segment 2: Sent = 0.041737; ACK = 0.077294; RTT = 0.035557 seconds
Segment 3: Sent = 0.054026; ACK = 0.124085; RTT = 0.070059 seconds
Segment 4: Sent = 0.054690; ACK = 0.169118; RTT = 0.11443 seconds
Segment 5: Sent = 0.077405; ACK = 0.217299; RTT = 0.13989 seconds
Segment 6: Sent = 0.078157; ACK = 0.267802; RTT = 0.18964 seconds

EstimatedRTT = 0.875 * EstimatedRTT + 0.125 * SampleRTT

EstimatedRTT after the receipt of the ACK of segment 1:
EstimatedRTT = RTT for Segment 1 = 0.02746 seconds

EstimatedRTT after the receipt of the ACK of segment 2:
EstimatedRTT = 0.875 * 0.02746 + 0.125 * 0.035557 = 0.0285 seconds

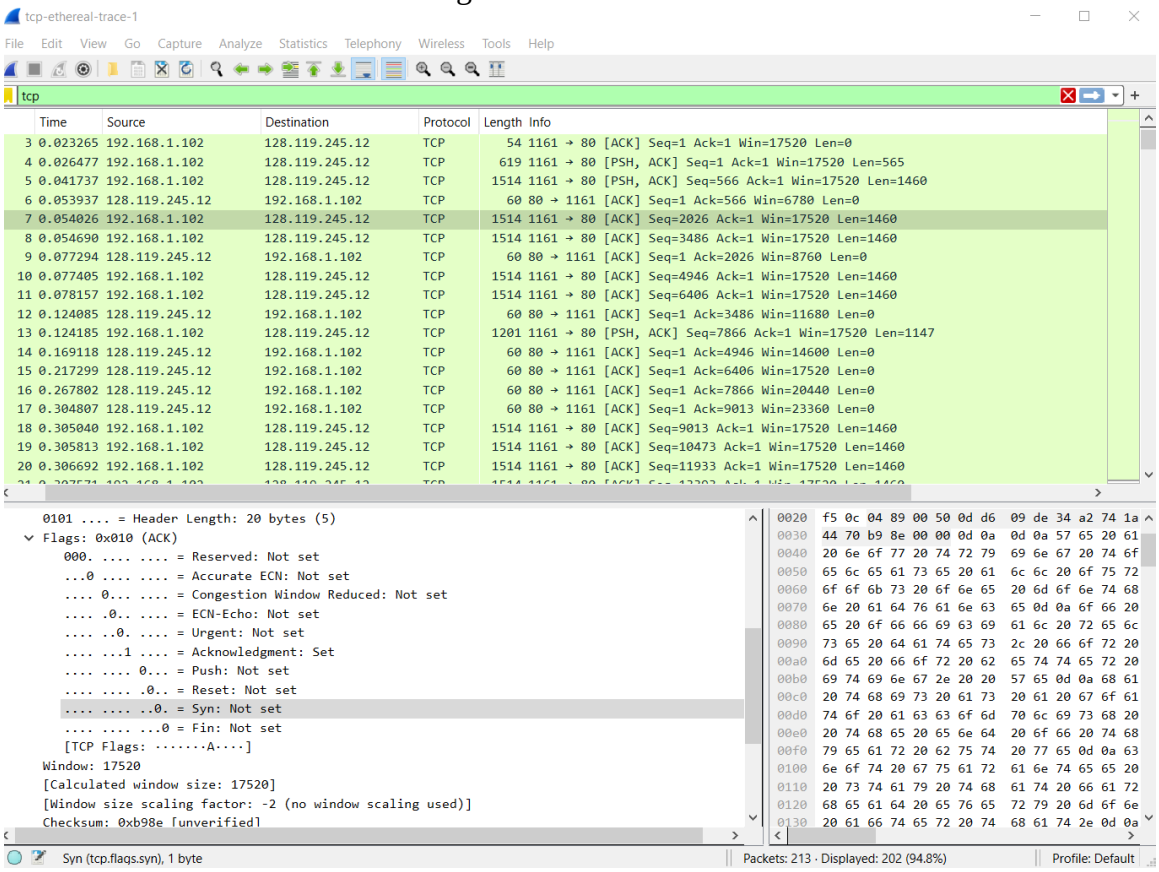
EstimatedRTT after the receipt of the ACK of segment 3:
EstimatedRTT = 0.875 * 0.0285 + 0.125 * 0.070059 = 0.0337 seconds

EstimatedRTT after the receipt of the ACK of segment 4:
EstimatedRTT = 0.875 * 0.0337+ 0.125 * 0.11443 = 0.0438 seconds

EstimatedRTT after the receipt of the ACK of segment 5:
EstimatedRTT = 0.875 * 0.0438 + 0.125 * 0.13989 = 0.0558 seconds

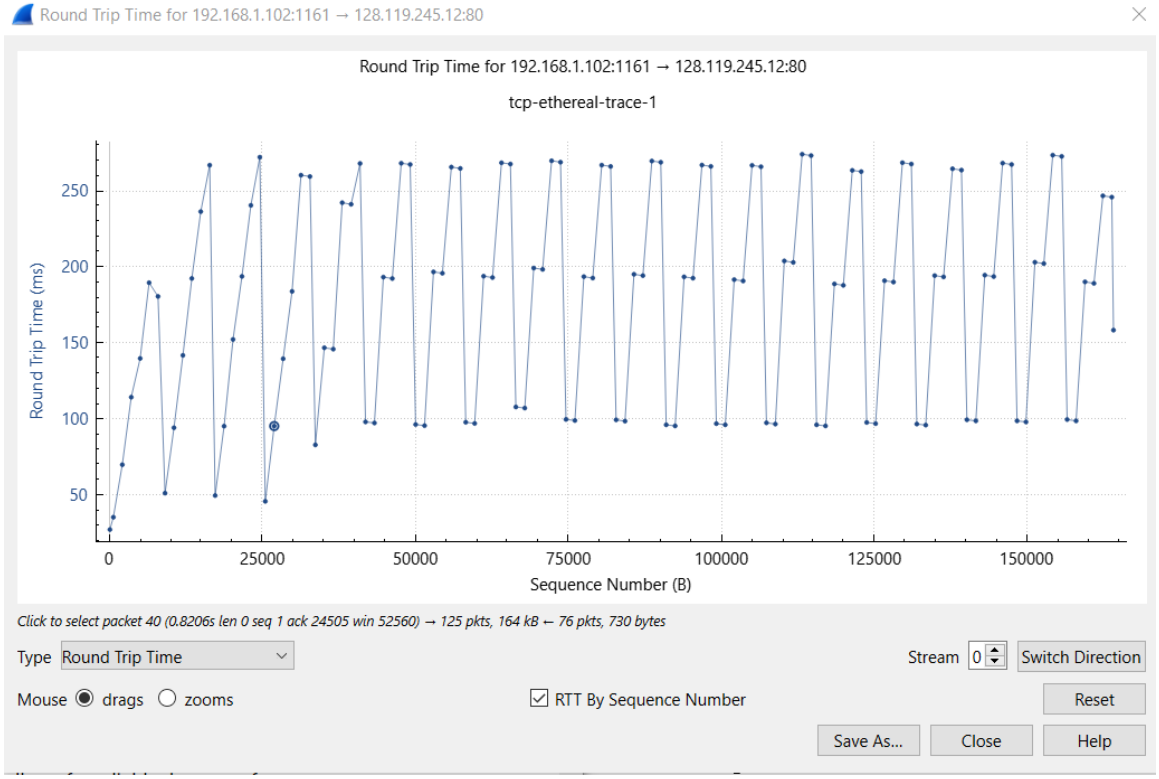
EstimatedRTT after the receipt of the ACK of segment 6:
EstimatedRTT = 0.875 * 0.0558 + 0.125 * 0.18964 = 0.0725 seconds

Screenshot from wireshark used to get time values:

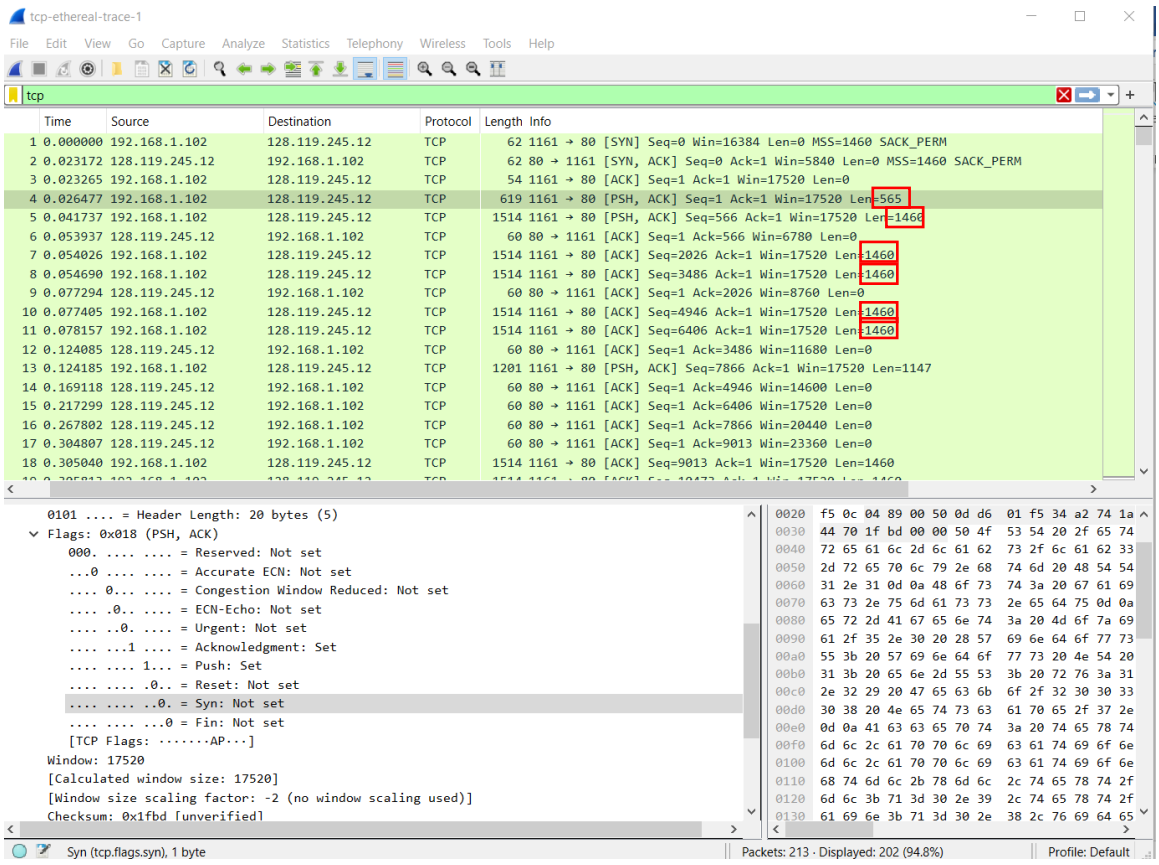


Note: Wireshark has a nice feature that allows you to plot the RTT for each of the TCP segments sent. Select a TCP segment in the "listing of captured packets" window that is being sent from the client to the

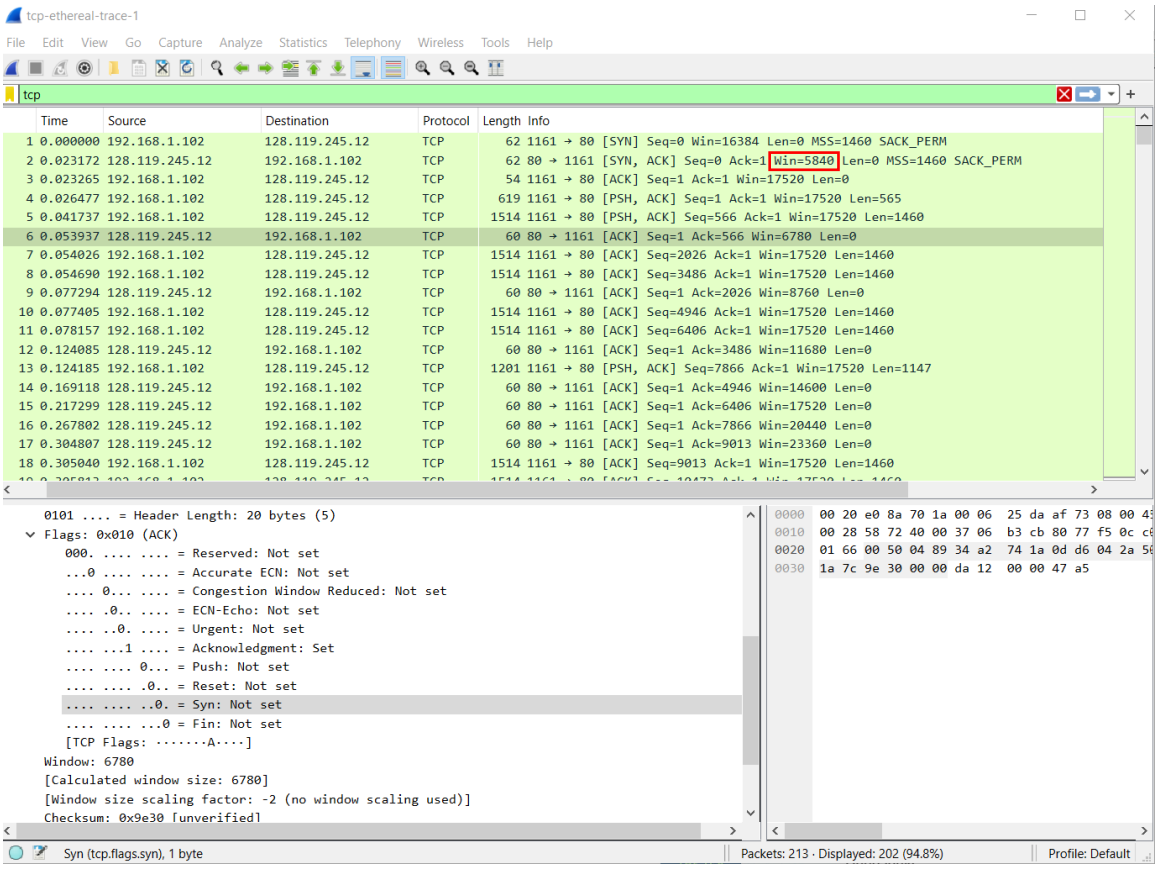
gaia.cs.umass.edu server. Then select: *Statistics->TCP Stream Graph->Round Trip Time Graph.*



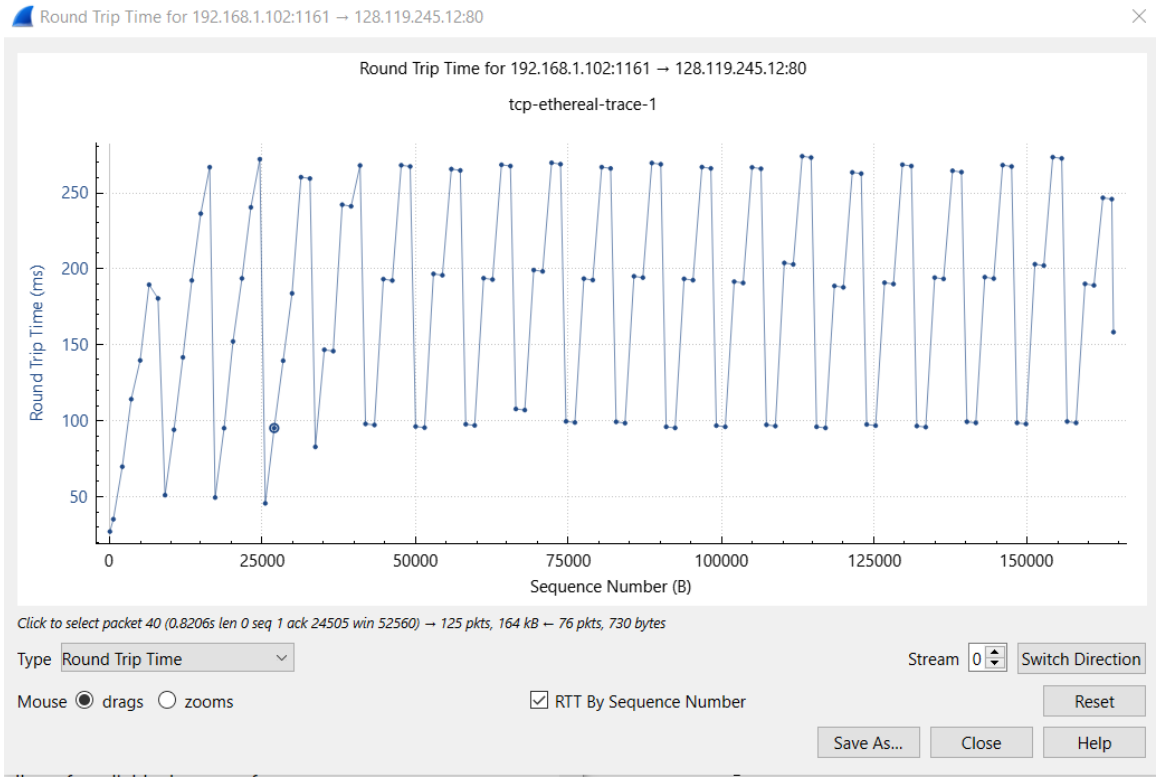
8. What is the length of each of the first six TCP segments?[1]
    a. Packet 4 = 565 bytes
    b. Packet 5 = 1460 bytes
    c. Packet 7 = 1460 bytes
    d. Packet 8 = 1460 bytes
    e. Packet 10 = 1460 bytes
    f. Packet 11 = 1460 bytes



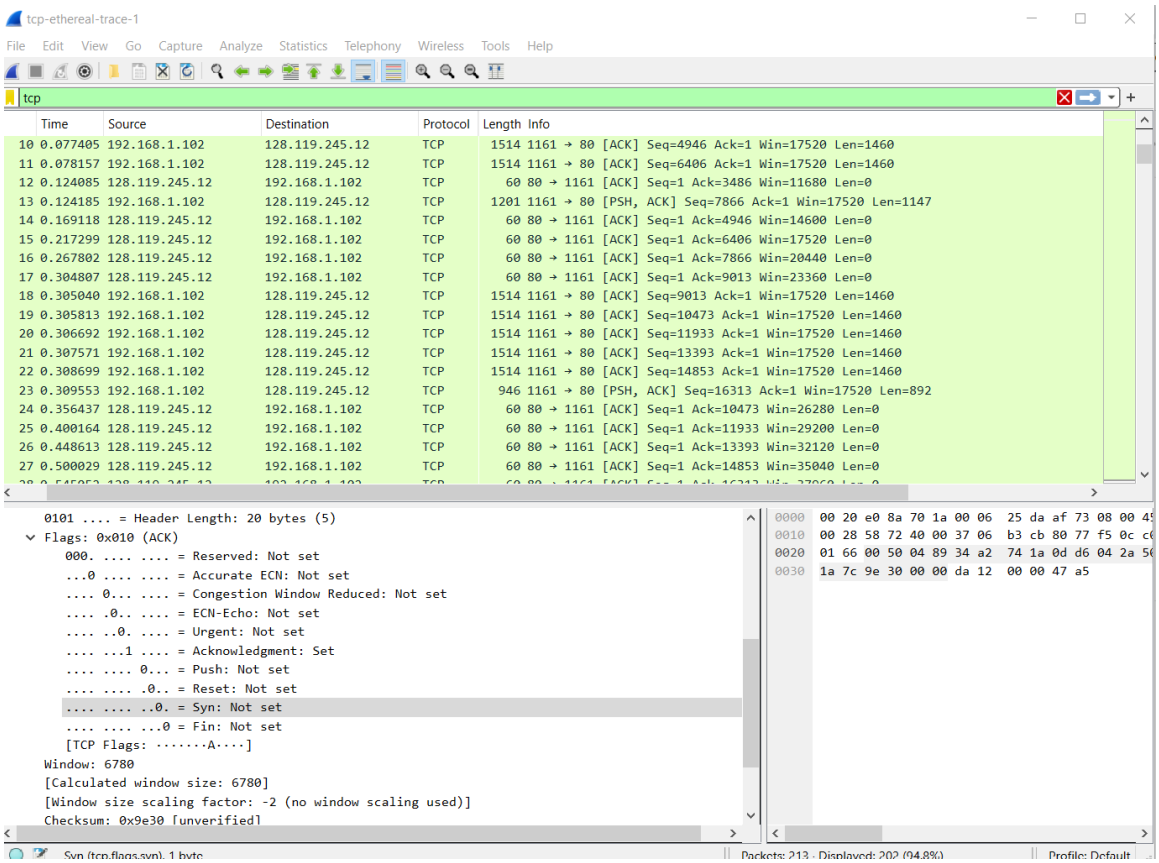
9. What is the minimum amount of available buffer space advertised at the received for the entire trace? Does the lack of receiver buffer space ever throttle the sender?
    a. Window: 5840 bytes, which shows in first ACK from server.
    b. No, it did not. The buffer steadily increased in size until a max receiver buffer of 62,780 bytes were received.



10. Are there any retransmitted segments in the trace file? What did you check for (in the trace) in order to answer this question?
    a. No, there were no retransmitted segments. You can tell by looking at the graph. It never goes backwards and each packet was sent successfully.



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).
    a. We can see that the ACK numbers increase as we go. We can also tell that the ACK numbers increase by 1460 each time, indicating that the receiver is acknowledging 1460 bytes.



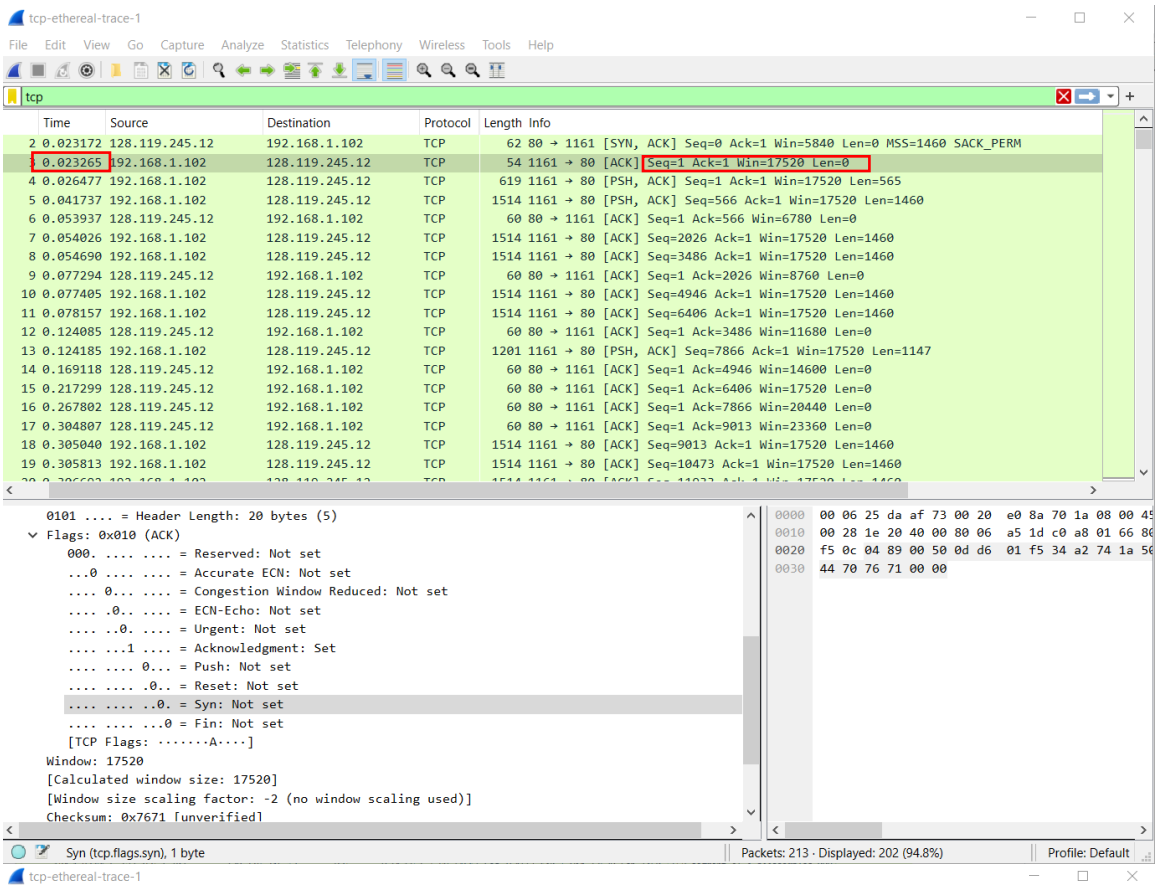
12. What is the throughput (bytes transferred per unit time) for the TCP connection? Explain how you calculated this value.

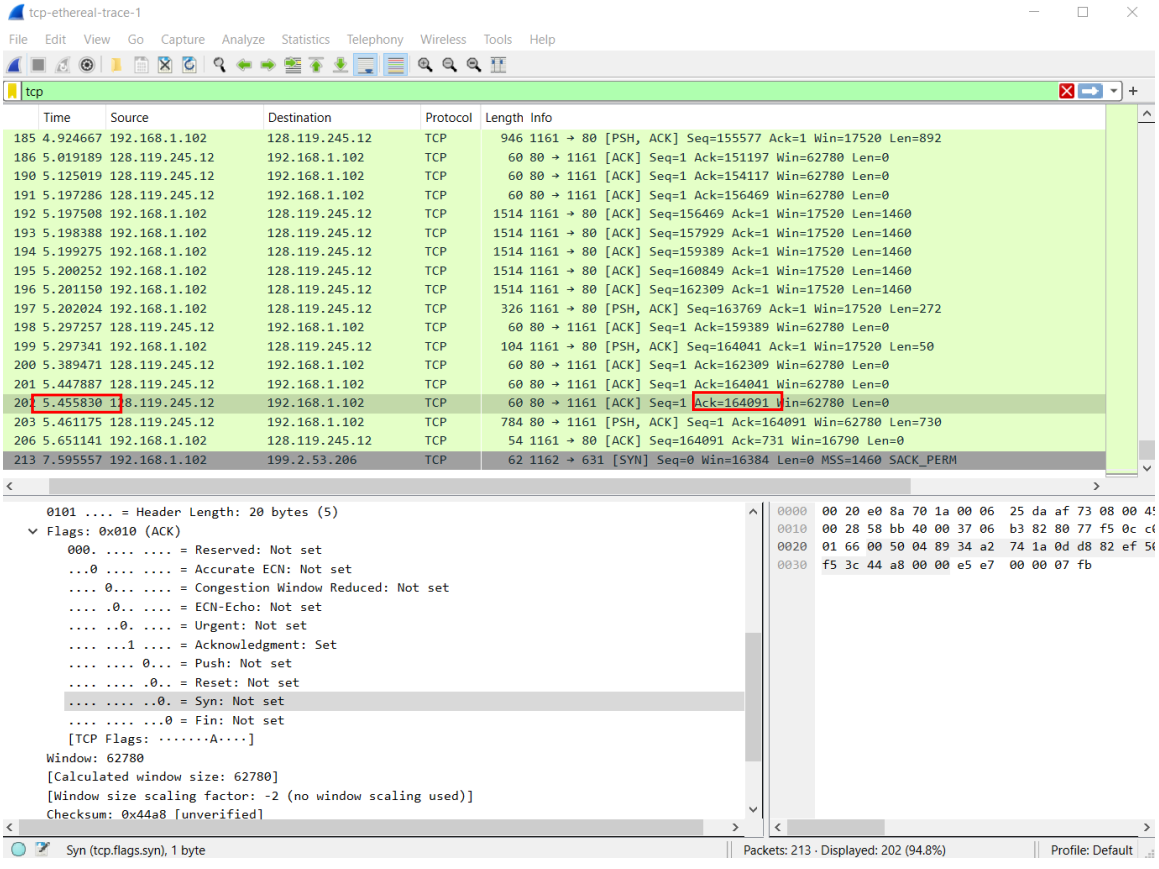
    Total trans in bytes = 164,091-1=164,090 bytes

    Time from first ACK to last = 5.455830 seconds – 0.023265 seconds = 5.432565 seconds

    Throughput = 164,090 bytes / 5.432565 seconds = 30,204.88 bytes/sec

    a. To calculate the throughput, we take the first ACK number that contains seq=1 and len=0, then subtract the first ACK from the last ACK(which is 1). This gives us the total amount transmitted in byte.
    b. Next, we calculate the time from the first ACK to the last.
    c. Lastly, we calculate the throughput by dividing the total transmitted in bytes/time from first ACK to last ACK.

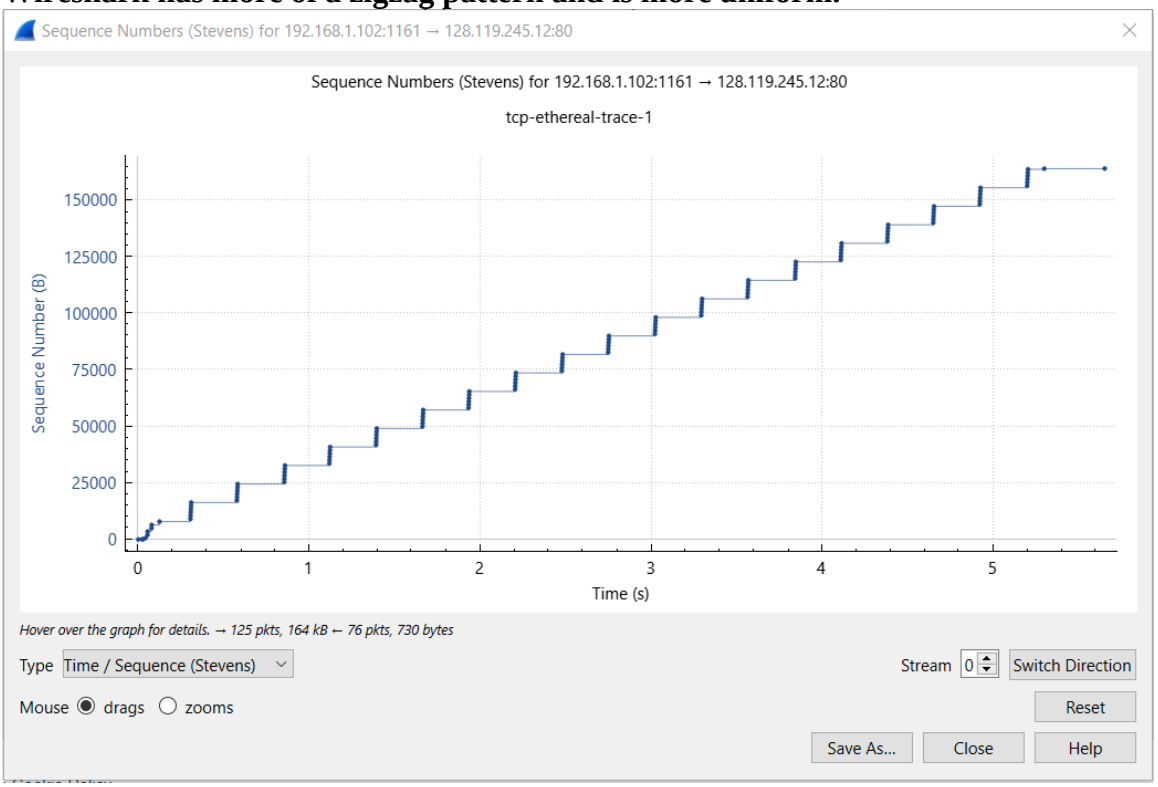

tcp-ethereal-trace-1
File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help
tcp
Time Source Destination Protocol Length Info
2 0.023172 128.119.245.12 192.168.1.102 TCP 62 80 → 1161 [SYN, ACK] Seq=0 Ack=1 Win=5840 Len=0 MSS=1460 SACK_PERM
3 0.023265 192.168.1.102 128.119.245.12 TCP 54 1161 → 80 [ACK] Seq=1 Ack=1 Win=17520 Len=0
4 0.026477 192.168.1.102 128.119.245.12 TCP 619 1161 → 80 [PSH, ACK] Seq=1 Ack=1 Win=17520 Len=565
5 0.041737 192.168.1.102 128.119.245.12 TCP 1514 1161 → 80 [PSH, ACK] Seq=566 Ack=1 Win=17520 Len=1460
6 0.053937 128.119.245.12 192.168.1.102 TCP 60 80 → 1161 [ACK] Seq=1 Ack=566 Win=6780 Len=0
7 0.054026 192.168.1.102 128.119.245.12 TCP 1514 1161 → 80 [ACK] Seq=2026 Ack=1 Win=17520 Len=1460
8 0.054690 192.168.1.102 128.119.245.12 TCP 1514 1161 → 80 [ACK] Seq=3486 Ack=1 Win=17520 Len=1460
9 0.077294 128.119.245.12 192.168.1.102 TCP 60 80 → 1161 [ACK] Seq=1 Ack=2026 Win=8760 Len=0
10 0.077405 192.168.1.102 128.119.245.12 TCP 1514 1161 → 80 [ACK] Seq=4946 Ack=1 Win=17520 Len=1460
11 0.078157 192.168.1.102 128.119.245.12 TCP 1514 1161 → 80 [ACK] Seq=6406 Ack=1 Win=17520 Len=1460
12 0.124085 128.119.245.12 192.168.1.102 TCP 60 80 → 1161 [ACK] Seq=1 Ack=3486 Win=11680 Len=0
13 0.124185 192.168.1.102 128.119.245.12 TCP 1201 1161 → 80 [PSH, ACK] Seq=7866 Ack=1 Win=17520 Len=1147
14 0.169118 128.119.245.12 192.168.1.102 TCP 60 80 → 1161 [ACK] Seq=1 Ack=4946 Win=14600 Len=0
15 0.217299 128.119.245.12 192.168.1.102 TCP 60 80 → 1161 [ACK] Seq=1 Ack=6406 Win=17520 Len=0
16 0.267802 128.119.245.12 192.168.1.102 TCP 60 80 → 1161 [ACK] Seq=1 Ack=7866 Win=20440 Len=0
17 0.304807 128.119.245.12 192.168.1.102 TCP 60 80 → 1161 [ACK] Seq=1 Ack=9013 Win=23360 Len=0
18 0.305040 192.168.1.102 128.119.245.12 TCP 1514 1161 → 80 [ACK] Seq=9013 Ack=1 Win=17520 Len=1460
19 0.305813 192.168.1.102 128.119.245.12 TCP 1514 1161 → 80 [ACK] Seq=10473 Ack=1 Win=17520 Len=1460
0101 .... = Header Length: 20 bytes (5)
Flags: 0x010 (ACK)
000. .... = Reserved: Not set
...0 .... = Accurate ECN: Not set
.... 0... .... = Congestion Window Reduced: Not set
.... .0.. .... = ECN-Echo: Not set
.... ..0. .... = Urgent: Not set
....1 .... = Acknowledgment: Set
.... 0... = Push: Not set
.... .0.. = Reset: Not set
....0. = Syn: Not set
....0 = Fin: Not set
[TCP Flags: ·····A····]
Window: 17520
[Calculated window size: 17520]
[Window size scaling factor: -2 (no window scaling used)]
Checksum: 0x7671 [unverified]
0000 00 06 25 da af 73 00 20 e0 8a 70 1a 08 00 45
0010 00 28 1e 20 40 00 80 06 a5 1d c0 a8 01 66 80
0020 f5 0c 04 89 00 50 0d d6 01 f5 34 a2 74 1a 50
0030 44 70 76 71 00 00
Syn (tcp.flags.syn), 1 byte
Packets: 213 · Displayed: 202 (94.8%)
Profile: Default


tcp-ethereal-trace-1
File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help
tcp
Time Source Destination Protocol Length Info
185 4.924667 192.168.1.102 128.119.245.12 TCP 946 1161 → 80 [PSH, ACK] Seq=155577 Ack=1 Win=17520 Len=892
186 5.019189 128.119.245.12 192.168.1.102 TCP 60 80 → 1161 [ACK] Seq=1 Ack=151197 Win=62780 Len=0
190 5.125019 128.119.245.12 192.168.1.102 TCP 60 80 → 1161 [ACK] Seq=1 Ack=154117 Win=62780 Len=0
191 5.197286 128.119.245.12 192.168.1.102 TCP 60 80 → 1161 [ACK] Seq=1 Ack=156469 Win=62780 Len=0
192 5.197508 192.168.1.102 128.119.245.12 TCP 1514 1161 → 80 [ACK] Seq=156469 Ack=1 Win=17520 Len=1460
193 5.198388 192.168.1.102 128.119.245.12 TCP 1514 1161 → 80 [ACK] Seq=157929 Ack=1 Win=17520 Len=1460
194 5.199275 192.168.1.102 128.119.245.12 TCP 1514 1161 → 80 [ACK] Seq=159389 Ack=1 Win=17520 Len=1460
195 5.200252 192.168.1.102 128.119.245.12 TCP 1514 1161 → 80 [ACK] Seq=160849 Ack=1 Win=17520 Len=1460
196 5.201150 192.168.1.102 128.119.245.12 TCP 1514 1161 → 80 [ACK] Seq=162309 Ack=1 Win=17520 Len=1460
197 5.202024 192.168.1.102 128.119.245.12 TCP 326 1161 → 80 [PSH, ACK] Seq=163769 Ack=1 Win=17520 Len=272
198 5.297257 128.119.245.12 192.168.1.102 TCP 60 80 → 1161 [ACK] Seq=1 Ack=159389 Win=62780 Len=0
199 5.297341 192.168.1.102 128.119.245.12 TCP 104 1161 → 80 [PSH, ACK] Seq=164041 Ack=1 Win=17520 Len=50
200 5.389471 128.119.245.12 192.168.1.102 TCP 60 80 → 1161 [ACK] Seq=1 Ack=162309 Win=62780 Len=0
201 5.447887 128.119.245.12 192.168.1.102 TCP 60 80 → 1161 [ACK] Seq=1 Ack=164041 Win=62780 Len=0
202 5.455830 128.119.245.12 192.168.1.102 TCP 60 80 → 1161 [ACK] Seq=1 Ack=164091 Win=62780 Len=0
203 5.461175 128.119.245.12 192.168.1.102 TCP 784 80 → 1161 [PSH, ACK] Seq=1 Ack=164091 Win=62780 Len=730
206 5.651141 192.168.1.102 128.119.245.12 TCP 54 1161 → 80 [ACK] Seq=164091 Ack=731 Win=16790 Len=0
213 7.595557 192.168.1.102 199.2.53.206 TCP 62 1162 → 631 [SYN] Seq=0 Win=16384 Len=0 MSS=1460 SACK_PERM
0101 .... = Header Length: 20 bytes (5)
Flags: 0x010 (ACK)
000. .... = Reserved: Not set
...0 .... = Accurate ECN: Not set
.... 0... .... = Congestion Window Reduced: Not set
.... .0.. .... = ECN-Echo: Not set
.... ..0. .... = Urgent: Not set
....1 .... = Acknowledgment: Set
.... 0... = Push: Not set
.... .0.. = Reset: Not set
....0. = Syn: Not set
....0 = Fin: Not set
[TCP Flags: ·····A····]
Window: 62780
[Calculated window size: 62780]
[Window size scaling factor: -2 (no window scaling used)]
Checksum: 0x44a8 [unverified]
0000 00 20 e0 8a 70 1a 00 06 25 da af 73 08 00 45
0010 00 28 58 bb 40 00 37 06 b3 82 80 77 f5 0c c0
0020 01 66 00 50 04 89 34 a2 74 1a 0d d8 82 ef 50
0030 f5 3c 44 a8 00 00 e5 e7 00 00 07 fb
Syn (tcp.flags.syn), 1 byte
Packets: 213 · Displayed: 202 (94.8%)
Profile: Default

4. TCP congestion control in action

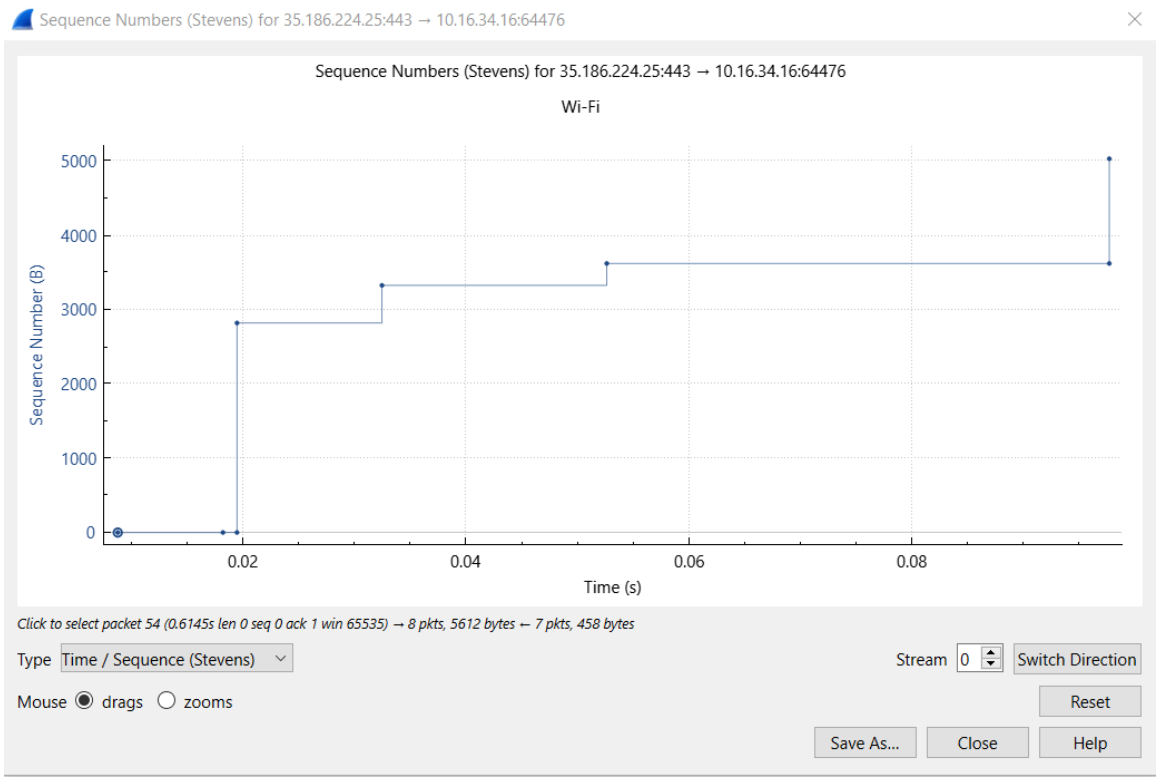
13. Use the *Time-Sequence-Graph(Stevens)* plotting tool to view the sequence number versus time plot of segments being sent from the client to the gaia.cs.umass.edu server. Can you identify where TCP's slowstart phase begins and ends, and where congestion avoidance takes over? Comment on ways in which the measured data differs from the idealized behavior of TCP that we've studied in the text.

    The slowstart phase begins around 0 and ends around 0.1 or 0.15. The congestion avoidance takes over around 0.3 and occurs at every vertical bar with 6 packets until about 5.2 seconds. The text shows a slow start exponential graph which looks very different than our graph. Our graph in Wireshark has more of a zigzag pattern and is more uniform.



14. Answer Question 13 for the trace that you captured when you transferred a file from your ***own*** computer to gaia.cs.umass.edu

    Based on the trace I pulled, there is a slow start, then it grows pretty quickly and then levels off again. My trace wasn't nearly as uniform as the trace from the zip file.


Sequence Numbers (Stevens) for 35.186.224.25:443 → 10.16.34.16:64476
Sequence Numbers (Stevens) for 35.186.224.25:443 → 10.16.34.16:64476
Wi-Fi
Sequence Number (B)
5000
4000
3000
2000
1000
0
0.02
0.04
0.06
0.08
Time (s)
Click to select packet 54 (0.6145s len 0 seq 0 ack 1 win 65535) → 8 pkts, 5612 bytes ← 7 pkts, 458 bytes
Type
Time / Sequence (Stevens)
Stream
0
Switch Direction
Mouse
drags
zooms
Reset
Save As...
Close
Help