CS 372 Lab 2

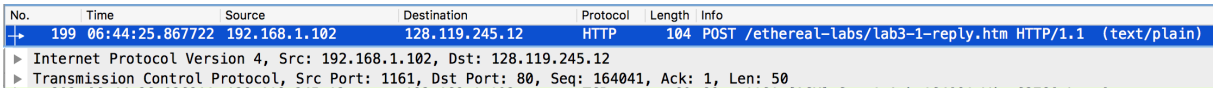
Joanna Lew

Summer 2017

Question 1

Q: 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, its 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 youre uncertain about the Wireshark windows.)

A: The IP address of the client is 192.168.1.102, and the port number is 1161.



Question 2

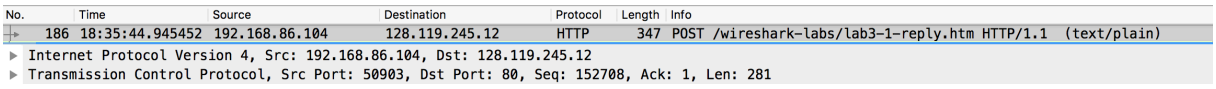
Q: What is the IP address of gaia.cs.umass.edu? On what port number is it sending and receiving TCP segments for this connection?

A: The IP address of gaia.cs.umass.edu is 128.119.245.12, and the port number is 80.
(see image from Question 1)

Question 3

Q: If you have been able to create your own trace, answer the following question:
What is the IP address and TCP port number used by your client computer (source) to transfer the file to gaia.cs.umass.edu?

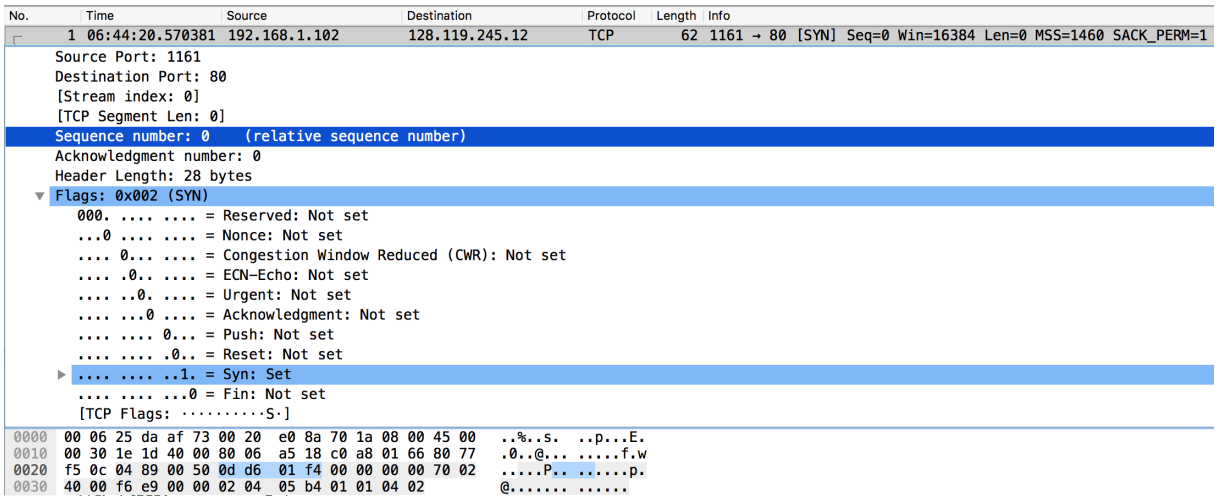
A: The IP address of the client is 192.168.86.104, and the port number is 50903.



Question 4

Q: 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?

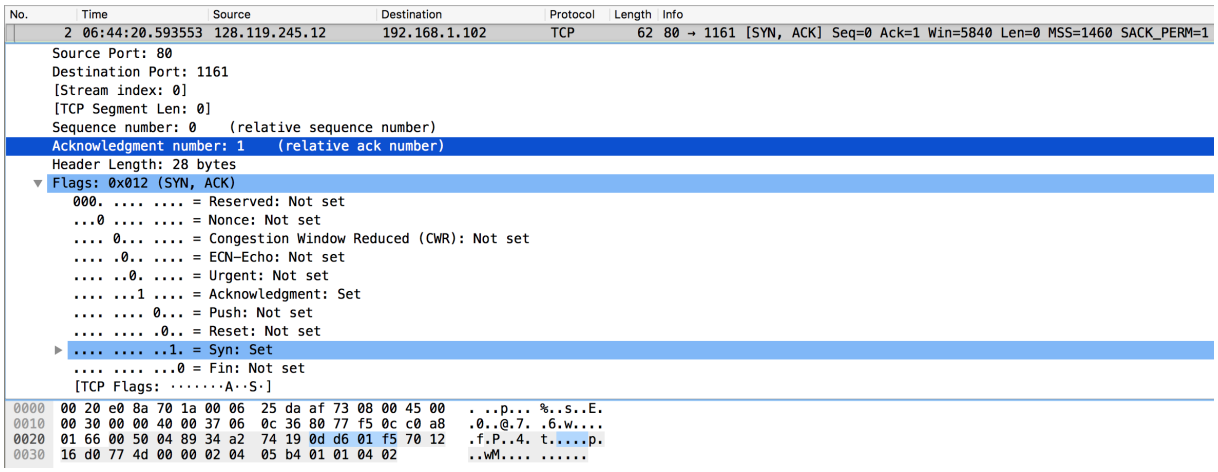
A: The relative sequence number of the TCP SYN segment is 0 (actual is 0x0dd601f4). The Flags value 0x002 identifies the segment as a SYN segment.



Question 5

Q: 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?

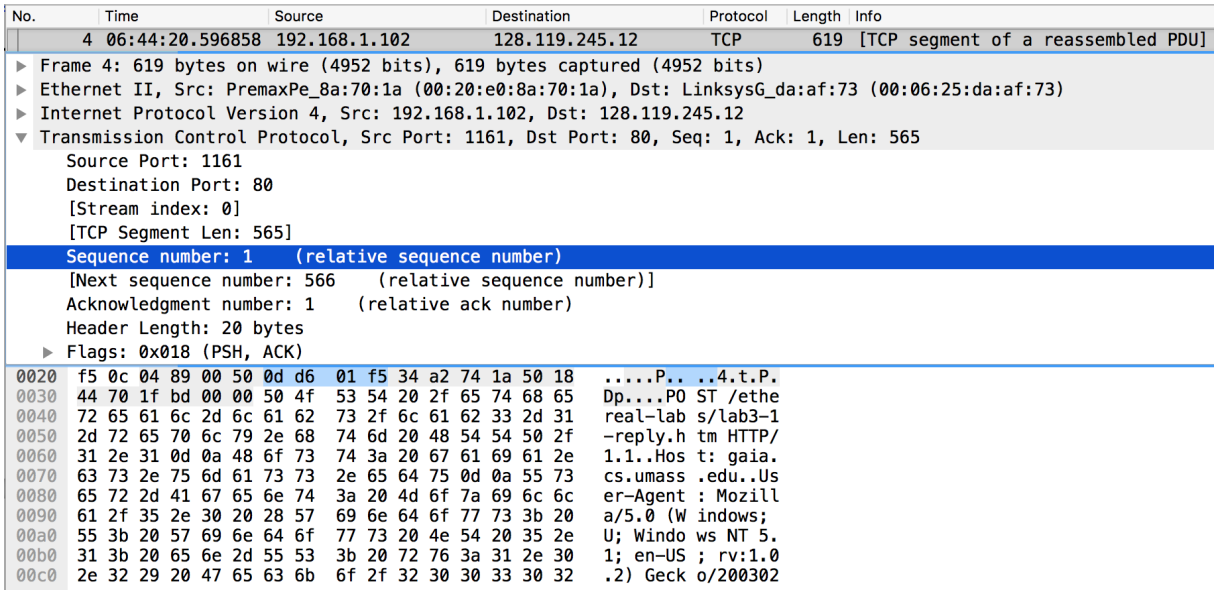
A: The relative sequence number is 0 (actual is 0x34a27419). The value of the Acknowledgement field is 1. It determines the value by adding 1 to the client's sequence number. The Flags value 0x012 identifies the segment as a SYNACK segment.



Question 6

Q: What is the sequence number of the TCP segment containing the HTTP POST command? Note that in order to find the POST command, youll 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.

A: The sequence number is 1.



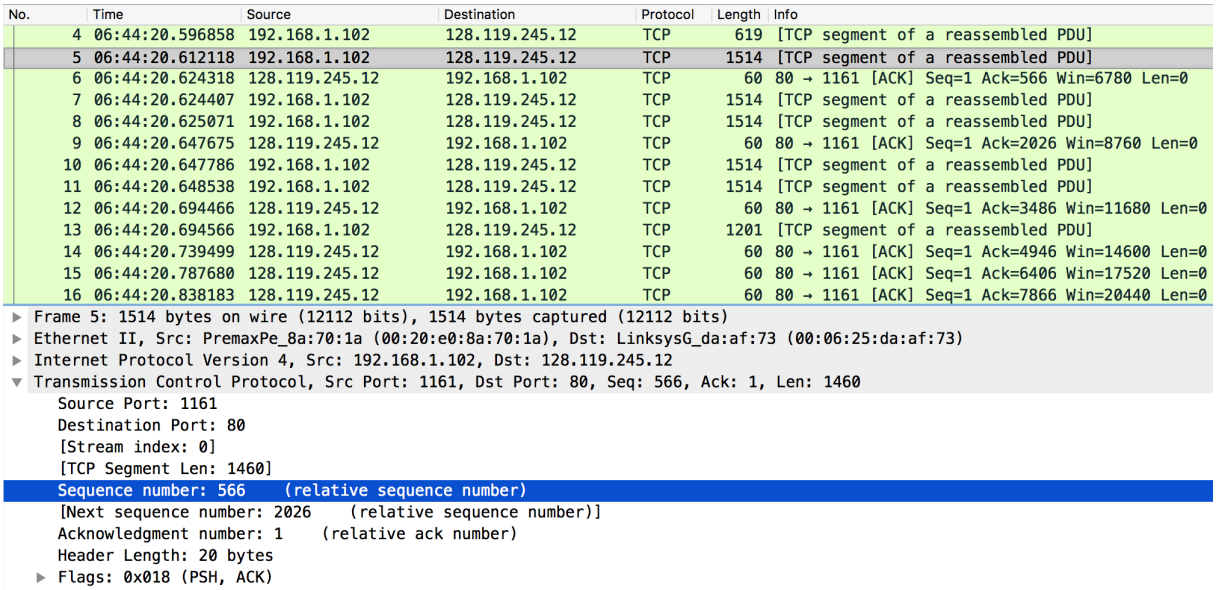
Question 7

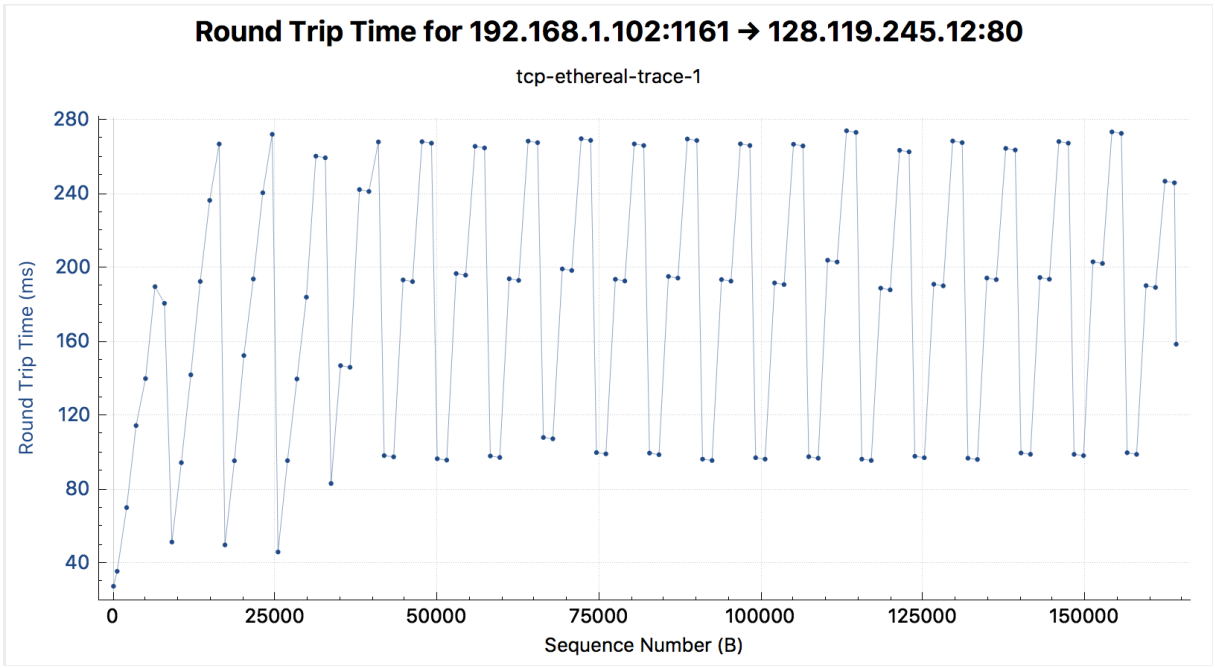
Q: Consider the TCP segment containing the HTTP POST as the first segment in the TCP connection. What are the sequence numbers of the first six segments in the TCP connection (including the segment containing the HTTP POST)? At what time was each segment sent? When was the ACK for each segment received? 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, p. 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.

A: `EstimatedRTT` was calculated using the formula:
$EstimatedRTT = (1 - \alpha) \cdot EstimatedRTT + \alpha \cdot SampleRTT$
where $\alpha = 0.125$

Segment	SEQ Num	Time Sent	ACK received	RTT value	Est. RTT value
1	1	6:44:20.596858	6:44:20.624318	27.46 ms	27.46 ms
2	566	6:44:20.612118	6:44:20.647675	35.56 ms	28.47 ms
3	2026	6:44:20.624407	6:44:20.694466	70.06 ms	33.67 ms
4	3486	6:44:20.625071	6:44:20.739499	114.43 ms	43.77 ms
5	4946	6:44:20.647786	6:44:20.787680	139.90 ms	55.78 ms
6	6406	6:44:20.648538	6:44:20.838183	189.64 ms	72.51 ms

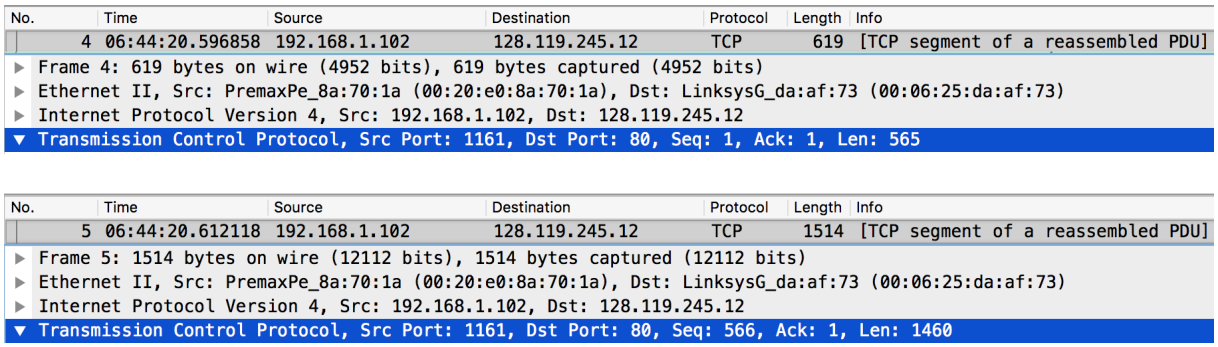




Question 8

Q: What is the length of each of the first six TCP segments?

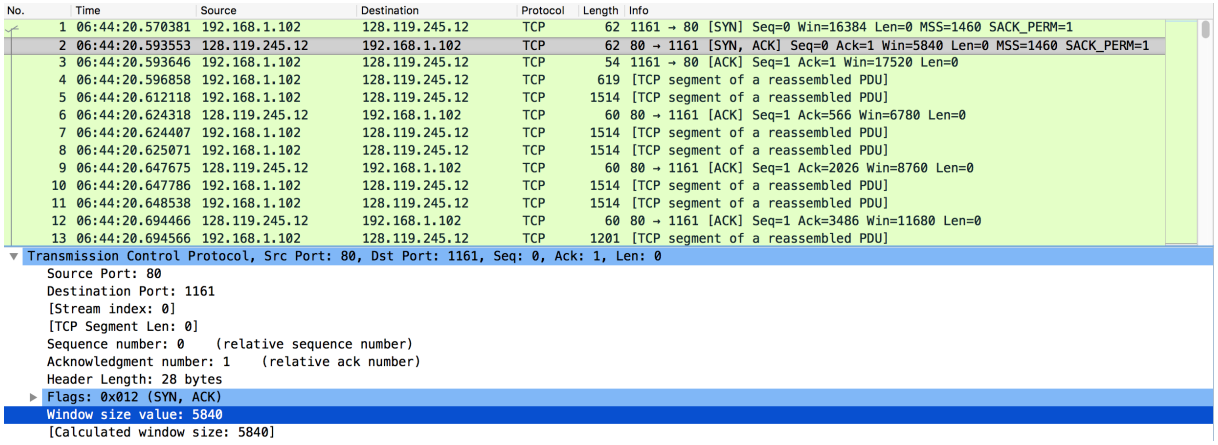
A: The length of the first segment is 565 bytes. The following five segments are each 1460 bytes.



Question 9

Q: 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: The minimum buffer space for the entire trace is 5840 bytes. It appears during the first acknowledgement sent by gaia.cs.umass.edu. There is never a lack of receiver buffer space in this trace. The TCP Window grows consistently until it reaches a max of 62,780 bytes, and doesn't dip back down. There are no instances where the TCP Window drops to zero, which would occur if the lack of buffer space was throttling the sender.

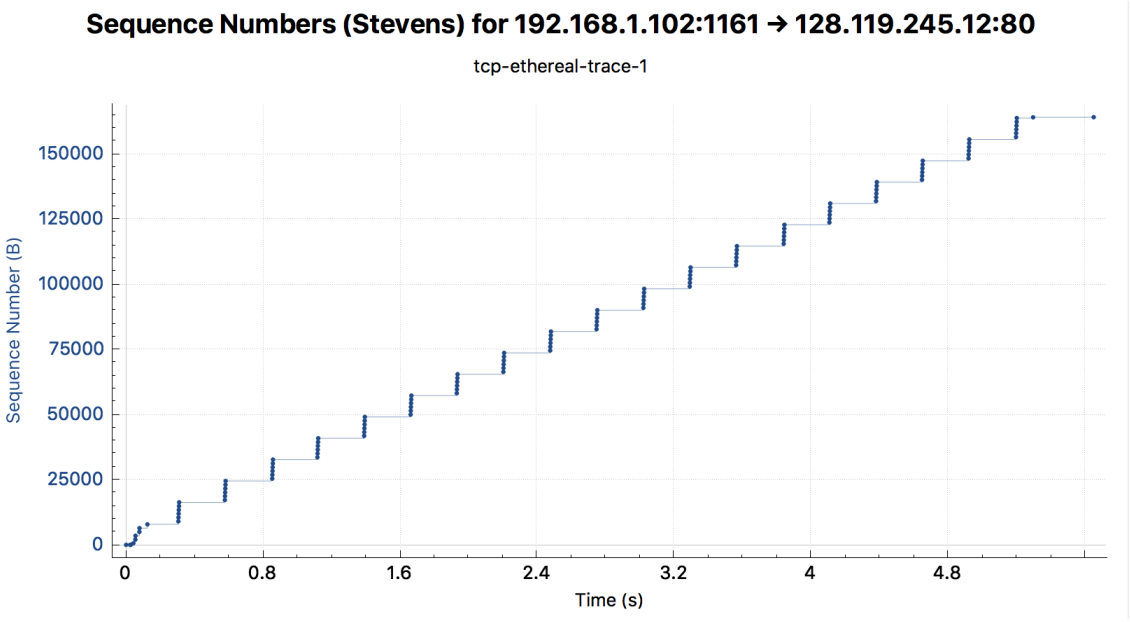


Question 10

Q: 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 aren't any retransmitted segments. We can check by looking at the each segment's data and seeing if any are sent twice. Another way we can check is by plotting the Time-Sequence

Graph (Stevens) and making sure that the sequence number increases at a steady rate over time. If there was a retransmitted packet, then the sequence number of the retransmitted packet should be noticeably smaller than the sequence numbers of the segments near it. Also, if there was a retransmitted segment, Wireshark would also make a note labelled [TCP Retransmit] in the "Info" section.



Question 11

Q: 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 p. 250 in the text).

A: The receiver typically acknowledges 1460 bytes of data in an ACK. Segments 60 and 61 ACK every other received segment. The acknowledged data in both segments is 2920, which is 1460 doubled.

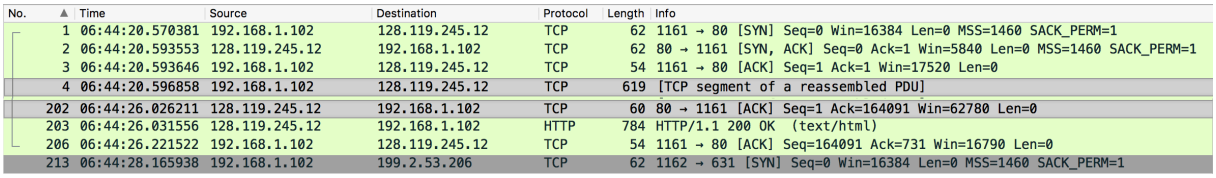
No.	Time	Source	Destination	Protocol	Length	Info
6	06:44:20.624318	128.119.245.12	192.168.1.102	TCP	60	80 → 1161 [ACK] Seq=1 Ack=566 Win=6780 Len=0
7	06:44:20.624407	192.168.1.102	128.119.245.12	TCP	1514	[TCP segment of a reassembled PDU]
8	06:44:20.625071	192.168.1.102	128.119.245.12	TCP	1514	[TCP segment of a reassembled PDU]
9	06:44:20.647675	128.119.245.12	192.168.1.102	TCP	60	80 → 1161 [ACK] Seq=1 Ack=2026 Win=8760 Len=0
10	06:44:20.647786	192.168.1.102	128.119.245.12	TCP	1514	[TCP segment of a reassembled PDU]
11	06:44:20.648538	192.168.1.102	128.119.245.12	TCP	1514	[TCP segment of a reassembled PDU]
12	06:44:20.694466	128.119.245.12	192.168.1.102	TCP	60	80 → 1161 [ACK] Seq=1 Ack=3486 Win=11680 Len=0
13	06:44:20.694566	192.168.1.102	128.119.245.12	TCP	1201	[TCP segment of a reassembled PDU]
14	06:44:20.739499	128.119.245.12	192.168.1.102	TCP	60	80 → 1161 [ACK] Seq=1 Ack=4946 Win=14600 Len=0
15	06:44:20.787680	128.119.245.12	192.168.1.102	TCP	60	80 → 1161 [ACK] Seq=1 Ack=6406 Win=17520 Len=0
16	06:44:20.838183	128.119.245.12	192.168.1.102	TCP	60	80 → 1161 [ACK] Seq=1 Ack=7866 Win=20440 Len=0
17	06:44:20.875188	128.119.245.12	192.168.1.102	TCP	60	80 → 1161 [ACK] Seq=1 Ack=9013 Win=23360 Len=0
18	06:44:20.875421	192.168.1.102	128.119.245.12	TCP	1514	[TCP segment of a reassembled PDU]
58	06:44:21.692272	192.168.1.102	128.119.245.12	TCP	946	[TCP segment of a reassembled PDU]
59	06:44:21.770802	128.119.245.12	192.168.1.102	TCP	60	80 → 1161 [ACK] Seq=1 Ack=35049 Win=62780 Len=0
60	06:44:21.835407	128.119.245.12	192.168.1.102	TCP	60	80 → 1161 [ACK] Seq=1 Ack=37969 Win=62780 Len=0
61	06:44:21.932455	128.119.245.12	192.168.1.102	TCP	60	80 → 1161 [ACK] Seq=1 Ack=40889 Win=62780 Len=0
62	06:44:21.960267	128.119.245.12	192.168.1.102	TCP	60	80 → 1161 [ACK] Seq=1 Ack=41781 Win=62780 Len=0

Num	ACK Number	Data Acknowledged
1	566	566
2	2026	1460
3	3486	1460
4	4946	1460
5	6406	1460
6	7866	1460
7	9013	1147

Question 12

Q: What is the throughput (bytes transferred per unit time) for the TCP connection?

A: The equation for throughput is total bytes transferred over the connection time.
Total bytes transferred = 164091 − 1 = 164090 bytes = 164.090 KB
Total Time = 6:44:26.026211 − 6:44:20.596858 = 5.42935 secs
Throughput = $\frac{164.090}{5.42935}$ = 30.223 kilobytes/sec



No.	Time	Source	Destination	Protocol	Length	Info
1	06:44:20.570381	192.168.1.102	128.119.245.12	TCP	62	1161 → 80 [SYN] Seq=0 Win=16384 Len=0 MSS=1460 SACK_PERM=1
2	06:44:20.593553	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=1
3	06:44:20.593646	192.168.1.102	128.119.245.12	TCP	54	1161 → 80 [ACK] Seq=1 Ack=1 Win=17520 Len=0
4	06:44:20.596858	192.168.1.102	128.119.245.12	TCP	619	[TCP segment of a reassembled PDU]
202	06:44:26.026211	128.119.245.12	192.168.1.102	TCP	60	80 → 1161 [ACK] Seq=1 Ack=164091 Win=62780 Len=0
203	06:44:26.031556	128.119.245.12	192.168.1.102	HTTP	784	HTTP/1.1 200 OK (text/html)
206	06:44:26.221522	192.168.1.102	128.119.245.12	TCP	54	1161 → 80 [ACK] Seq=164091 Ack=731 Win=16790 Len=0
213	06:44:28.165938	192.168.1.102	199.2.53.206	TCP	62	1162 → 631 [SYN] Seq=0 Win=16384 Len=0 MSS=1460 SACK_PERM=1

Question 13

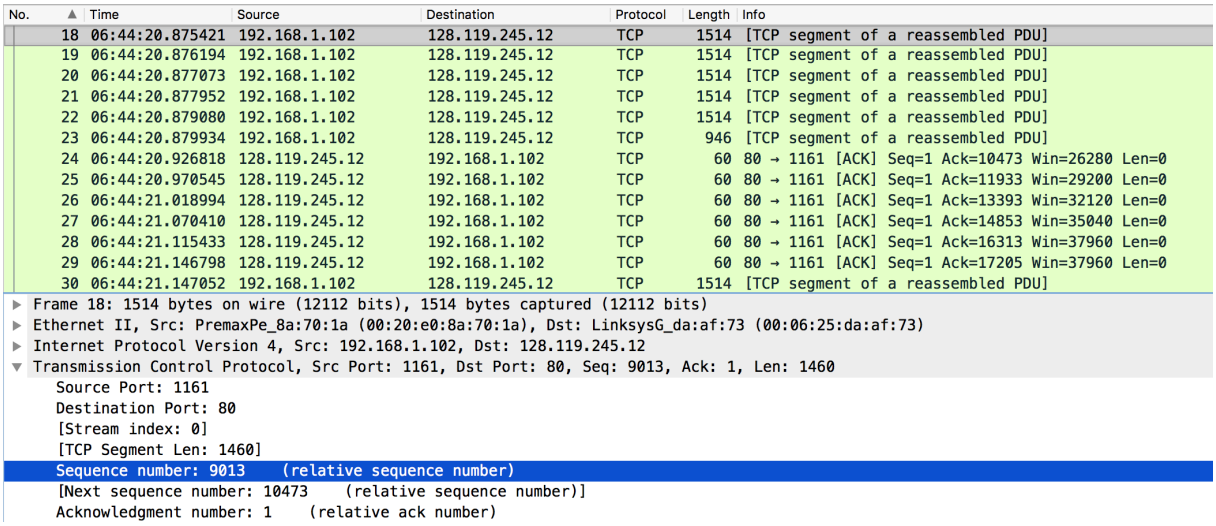
Q: Use the Time-Sequence-Graph (Stevens) plotting tool to view the sequence number versus time plot of segments being sent from the client to the gaia.cs.umass.edu server. Can you identify where TCPs slowstart phase begins and ends, and where congestion avoidance takes over? Comment on ways in which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

A: TCP's slow start begins when the connection begins (Segment 4, when the HTTP POST is sent). However, we cannot identify where TCP's slow start ends and congestion avoidance begins. Congestion avoidance occurs when there is a loss event (i.e. packets dropped, timeout) or the slow state threshold is reached. In the example trace given, the client never sends more data than the server buffer can handle. By inspecting the amount of data outstanding (sent but not yet acknowledged), we can see that the client sends out at most 8192 bytes outstanding. Before the data is acknowledged, the client temporarily stops transmitting data.

The idealized behavior of TCP is that clients should continue to increase their sending rate until they see packets drop. TCP then interprets packet drop as network congestion, and the client's transmission rate slows down. Clients periodically probe the network to check whether more bandwidth is available. The measured data differs since despite not detecting any network congestion or packet loss, the client does not continue to send data, but rather stops.

(see Question 10 for Time-Sequence Graph (Stevens))

Num	SEQ Num	ACK Num	Length	Data Outstanding
18	9013	-	1460	1460
19	10473	-	1460	2920
20	11933	-	1460	4380
21	13393	-	1460	5840
22	14853	-	1460	7300
23	16313	-	892	8192
24	-	10473	-	6732
25	-	11933	-	5272
26	-	13393	-	3812
27	-	14853	-	2352
28	-	16313	-	892
29	-	17205	-	0

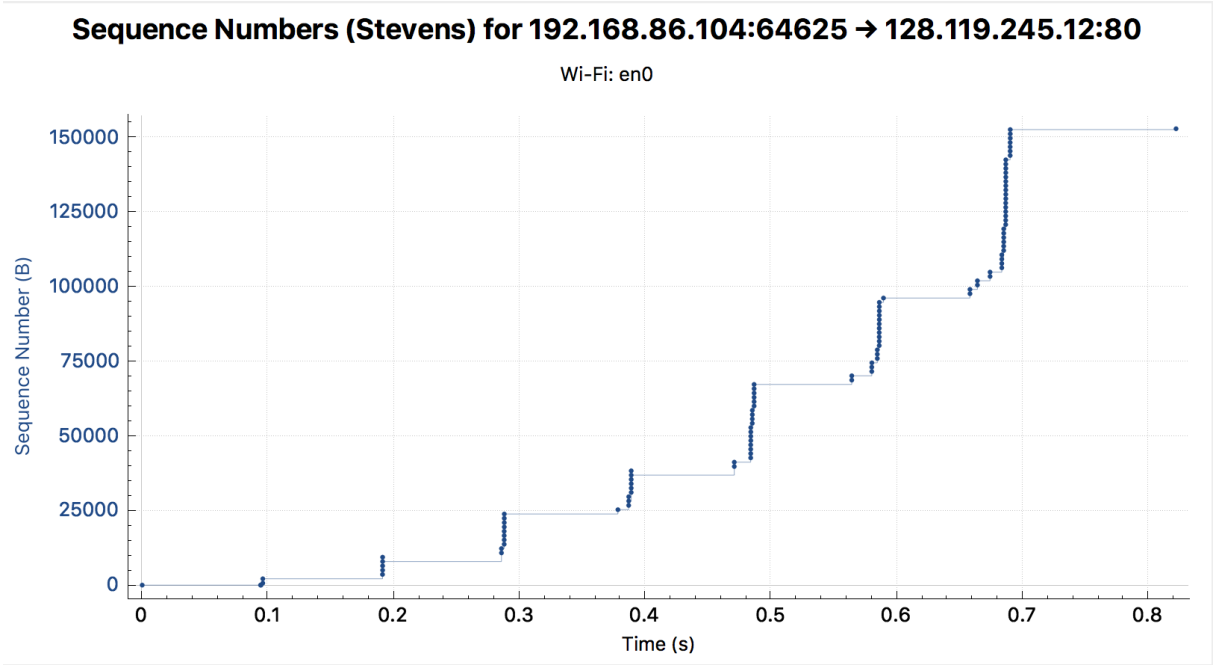


Question 14

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

A: TCP's slow start begins when the connection begins, which is Segment 17, when the HTTP POST is sent. Slow start doesn't end before the file is finished transferring. The Time-Sequence (Stevens) graph below shows that doubling of packets sent occurs until the very end of the file transfer process. The ACK statements increase by 1448 (length of each segment), meaning that one ACK statement is being sent for each packet, causing the rate of transmission to grow exponentially.

Since TCP slow start never ends, congestion avoidance doesn't begin.



No.	Time	Source	Destination	Protocol	Length	Info
12	00:52:43.044167	192.168.86.104	128.119.245.12	TCP	78	64625 → 80 [SYN] Seq=0 Win=65535 Len=0 MSS=1460 WS=32 TSval=1508859767
13	00:52:43.091726	128.119.245.12	192.168.86.104	TCP	66	80 → 64624 [ACK] Seq=1 Ack=2 Win=235 Len=0 TSval=2466138598 TSecr=1508
15	00:52:43.138146	128.119.245.12	192.168.86.104	TCP	74	80 → 64625 [SYN, ACK] Seq=0 Ack=1 Win=28960 Len=0 MSS=1460 SACK_PERM=1
16	00:52:43.138221	192.168.86.104	128.119.245.12	TCP	66	64625 → 80 [ACK] Seq=1 Ack=1 Win=131744 Len=0 TSval=1508859860 TSecr=2
17	00:52:43.138672	192.168.86.104	128.119.245.12	TCP	733	[TCP segment of a reassembled PDU]
18	00:52:43.140045	192.168.86.104	128.119.245.12	TCP	1514	[TCP segment of a reassembled PDU]
19	00:52:43.140047	192.168.86.104	128.119.245.12	TCP	1514	[TCP segment of a reassembled PDU]
20	00:52:43.235122	128.119.245.12	192.168.86.104	TCP	66	80 → 64625 [ACK] Seq=1 Ack=668 Win=30336 Len=0 TSval=2466138742 TSecr=
21	00:52:43.235128	128.119.245.12	192.168.86.104	TCP	66	80 → 64625 [ACK] Seq=1 Ack=2116 Win=33280 Len=0 TSval=2466138742 TSecr
22	00:52:43.235129	128.119.245.12	192.168.86.104	TCP	66	80 → 64625 [ACK] Seq=1 Ack=3564 Win=36096 Len=0 TSval=2466138742 TSecr
23	00:52:43.235228	192.168.86.104	128.119.245.12	TCP	1514	[TCP segment of a reassembled PDU]
24	00:52:43.235314	192.168.86.104	128.119.245.12	TCP	1514	[TCP segment of a reassembled PDU]
25	00:52:43.235315	192.168.86.104	128.119.245.12	TCP	1514	[TCP segment of a reassembled PDU]
26	00:52:43.235317	192.168.86.104	128.119.245.12	TCP	1514	[TCP segment of a reassembled PDU]
27	00:52:43.235317	192.168.86.104	128.119.245.12	TCP	1514	[TCP segment of a reassembled PDU]
28	00:52:43.329737	128.119.245.12	192.168.86.104	TCP	66	80 → 64625 [ACK] Seq=1 Ack=5012 Win=39040 Len=0 TSval=2466138837 TSecr
29	00:52:43.329809	192.168.86.104	128.119.245.12	TCP	1514	[TCP segment of a reassembled PDU]
30	00:52:43.329810	192.168.86.104	128.119.245.12	TCP	1514	[TCP segment of a reassembled PDU]
31	00:52:43.331872	128.119.245.12	192.168.86.104	TCP	66	80 → 64625 [ACK] Seq=1 Ack=6460 Win=41984 Len=0 TSval=2466138837 TSecr
32	00:52:43.331878	128.119.245.12	192.168.86.104	TCP	66	80 → 64625 [ACK] Seq=1 Ack=7908 Win=44800 Len=0 TSval=2466138838 TSecr
33	00:52:43.331879	128.119.245.12	192.168.86.104	TCP	66	80 → 64625 [ACK] Seq=1 Ack=9356 Win=47744 Len=0 TSval=2466138838 TSecr
34	00:52:43.331880	128.119.245.12	192.168.86.104	TCP	66	80 → 64625 [ACK] Seq=1 Ack=10804 Win=50688 Len=0 TSval=2466138838 TSec
35	00:52:43.331968	192.168.86.104	128.119.245.12	TCP	1514	[TCP segment of a reassembled PDU]
36	00:52:43.331970	192.168.86.104	128.119.245.12	TCP	1514	[TCP segment of a reassembled PDU]
37	00:52:43.331970	192.168.86.104	128.119.245.12	TCP	1514	[TCP segment of a reassembled PDU]
38	00:52:43.331971	192.168.86.104	128.119.245.12	TCP	1514	[TCP segment of a reassembled PDU]
39	00:52:43.332043	192.168.86.104	128.119.245.12	TCP	1514	[TCP segment of a reassembled PDU]
40	00:52:43.332044	192.168.86.104	128.119.245.12	TCP	1514	[TCP segment of a reassembled PDU]
41	00:52:43.332045	192.168.86.104	128.119.245.12	TCP	1514	[TCP segment of a reassembled PDU]
42	00:52:43.332046	192.168.86.104	128.119.245.12	TCP	1514	[TCP segment of a reassembled PDU]

No.	Time	Source	Destination	Protocol	Length	Info
46	00:52:43.422314	128.119.245.12	192.168.86.104	TCP	66	80 → 64625 [ACK] Seq=1 Ack=12252 Win=53504 Len=0 TSval=2466138929 TSec
47	00:52:43.422394	192.168.86.104	128.119.245.12	TCP	1514	[TCP segment of a reassembled PDU]
48	00:52:43.430873	128.119.245.12	192.168.86.104	TCP	66	80 → 64625 [ACK] Seq=1 Ack=13700 Win=56448 Len=0 TSval=2466138937 TSec
49	00:52:43.430877	128.119.245.12	192.168.86.104	TCP	66	80 → 64625 [ACK] Seq=1 Ack=15148 Win=59264 Len=0 TSval=2466138937 TSec
50	00:52:43.430878	128.119.245.12	192.168.86.104	TCP	66	80 → 64625 [ACK] Seq=1 Ack=16596 Win=62208 Len=0 TSval=2466138937 TSec
51	00:52:43.430943	192.168.86.104	128.119.245.12	TCP	1514	[TCP segment of a reassembled PDU]
52	00:52:43.430983	192.168.86.104	128.119.245.12	TCP	1514	[TCP segment of a reassembled PDU]
53	00:52:43.430983	192.168.86.104	128.119.245.12	TCP	1514	[TCP segment of a reassembled PDU]
54	00:52:43.432781	128.119.245.12	192.168.86.104	TCP	66	80 → 64625 [ACK] Seq=1 Ack=18044 Win=65152 Len=0 TSval=2466138938 TSec
55	00:52:43.432784	128.119.245.12	192.168.86.104	TCP	66	80 → 64625 [ACK] Seq=1 Ack=19492 Win=67968 Len=0 TSval=2466138938 TSec
56	00:52:43.432785	128.119.245.12	192.168.86.104	TCP	66	80 → 64625 [ACK] Seq=1 Ack=20940 Win=70912 Len=0 TSval=2466138938 TSec
57	00:52:43.432786	128.119.245.12	192.168.86.104	TCP	66	80 → 64625 [ACK] Seq=1 Ack=22388 Win=73856 Len=0 TSval=2466138940 TSec
58	00:52:43.432786	128.119.245.12	192.168.86.104	TCP	66	80 → 64625 [ACK] Seq=1 Ack=23836 Win=76672 Len=0 TSval=2466138940 TSec
59	00:52:43.432787	128.119.245.12	192.168.86.104	TCP	66	80 → 64625 [ACK] Seq=1 Ack=25284 Win=79616 Len=0 TSval=2466138940 TSec
60	00:52:43.432975	192.168.86.104	128.119.245.12	TCP	1514	[TCP segment of a reassembled PDU]
61	00:52:43.432976	192.168.86.104	128.119.245.12	TCP	1514	[TCP segment of a reassembled PDU]
62	00:52:43.432977	192.168.86.104	128.119.245.12	TCP	1514	[TCP segment of a reassembled PDU]
63	00:52:43.432977	192.168.86.104	128.119.245.12	TCP	1514	[TCP segment of a reassembled PDU]
64	00:52:43.432978	192.168.86.104	128.119.245.12	TCP	1514	[TCP segment of a reassembled PDU]
65	00:52:43.432978	192.168.86.104	128.119.245.12	TCP	1514	[TCP segment of a reassembled PDU]
66	00:52:43.514874	128.119.245.12	192.168.86.104	TCP	66	80 → 64625 [ACK] Seq=1 Ack=26732 Win=82432 Len=0 TSval=2466139022 TSec
67	00:52:43.514928	192.168.86.104	128.119.245.12	TCP	1514	[TCP segment of a reassembled PDU]
68	00:52:43.514929	192.168.86.104	128.119.245.12	TCP	1514	[TCP segment of a reassembled PDU]
69	00:52:43.527888	128.119.245.12	192.168.86.104	TCP	66	80 → 64625 [ACK] Seq=1 Ack=28180 Win=85376 Len=0 TSval=2466139035 TSec
70	00:52:43.527894	128.119.245.12	192.168.86.104	TCP	66	80 → 64625 [ACK] Seq=1 Ack=29628 Win=88320 Len=0 TSval=2466139035 TSec
71	00:52:43.527895	128.119.245.12	192.168.86.104	TCP	66	80 → 64625 [ACK] Seq=1 Ack=31076 Win=91136 Len=0 TSval=2466139035 TSec
72	00:52:43.527896	128.119.245.12	192.168.86.104	TCP	66	80 → 64625 [ACK] Seq=1 Ack=32524 Win=94080 Len=0 TSval=2466139036 TSec
73	00:52:43.527972	192.168.86.104	128.119.245.12	TCP	1514	[TCP segment of a reassembled PDU]
74	00:52:43.527974	192.168.86.104	128.119.245.12	TCP	1514	[TCP segment of a reassembled PDU]
75	00:52:43.527975	192.168.86.104	128.119.245.12	TCP	1514	[TCP segment of a reassembled PDU]
76	00:52:43.527976	192.168.86.104	128.119.245.12	TCP	1514	[TCP segment of a reassembled PDU]