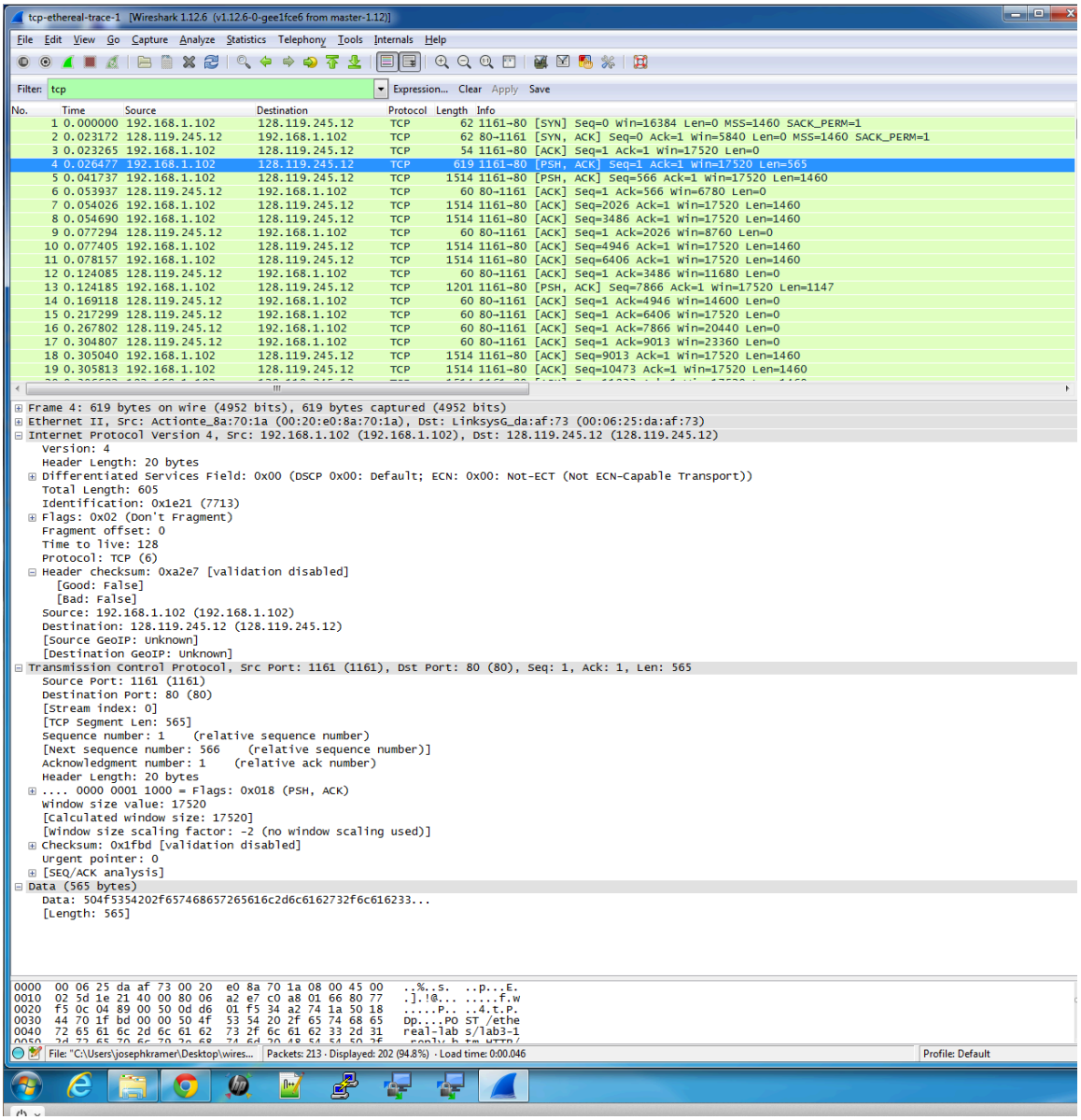
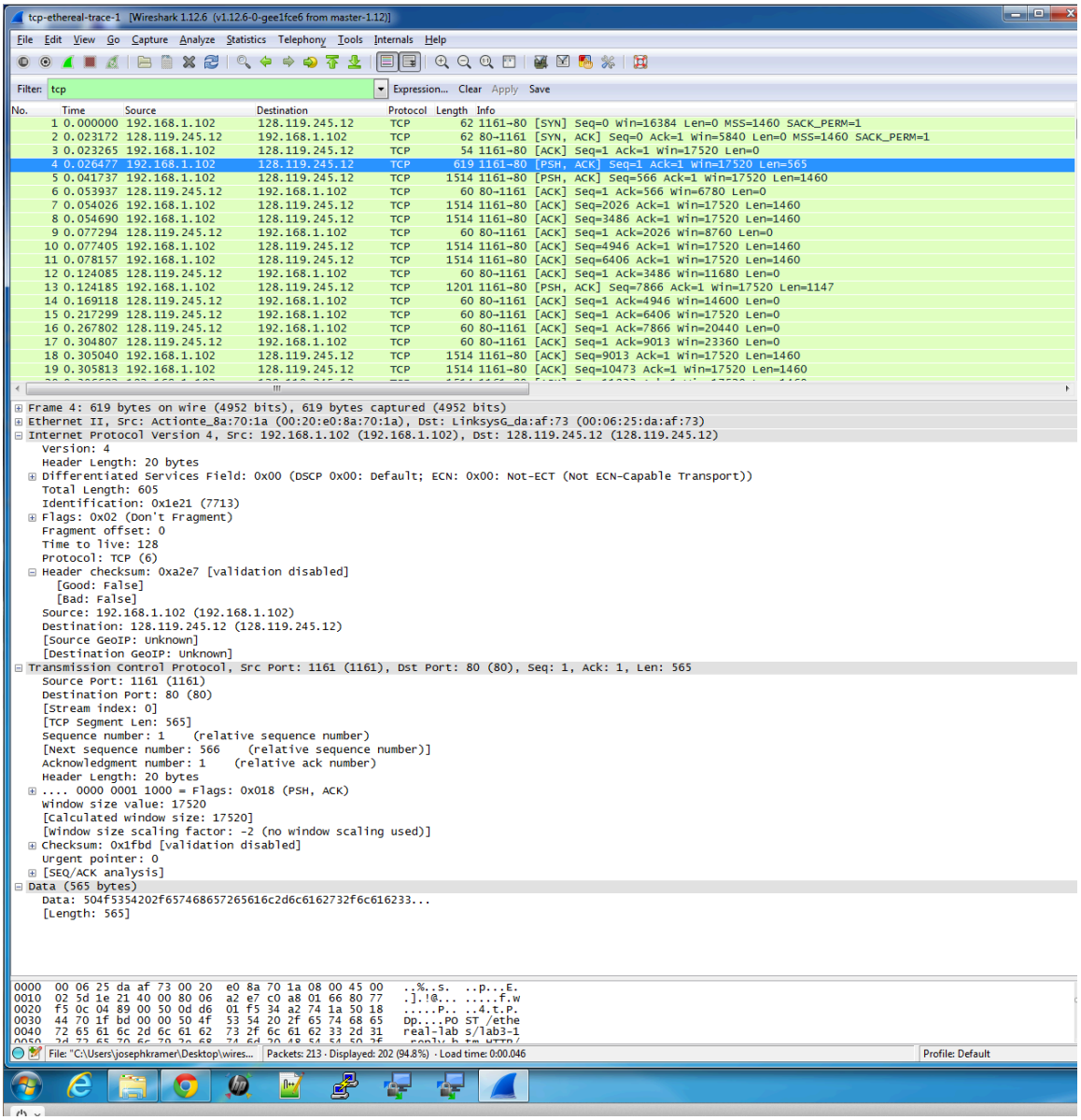
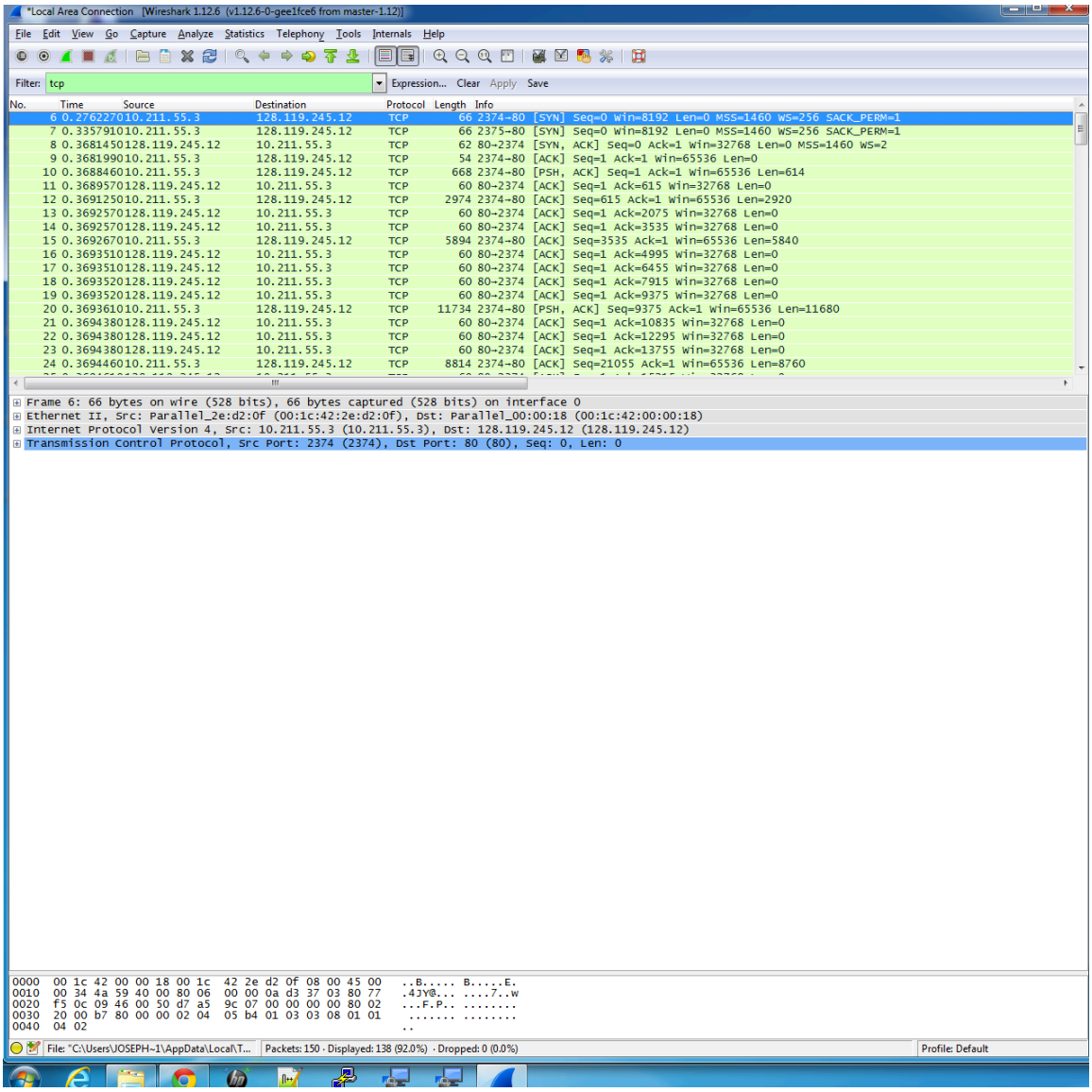
1
a)
IP Address: 192.168.1.102
TCP Port: 1161



b)
IP Address: 128.119.245.12
TCP Port: 80



c)
IP Address: 10.211.55.3
TCP Port: 2374



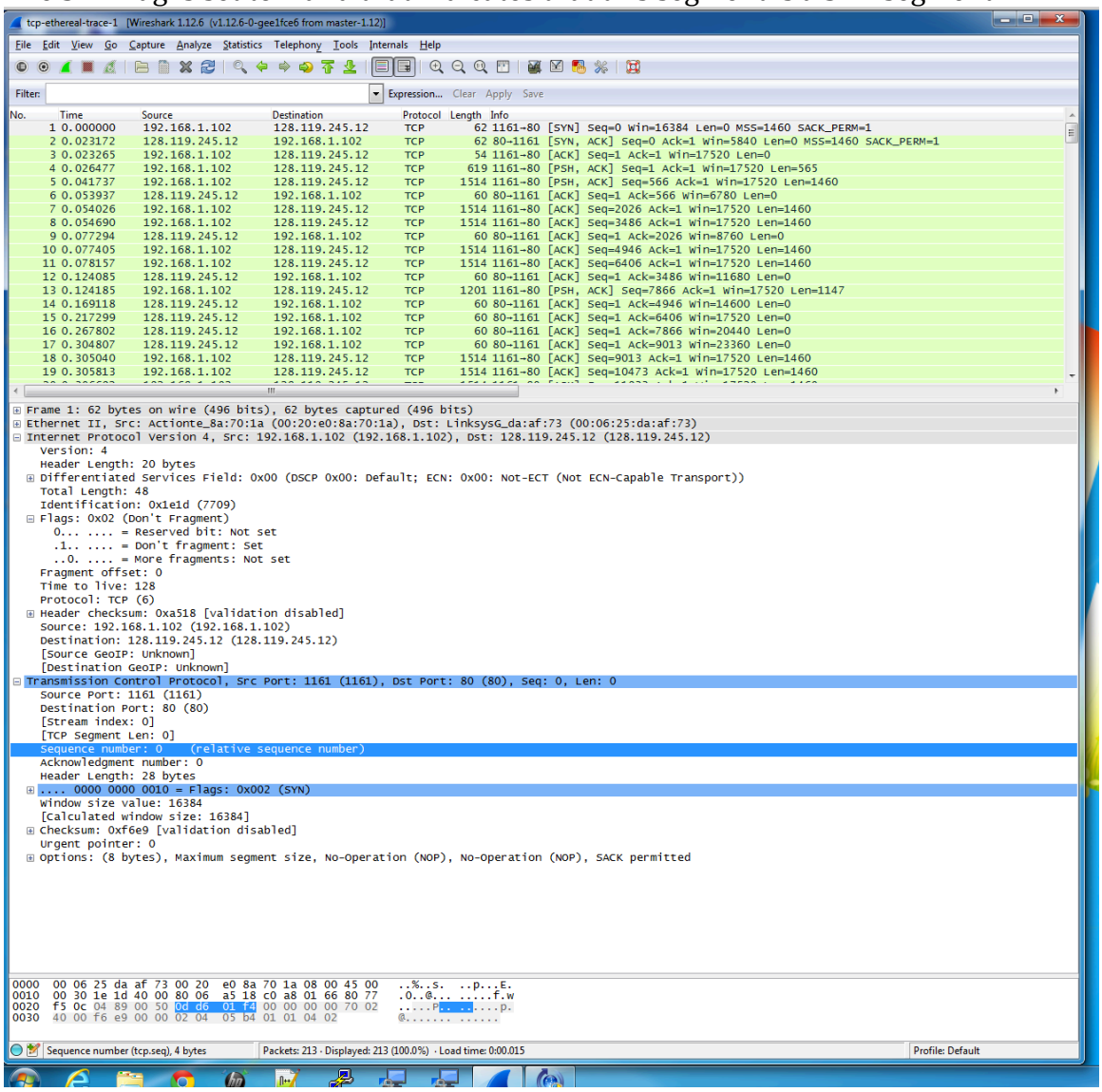
2.

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?

Sequence Number: Hex->0HDD601F4 or 232129012

What is it in the segment that identifies the segment as a SYN segment?

The SYN flag is set to 1 and that indicates that this segment is a SYN segment



3.

What is the sequence number of the SYNACK segment sent by gaia.cs.umass.edu to the client computer in reply to the SYN?

The sequence number of the SYNACK segment is: HEX->0H34A27419 or 883061785.

What is the value of the Acknowledgement field in the SYNACK segment?

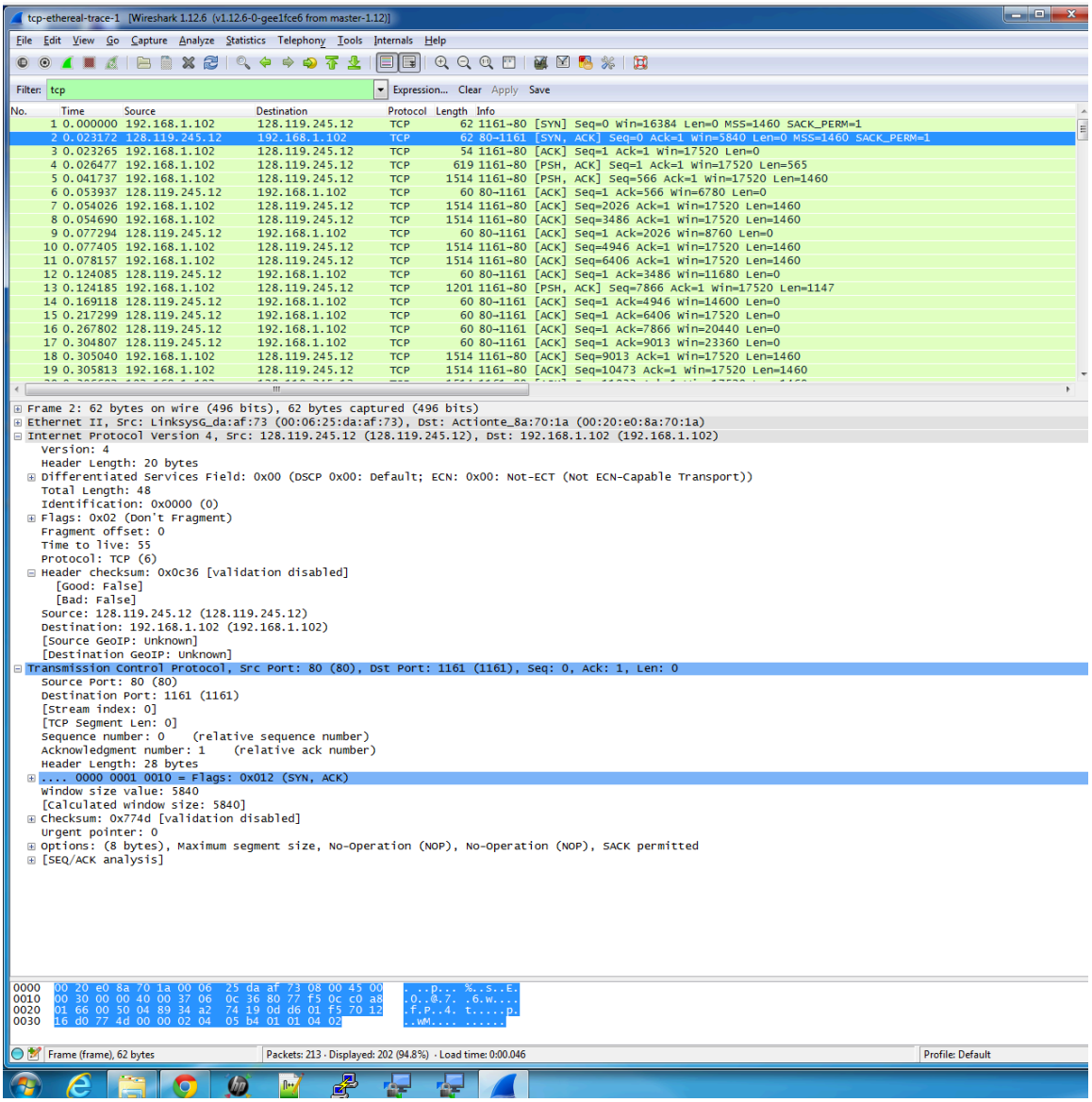
Acknowledgment number of the SYNACK segment is: HEX->DD601F5 or 232129013

How did gaia.cs.umass.edu determine that value?

The TCP receiver incremented increments the previous Sequence Number by 1, then it copies it into the acknowledgement field.

What is it in the segment that identifies the segment as a SYNACK segment?

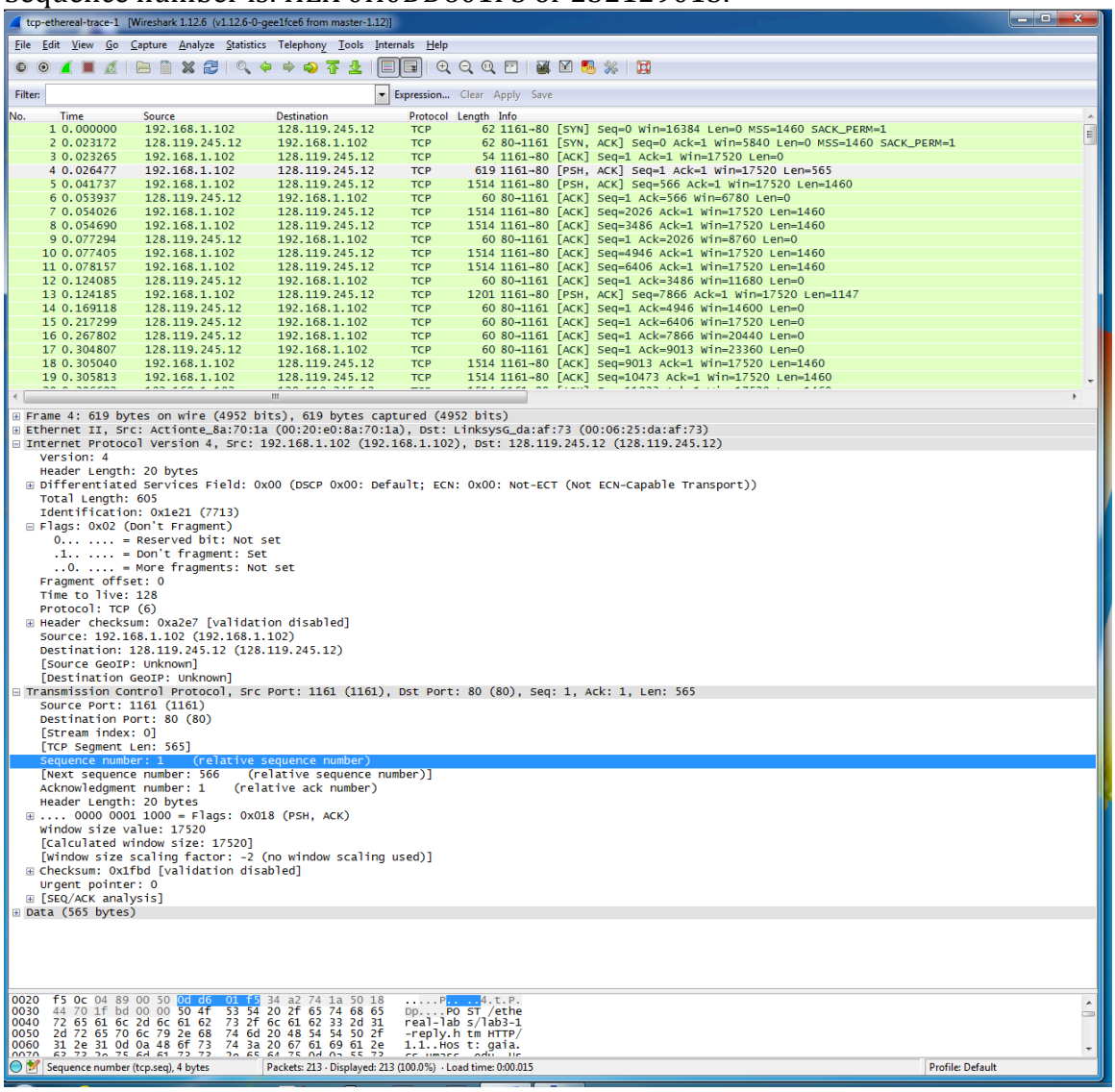
The SYN and ACK flag bits are both set to 1, this indicates that it's a SYNACK packet.



4.

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

Sequence number is: HEX 0H0DD601F5 or 232129013.



5.

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?

	Sequence #	Sent Time	ACK Received Time	RTT in Seconds
Segment 1	1	0.026477	0.053937	0.027460
Segment 2	566	0.041737	0.077294	0.035557
Segment 3	2026	0.054026	0.124085	0.070059
Segment 4	3486	0.054690	0.169118	0.114428
Segment 5	4946	0.077405	0.217299	0.139894
Segment 6	6406	0.078157	0.267802	0.189645

What is the EstimatedRTT value (see Section 3.5.3, page 239 in text) after the receipt of each ACK?

EstimatedRTT = (1-1/8) * EstimatedRTT + 1/8 * sampleRTT

EstimatedRTT = 0.02746 second

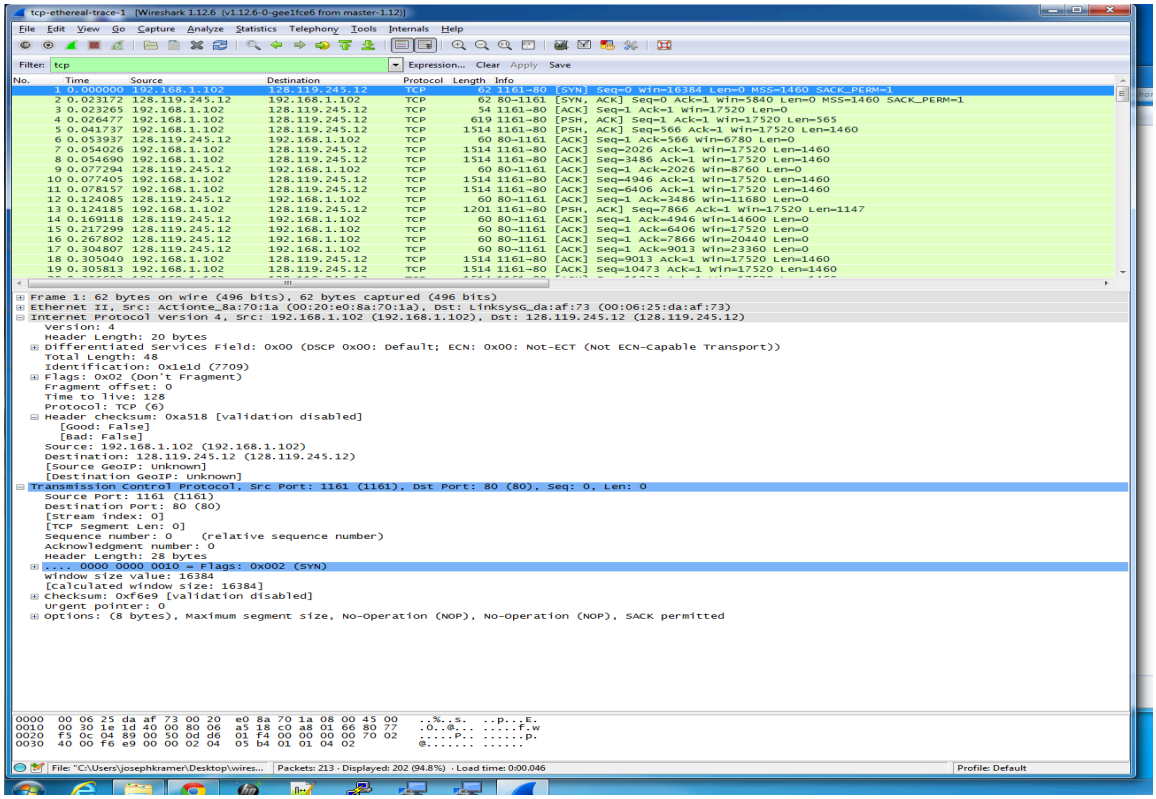
EstimatedRTT = (1-1/8) * 0.02746 + 1/8 * 0.035557 = 0.0285

EstimatedRTT = (1-1/8) * 0.0285 + 1/8 * 0.070059 = 0.0337

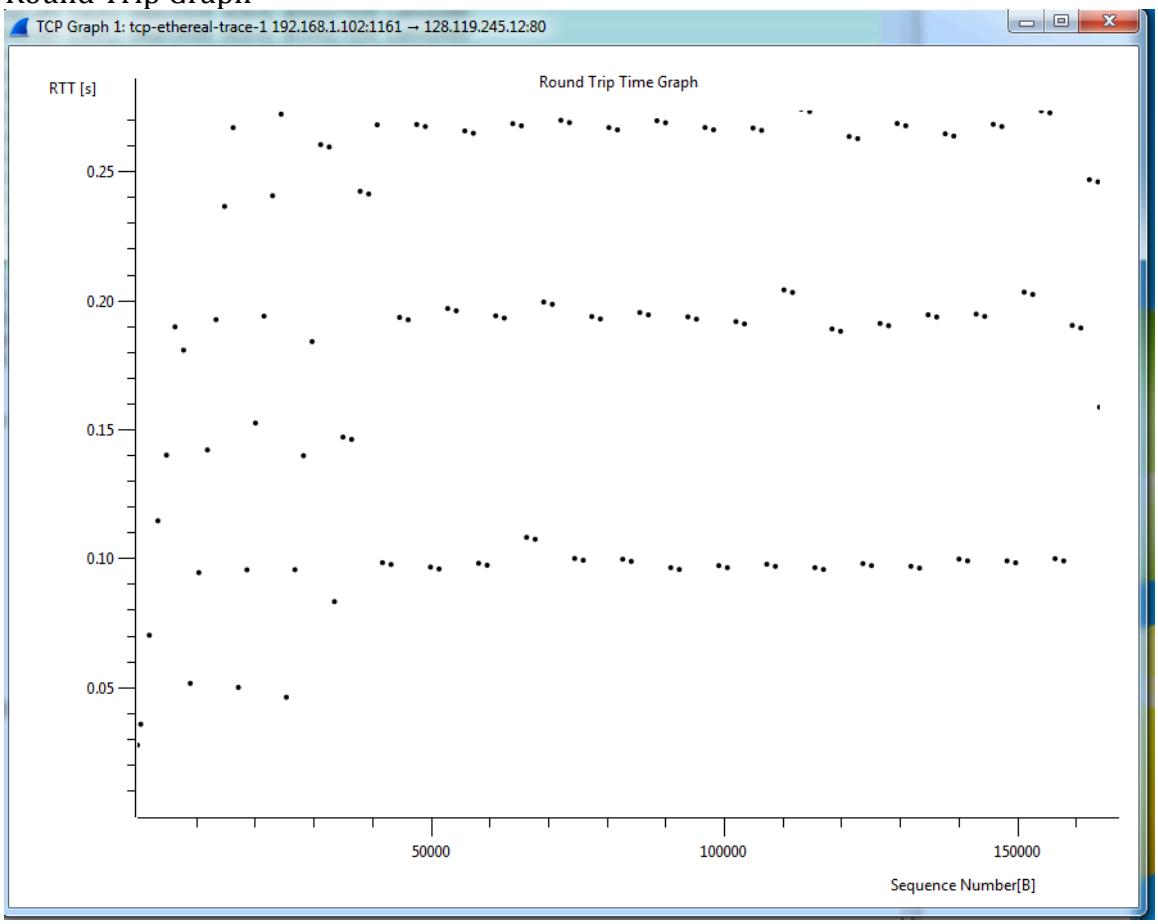
EstimatedRTT = (1-1/8) * 0.0337 + 1/8 * 0.114428 = 0.0438

EstimatedRTT = (1-1/8) * 0.0438 + 1/8 * 0.139894 = 0.0558

EstimatedRTT = (1-1/8) * 0.0558 + 1/8 * 0.189645 = 0.0725



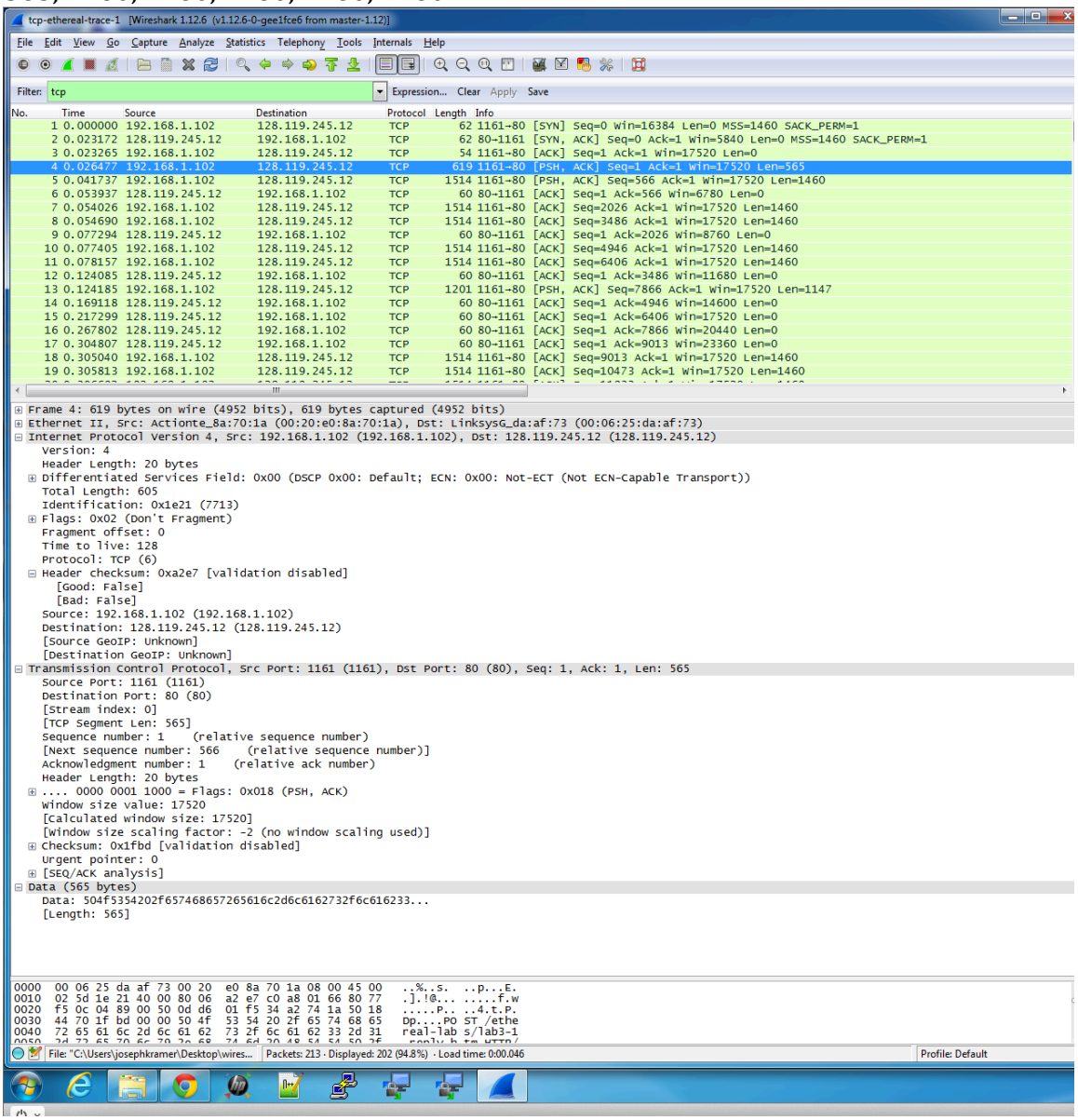
Round Trip Graph



6.

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

565, 1460, 1460, 1460, 1460, 1460



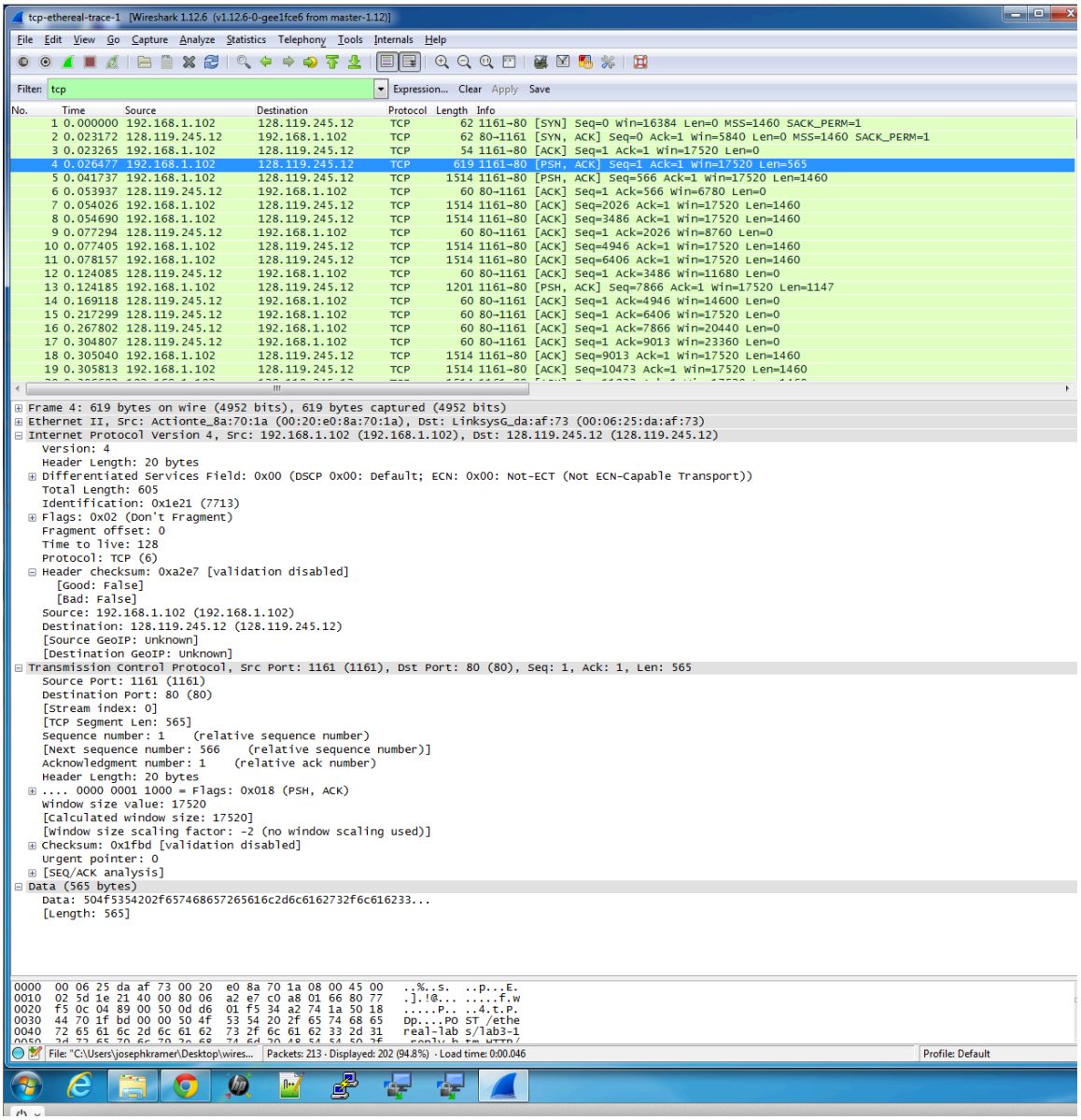
7.

What is the minimum amount of available buffer space advertised at the received for the entire trace?

Buffer: Win=5840

Does the lack of receiver buffer space ever throttle the sender?

The sender is never throttled back due to lack of receiver buffer space, because the size reaches 62780; which is 43 MSS segments. It does not appear that the lack of receiver buffer space is an issue. The sender may be more constrained by congestion then flow control.



8.

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

There are no retransmitted segments, because all sequence numbers form the source to the destination is increasing. There would need to be a repeat entry for a retransmitted segment with the same sequence number.

9.

How much data does the receiver typically acknowledge in an ACK?

1460 is the typical acknowledged data.

Can you identify cases where the receiver is ACKing every other received segment.

Yes, cases can be identified because there are times were ACK is more then usual. For examples there are several instances past segment number 60, where the receiver will send an ACK for every other received segment. At this point the receiver is sending a combined ACK for two segments. The textbook indicates that TCP's will used "delayed ACK's," this is when the receiver will wait 500ms for the arrival of another segment and then send them both back. Always look for if the data is double, because then it is ACKing every other.

10.

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

First we need to take the total data sent; which is number 2 through number 202. Now at 202 the total data sent is 164091. Then we need to subtract 1. Therefore we are left with 164090.

Now we do the same with time. We start at 4 and go to 202. The time at 2 is 0.026477 and at 202 it is 5.455830. Therefor we have 5. 455830 – 0.026477 = 5.429353

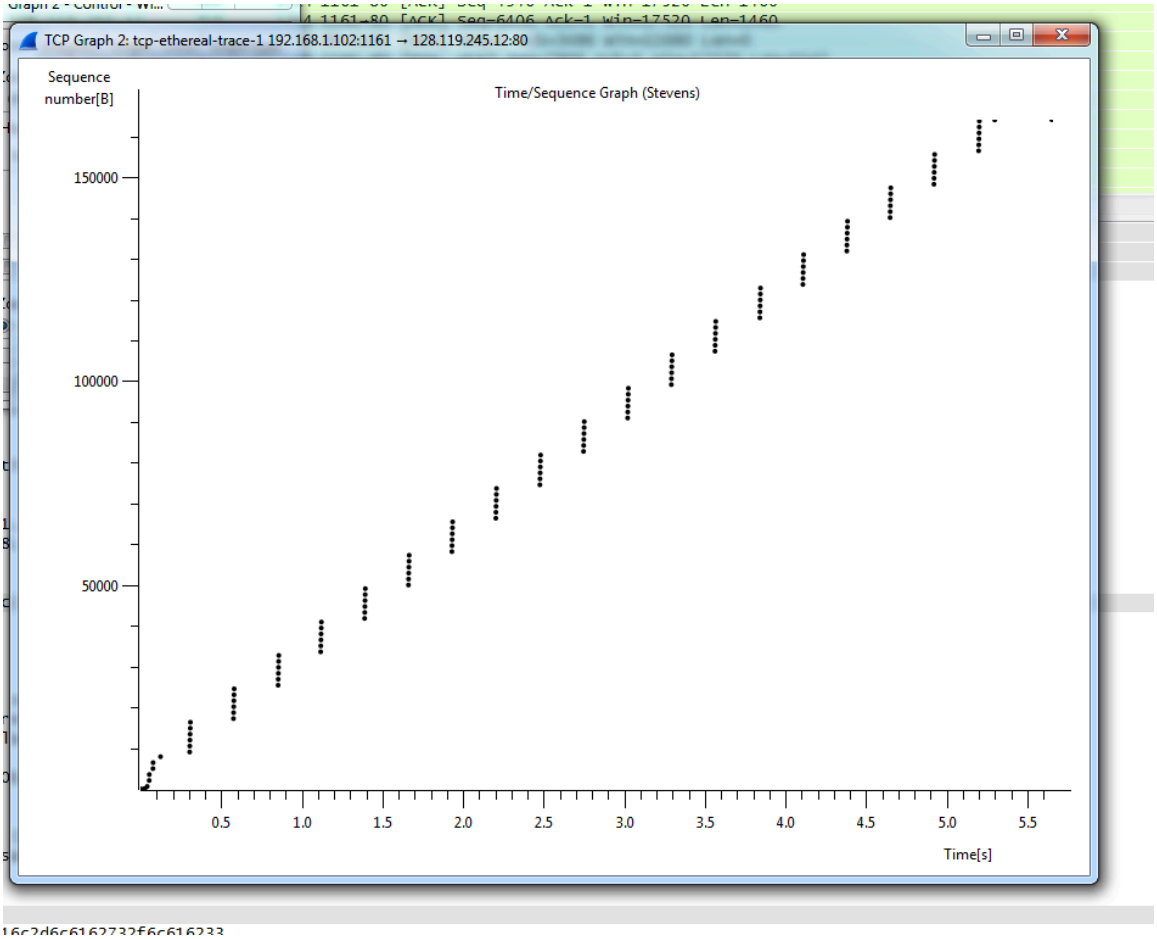
Now to calculate the throughput. 164090/5.429353 = 30222.75 bytes

11.

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.

It appears that TCP slowstart begins when the HTTP POST is sent out. The congestion window size is based on the TCP congestion avoidance and slow start phase. It does not appear that the congestion window size can be retrieved form the Time Sequence Graph. After review the graph it appears that the slow start phase last for 0.1 seconds and after that it is always operating in congestion avoidance.
I cannot see that the buffer is an issue, because the WIN is 5840 and that is not reached.
It does not appear the TCP is sending data at a state that would engage congestion avoidance. The largest data block the application sends out is 8192 bytes, before it receives and ACK for the bytes, it will not send anymore. Therefore the application will temporally stops transmission and this happens before the end of a slow start phase. It does not seem possible to precisely determine the slow start phase or the congestion avoidance phase. According to the graph it does appear to transmit packets in batches of 5; however this is not a flow control issue, because the receiver window is larger than 5 packets

How does this data differ from the book?
In this lab the file was small and it never got out of the slow start phase. The text would indicate that senders send out data as fast as possible all the time and that does not appear to be the case. I also think the TCP behavior depends on that type of application it is being used for. Since this was a small program, handling congestion was not a concern; however sending an HD movie would face congestion issues.

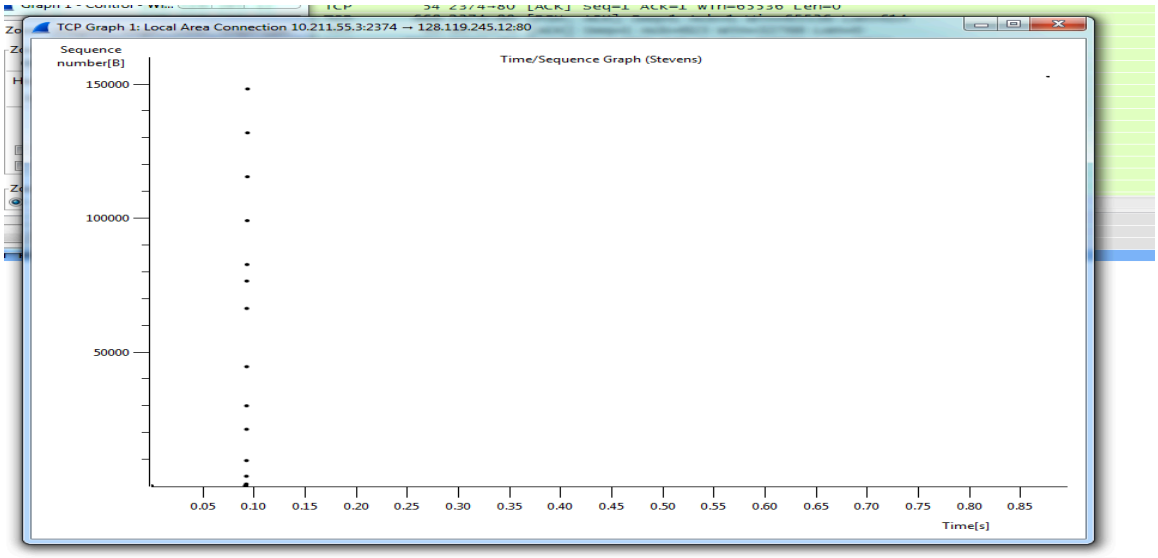

TCP Graph 2: tcp-ethereal-trace-1 192.168.1.102:1161 → 128.119.245.12:80
Sequence number[B]
Time/Sequence Graph (Stevens)
150000
100000
50000
0.5
1.0
1.5
2.0
2.5
3.0
3.5
4.0
4.5
5.0
5.5
Time[s]

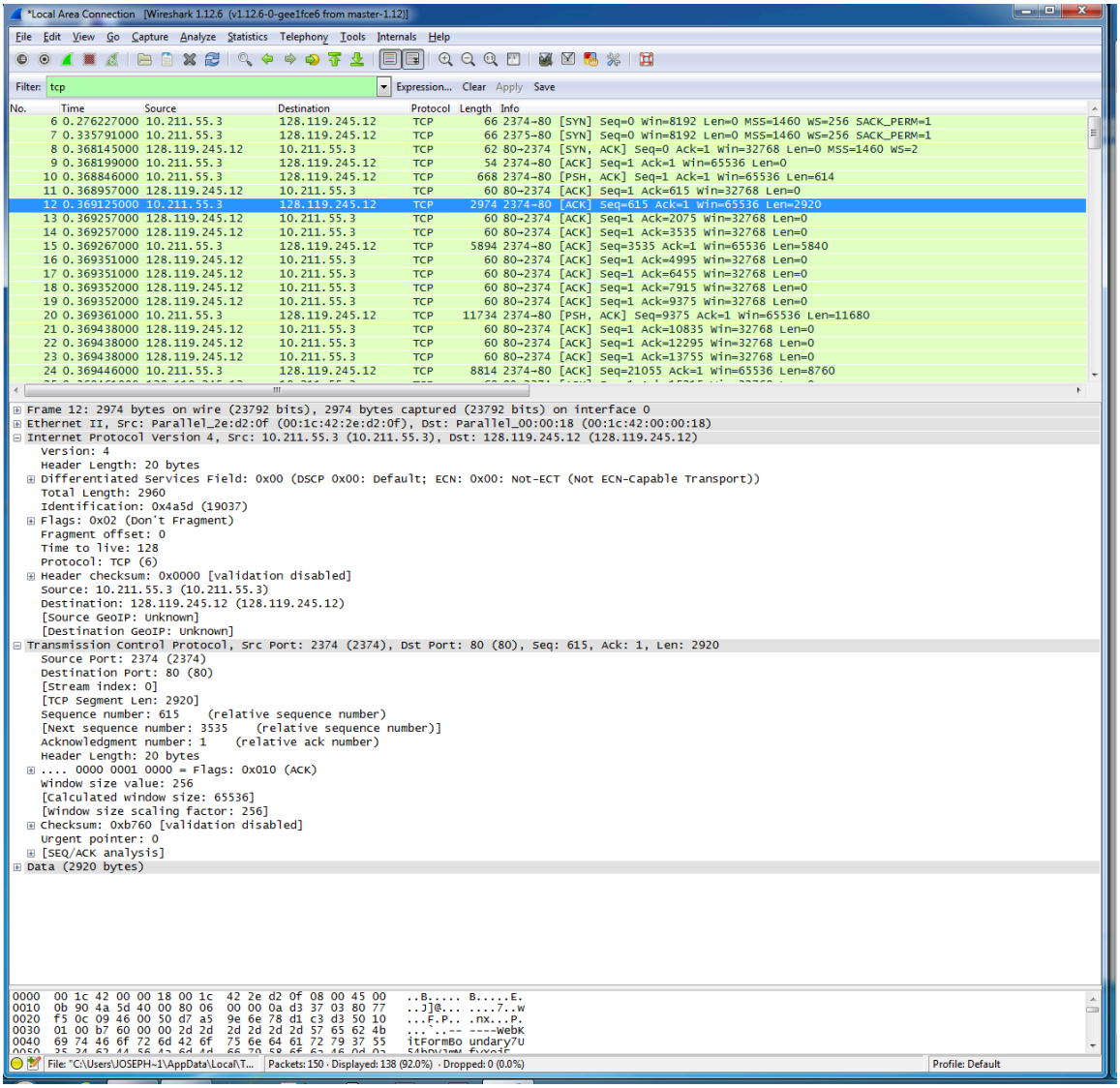
12.

Answer each of two questions in Question 11 for the trace that you have gathered when you transferred a file from your computer to gaia.cs.umass.edu

The Time Sequence Graph is much different. (I followed the direction exactly, however I'm running this on Parallels and maybe that makes a difference.) However it does not seem like the congestion wind size can be determined by this graph either. Here it does not appear that the buffer become an issue, because the WIN is 65536 and that is not reached. Also congestion avoidance did not become an issue, because the data plots are in a verticle line. Same as question 11, I cannot get the exact slow start or congestion avoidance phase.

It differs from the book the same as question 11. The textbook seems to deal with scenarios where the sender sends out data, as fast as possible, and in this lab the data does not appear to operate like that.




*Local Area Connection [Wireshark 1.12.6 (v1.12.6-0-gee1fce6 from master-1.12)]
File Edit View Go Capture Analyze Statistics Telephony Tools Internals Help
Filter: tcp
Expression... Clear Apply Save
No. Time Source Destination Protocol Length Info
6 0.276227000 10.211.55.3 128.119.245.12 TCP 66 2374→80 [SYN] Seq=0 Win=8192 Len=0 MSS=1460 WS=256 SACK_PERM=1
7 0.335791000 10.211.55.3 128.119.245.12 TCP 66 2375→80 [SYN] Seq=0 Win=8192 Len=0 MSS=1460 WS=256 SACK_PERM=1
8 0.368145000 128.119.245.12 10.211.55.3 TCP 62 80→2374 [SYN, ACK] Seq=0 Ack=1 Win=32768 Len=0 MSS=1460 WS=2
9 0.368199000 10.211.55.3 128.119.245.12 TCP 54 2374→80 [ACK] Seq=1 Ack=1 Win=65536 Len=0
10 0.368846000 10.211.55.3 128.119.245.12 TCP 668 2374→80 [PSH, ACK] Seq=1 Ack=1 Win=65536 Len=614
11 0.368957000 128.119.245.12 10.211.55.3 TCP 60 80→2374 [ACK] Seq=1 Ack=615 Win=32768 Len=0
12 0.369125000 10.211.55.3 128.119.245.12 TCP 2974 2374→80 [ACK] Seq=615 Ack=1 Win=65536 Len=2920
13 0.369257000 128.119.245.12 10.211.55.3 TCP 60 80→2374 [ACK] Seq=1 Ack=2075 Win=32768 Len=0
14 0.369257000 128.119.245.12 10.211.55.3 TCP 60 80→2374 [ACK] Seq=1 Ack=3535 Win=32768 Len=0
15 0.369267000 10.211.55.3 128.119.245.12 TCP 5894 2374→80 [ACK] Seq=3535 Ack=1 Win=65536 Len=5840
16 0.369351000 128.119.245.12 10.211.55.3 TCP 60 80→2374 [ACK] Seq=1 Ack=4995 Win=32768 Len=0
17 0.369351000 128.119.245.12 10.211.55.3 TCP 60 80→2374 [ACK] Seq=1 Ack=6455 Win=32768 Len=0
18 0.369352000 128.119.245.12 10.211.55.3 TCP 60 80→2374 [ACK] Seq=1 Ack=7915 Win=32768 Len=0
19 0.369352000 128.119.245.12 10.211.55.3 TCP 60 80→2374 [ACK] Seq=1 Ack=9375 Win=32768 Len=0
20 0.369361000 10.211.55.3 128.119.245.12 TCP 11734 2374→80 [PSH, ACK] Seq=9375 Ack=1 Win=65536 Len=11680
21 0.369438000 128.119.245.12 10.211.55.3 TCP 60 80→2374 [ACK] Seq=1 Ack=10835 Win=32768 Len=0
22 0.369438000 128.119.245.12 10.211.55.3 TCP 60 80→2374 [ACK] Seq=1 Ack=12295 Win=32768 Len=0
23 0.369438000 128.119.245.12 10.211.55.3 TCP 60 80→2374 [ACK] Seq=1 Ack=13755 Win=32768 Len=0
24 0.369446000 10.211.55.3 128.119.245.12 TCP 8814 2374→80 [ACK] Seq=21055 Ack=1 Win=65536 Len=8760
Frame 12: 2974 bytes on wire (23792 bits), 2974 bytes captured (23792 bits) on interface 0
Ethernet II, Src: Parallel_2e:d2:0f (00:1c:42:2e:d2:0f), Dst: Parallel_00:00:18 (00:1c:42:00:00:18)
Internet Protocol Version 4, Src: 10.211.55.3 (10.211.55.3), Dst: 128.119.245.12 (128.119.245.12)
Version: 4
Header Length: 20 bytes
Differentiated Services Field: 0x00 (DSCP 0x00: Default; ECN: 0x00: Not-ECT (Not ECN-Capable Transport))
Total Length: 2960
Identification: 0x4a5d (19037)
Flags: 0x02 (Don't Fragment)
Fragment offset: 0
Time to live: 128
Protocol: TCP (6)
Header checksum: 0x0000 [validation disabled]
Source: 10.211.55.3 (10.211.55.3)
Destination: 128.119.245.12 (128.119.245.12)
[Source GeoIP: Unknown]
[Destination GeoIP: Unknown]
Transmission Control Protocol, Src Port: 2374 (2374), Dst Port: 80 (80), Seq: 615, Ack: 1, Len: 2920
Source Port: 2374 (2374)
Destination Port: 80 (80)
[Stream index: 0]
[TCP Segment Len: 2920]
Sequence number: 615 (relative sequence number)
[Next sequence number: 3535 (relative sequence number)]
Acknowledgment number: 1 (relative ack number)
Header Length: 20 bytes
.... 0000 0001 0000 = Flags: 0x010 (ACK)
Window size value: 256
[Calculated window size: 65536]
[window size scaling factor: 256]
Checksum: 0xb760 [validation disabled]
Urgent pointer: 0
[SEQ/ACK analysis]
Data (2920 bytes)
0000 00 1c 42 00 00 18 00 1c 42 2e d2 0f 08 00 45 00 ..B..... B.....E.
0010 0b 90 4a 5d 40 00 80 06 00 00 0a d3 37 03 80 77 ..J]@... ....7..w
0020 f5 0c 09 46 00 50 d7 a5 9e 6e 78 d1 c3 d3 50 10 ...F.P.. .nx...P.
0030 01 00 b7 60 00 00 2d 2d 2d 2d 57 65 62 4b ...`..-- ----WebK
0040 69 74 46 6f 72 6d 42 6f 75 6e 64 61 72 79 37 55 itFormBo undary7U
File: "C:\Users\JOSEPH~1\AppData\Local\T... Packets: 150 · Displayed: 138 (92.0%) · Dropped: 0 (0.0%) Profile: Default