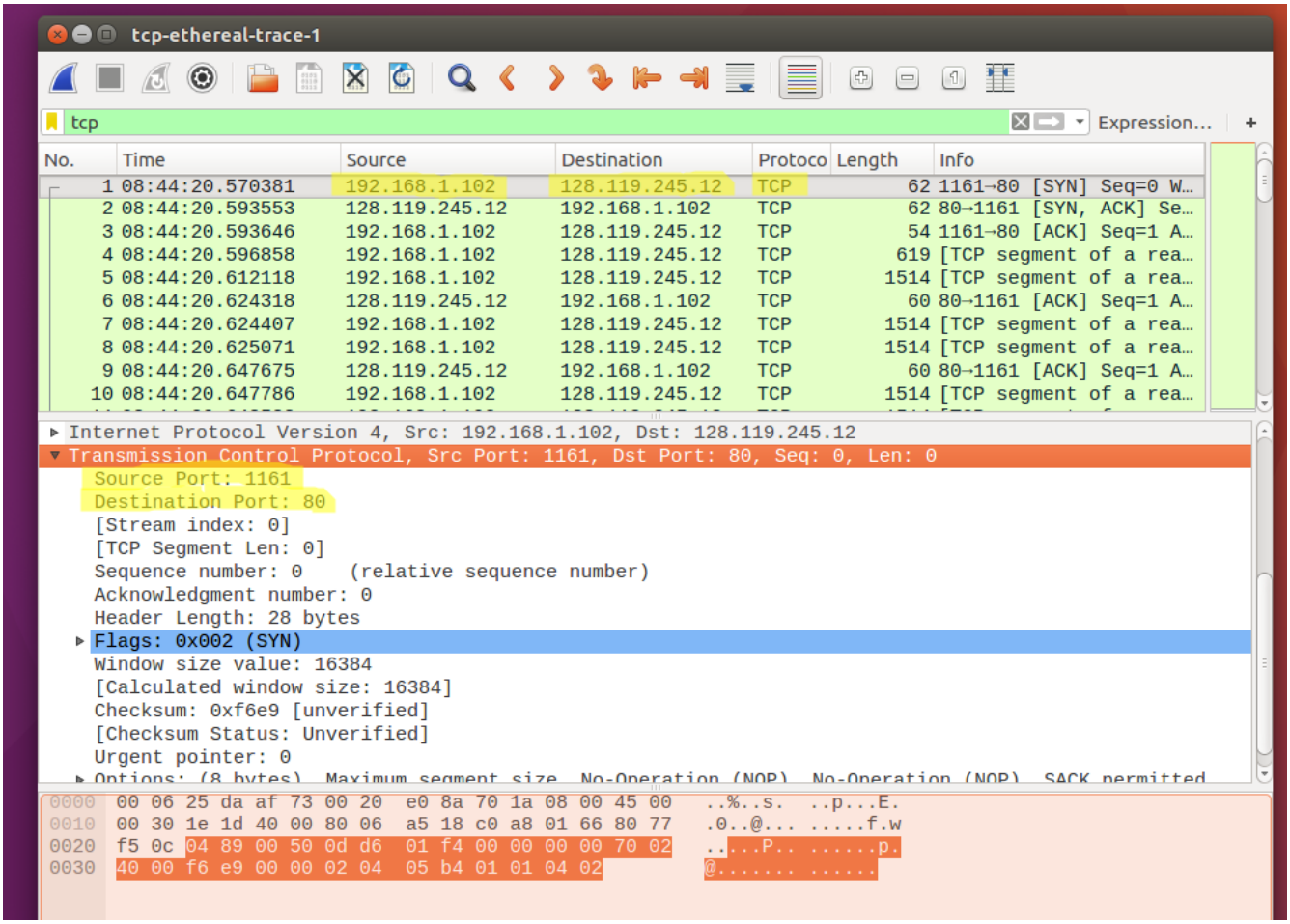
1. What is the IP address and TCP port number used by the client computer (source) that is transferring the file to gaia.cs.umass.edu? To answer this question, it's probably easiest to select an HTTP message and explore the details of the TCP packet used to carry this HTTP message, using the "details of the selected packet header window" (refer to Figure 2 in the "Getting Started with Wireshark" Lab if you're uncertain about the Wireshark windows.
ANS:

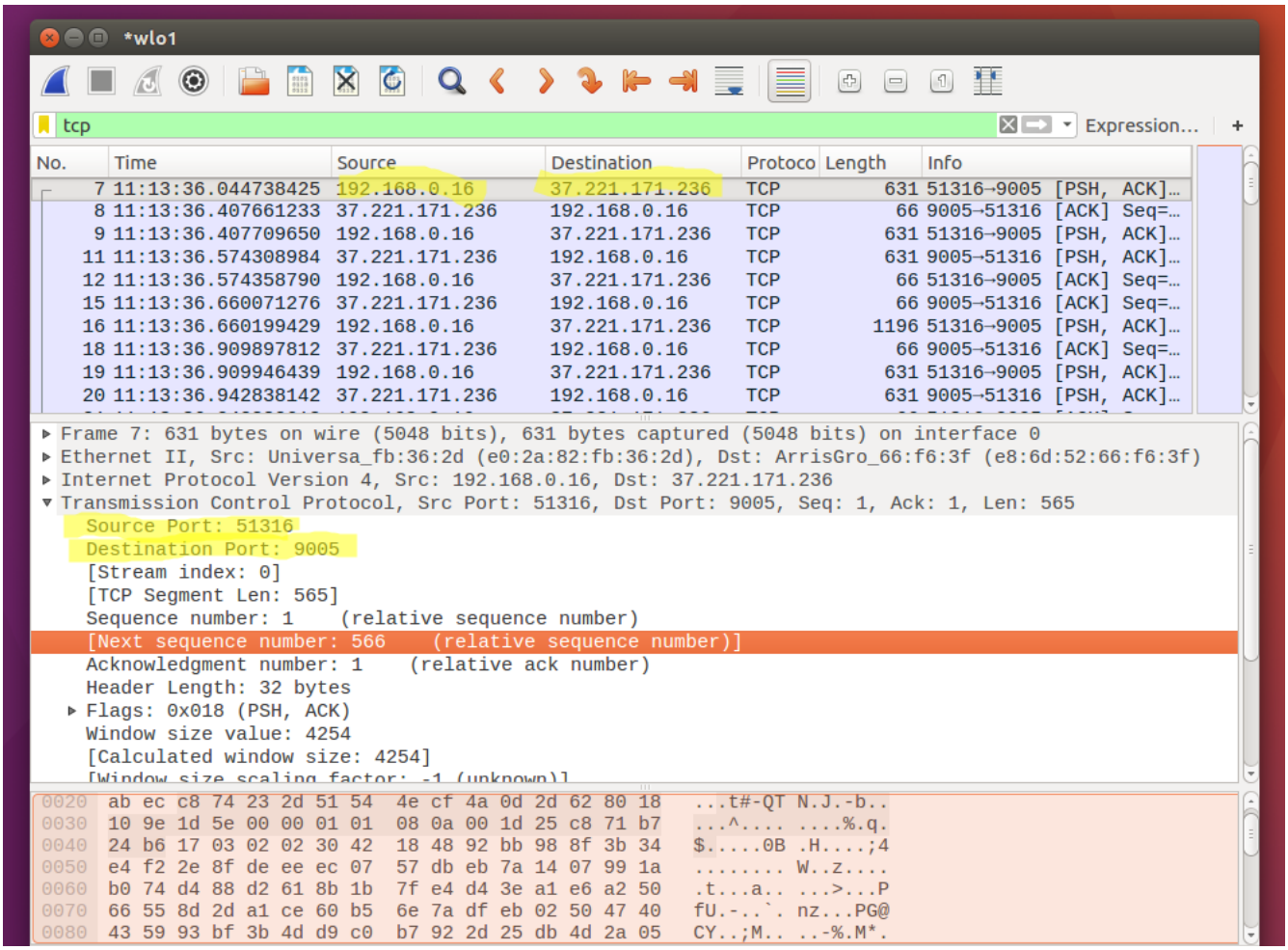


Client computer (source):
IP address: 192.168.1.102 (Source)
TCP port number: 1161 (source)

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?
ANS:
Destination computer(gaia.cs.umass.edu):
IP address: 128.119.245.12
TCP port number: 80

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

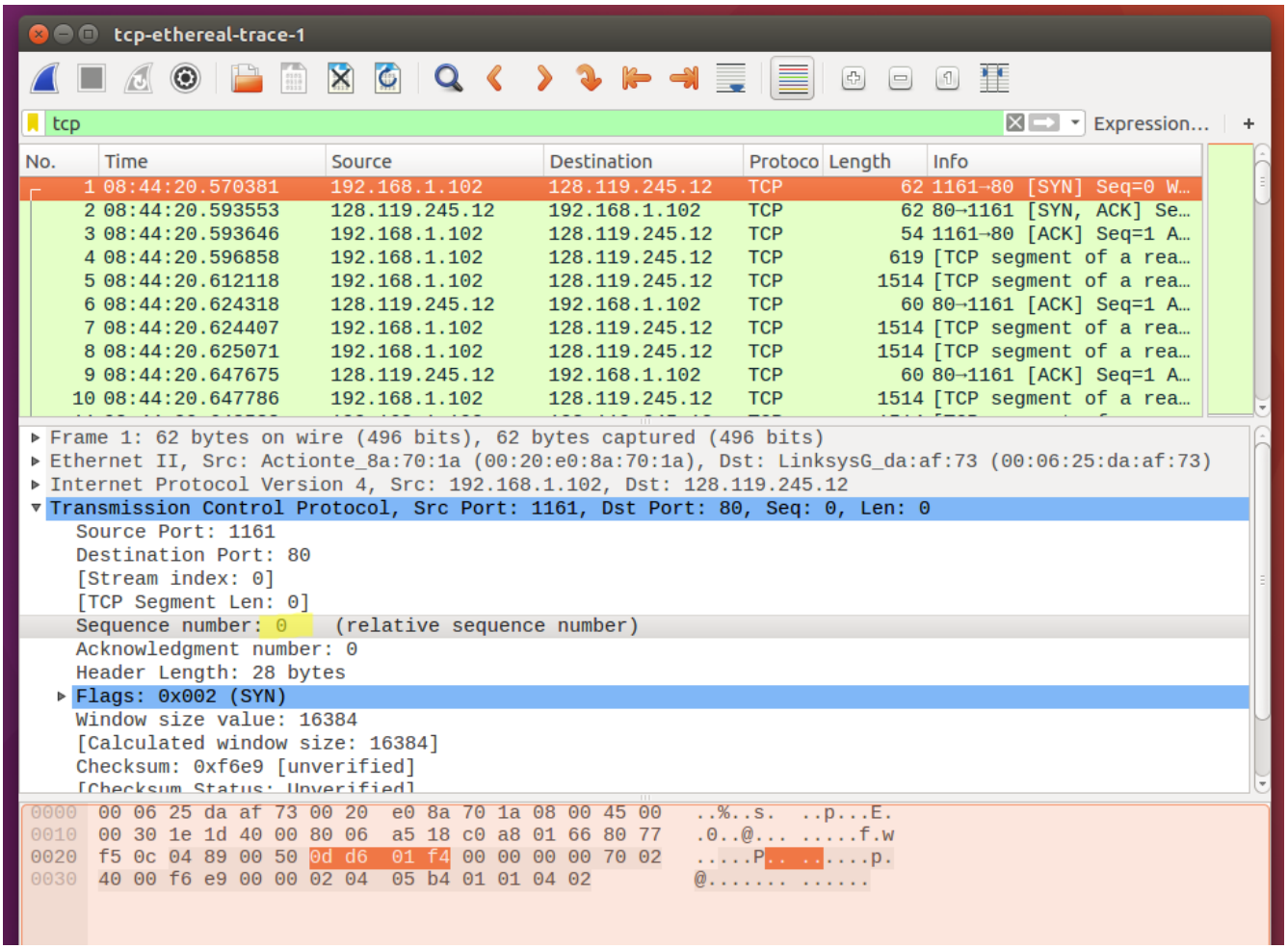
3. What is the IP address and TCP port number used by your client computer (source) to transfer the file to gaia.cs.umass.edu?
ANS:



IP address: 192.168.0.16
TCP port number: 51316

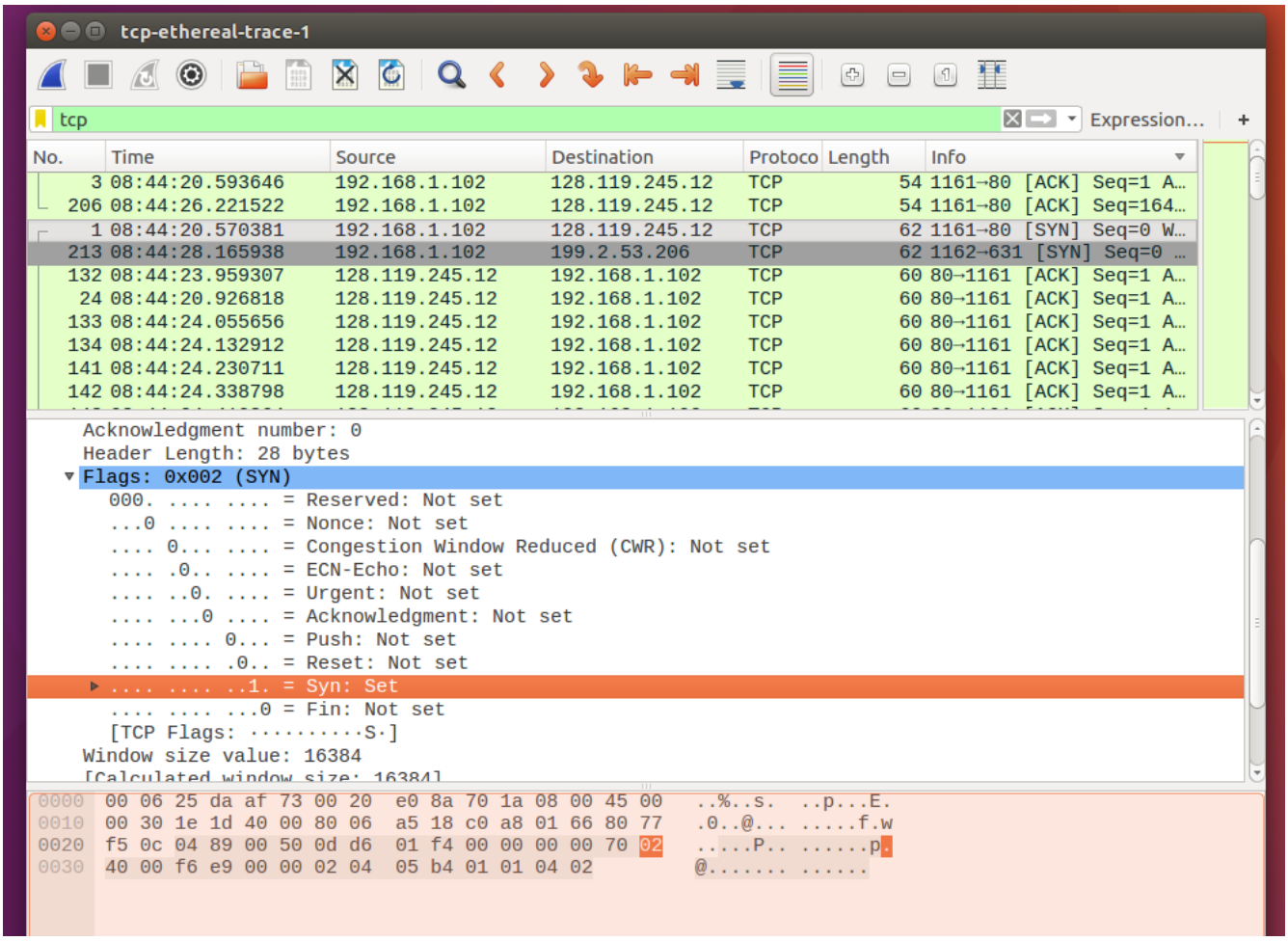
4. What is the sequence number of the TCP SYN segment that is used to initiate the TCP connection between the client computer and gaia.cs.umass.edu? What is it in the segment that identifies the segment as a SYN segment?
ANS:

Sequence number of the TCP SYN segment is used to initiate the TCP connection between the client computer and gaia.cs.umass.edu. The value is 0 in this trace. The SYN flag is set to 1 and it indicates that this segment is a SYN segment.

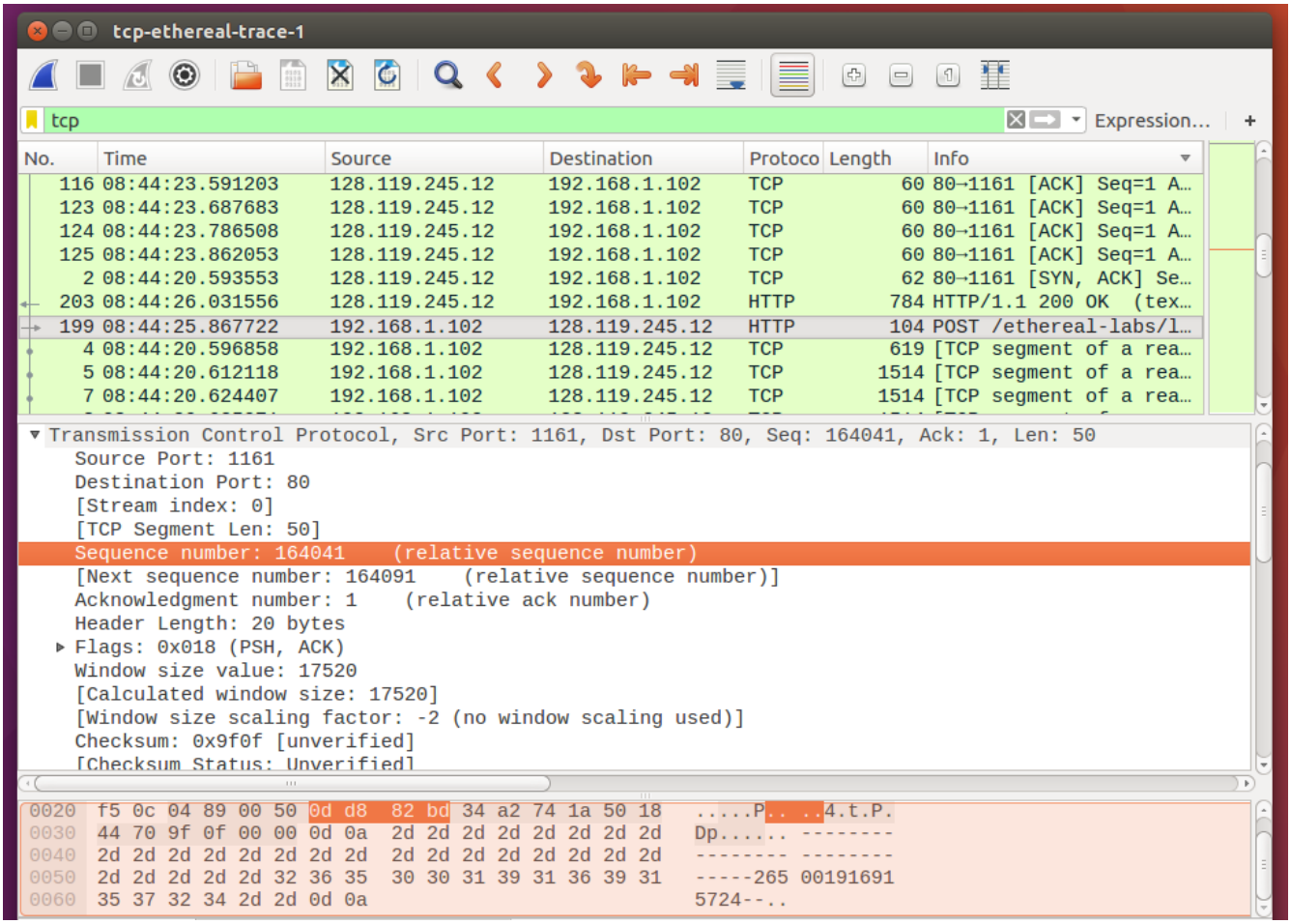


5. What is the sequence number of the SYNACK segment sent by gaia.cs.umass.edu to the client computer in reply to the SYN? What is the value of the Acknowledgement field in the SYNACK segment? How did gaia.cs.umass.edu determine that value? What is it in the segment that identifies the segment as a SYNACK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

Sequence number of the SYNACK segment from gaia.cs.umass.edu to the client computer in reply to the SYN has the value of 0 in this trace. The value of the ACKnowledgement field in the SYNACK segment is 1. The value of the ACKnowledgement field in the SYNACK segment is determined by gaia.cs.umass.edu by adding 1 to the initial sequence number of SYN segment from the client computer (i.e. the sequence number of the SYN segment initiated by the client computer is 0.). The SYN flag and Acknowledgement flag in the segment are set to 1 and they indicate that this segment is a SYNACK segment.



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.
ANS:
No. 199 segment is the TCP segment containing the HTTP POST command. The sequence number of this segment has the value of 164841.



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 239 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 239 for all subsequent segments.

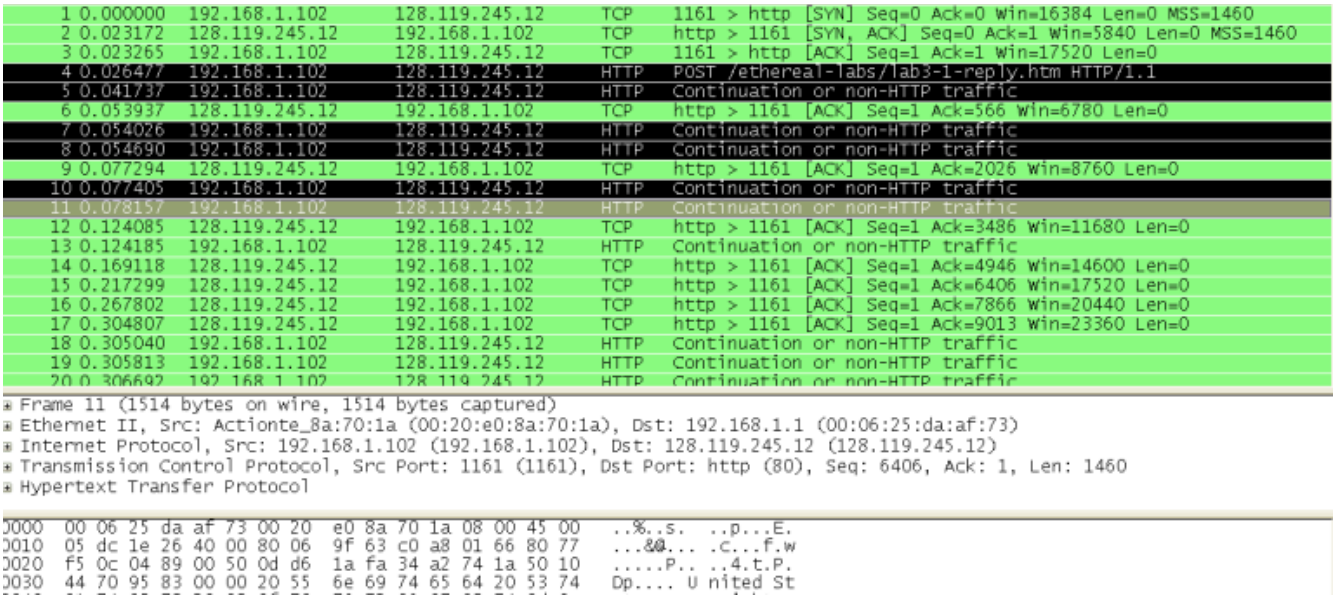
Note: Wireshark has a nice feature that allows you to plot the RTT for each of the TCP segments sent. Select a TCP segment in the "listing of captured packets" window that is being sent from the client to the gaia.cs.umass.edu server. Then select: Statistics->TCP Stream Graph- >Round Trip Time Graph.

ANS:

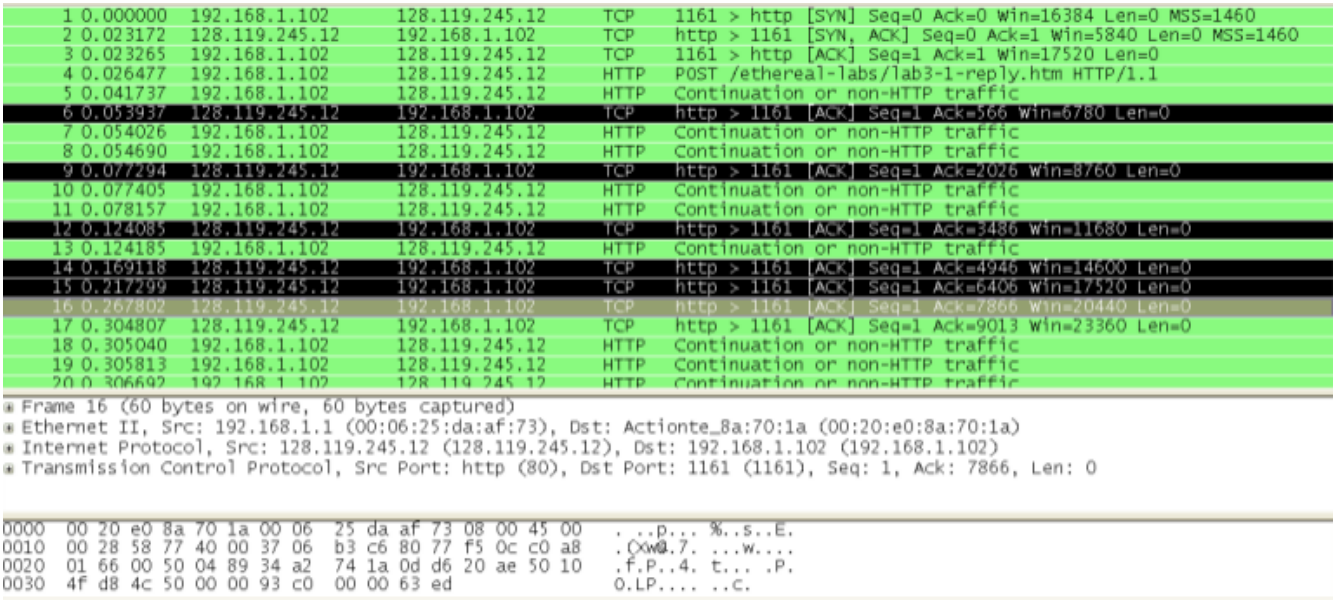
The HTTP POST segment is considered as the first segment. Segments 1 – 6 are No. 4, 5, 7, 8, 10, and 11 in this trace respectively. The ACKs of segments 1 – 6 are No. 6, 9, 12, 14, 15, and 16 in this trace.

Segment 1 sequence number: 1
Segment 2 sequence number: 566
Segment 3 sequence number: 2026
Segment 4 sequence number: 3486
Segment 5 sequence number: 4946
Segment 6 sequence number: 6406

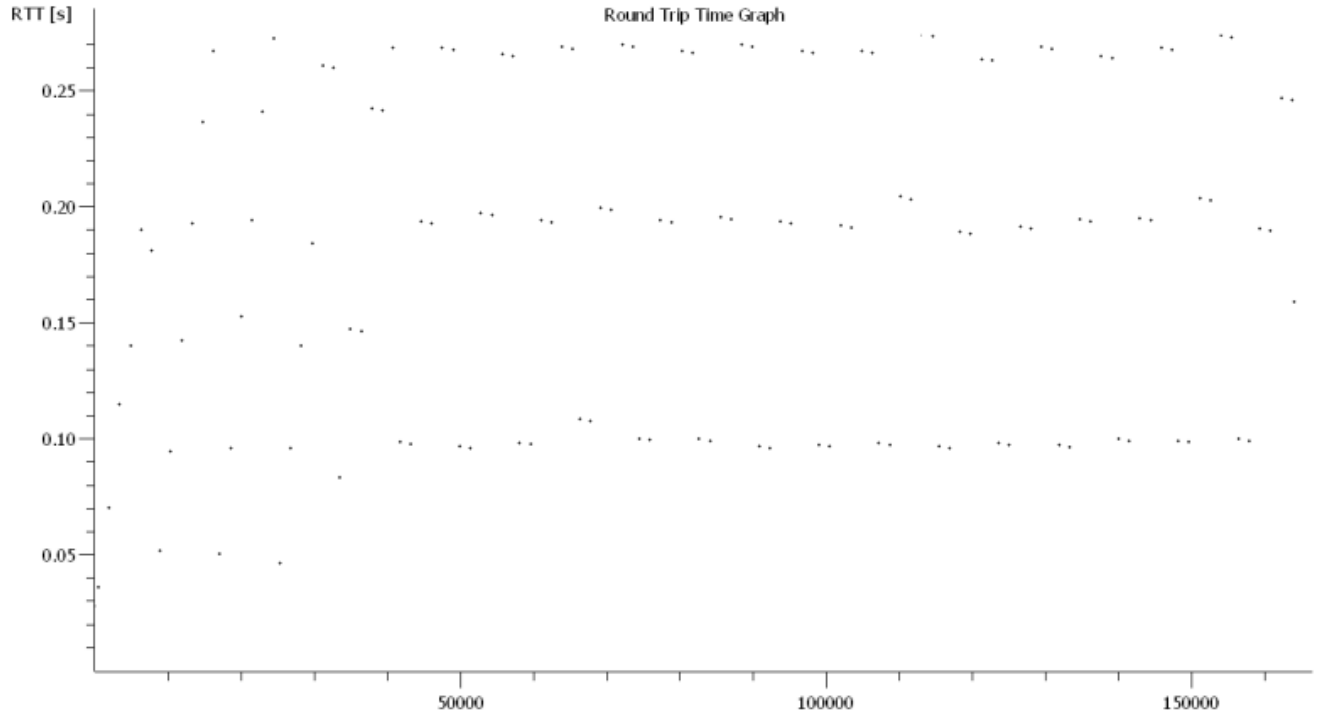
EstimatedRTT = 0.875 * EstimatedRTT + 0.125 * SampleRTT
EstimatedRTT after the receipt of the ACK of segment 1:
EstimatedRTT = RTT for Segment 1 = 0.02746 second
EstimatedRTT after the receipt of the ACK of segment 2:
EstimatedRTT = 0.875 * 0.02746 + 0.125 * 0.035557 = 0.0285
EstimatedRTT after the receipt of the ACK of segment 3:
EstimatedRTT = 0.875 * 0.0285 + 0.125 * 0.070059 = 0.0337
EstimatedRTT after the receipt of the ACK of segment 4:
EstimatedRTT = 0.875 * 0.0337+ 0.125 * 0.11443 = 0.0438
EstimatedRTT after the receipt of the ACK of segment 5:
EstimatedRTT = 0.875 * 0.0438 + 0.125 * 0.13989 = 0.0558
EstimatedRTT after the receipt of the ACK of segment 6:
EstimatedRTT = 0.875 * 0.0558 + 0.125 * 0.18964 = 0.0725 second



segments 1-6



ACK’s of segments 1-6



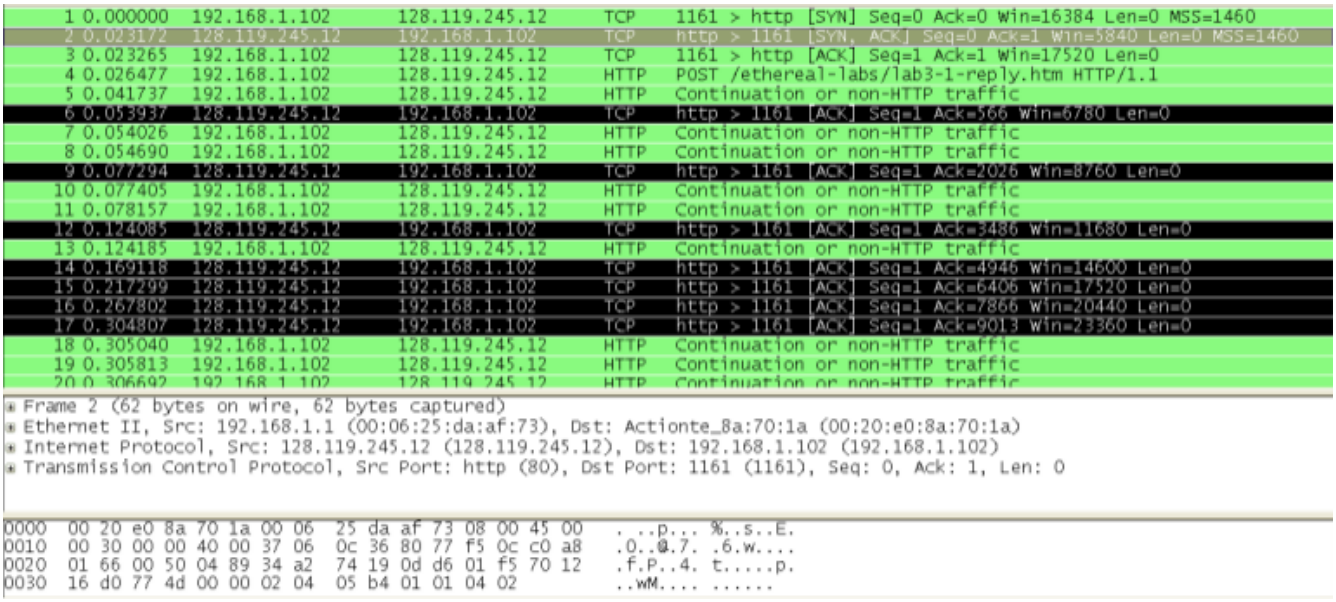
Round trip time graph

8. What is the length of each of the first six TCP segments?
ANS:
Length of the first TCP segment: 565 bytes
Length of each of the other five TCP segments: 1460 bytes

1	0.000000	192.168.1.102	128.119.245.12	TCP	1161 > http [SYN] Seq=0 Ack=0 Win=16384 Len=0 MSS=1460
2	0.023172	128.119.245.12	192.168.1.102	TCP	http > 1161 [SYN, ACK] Seq=0 Ack=1 Win=5840 Len=0 MSS=1460
3	0.023265	192.168.1.102	128.119.245.12	TCP	1161 > http [ACK] Seq=1 Ack=1 Win=17520 Len=0
4	0.026477	192.168.1.102	128.119.245.12	HTTP	POST /ethereal-labs/lab3-1-reply.htm HTTP/1.1
5	0.041737	192.168.1.102	128.119.245.12	HTTP	Continuation or non-HTTP traffic
6	0.053937	128.119.245.12	192.168.1.102	TCP	http > 1161 [ACK] Seq=1 Ack=566 Win=6780 Len=0
7	0.054026	192.168.1.102	128.119.245.12	HTTP	Continuation or non-HTTP traffic
8	0.054690	192.168.1.102	128.119.245.12	HTTP	Continuation or non-HTTP traffic
9	0.077294	128.119.245.12	192.168.1.102	TCP	http > 1161 [ACK] Seq=1 Ack=2026 Win=8760 Len=0
10	0.077405	192.168.1.102	128.119.245.12	HTTP	Continuation or non-HTTP traffic
11	0.078157	192.168.1.102	128.119.245.12	HTTP	Continuation or non-HTTP traffic
12	0.124085	128.119.245.12	192.168.1.102	TCP	http > 1161 [ACK] Seq=1 Ack=3486 Win=11680 Len=0
13	0.124185	192.168.1.102	128.119.245.12	HTTP	Continuation or non-HTTP traffic
14	0.169118	128.119.245.12	192.168.1.102	TCP	http > 1161 [ACK] Seq=1 Ack=4946 Win=14600 Len=0
15	0.217299	128.119.245.12	192.168.1.102	TCP	http > 1161 [ACK] Seq=1 Ack=6406 Win=17520 Len=0
16	0.267802	128.119.245.12	192.168.1.102	TCP	http > 1161 [ACK] Seq=1 Ack=7866 Win=20440 Len=0

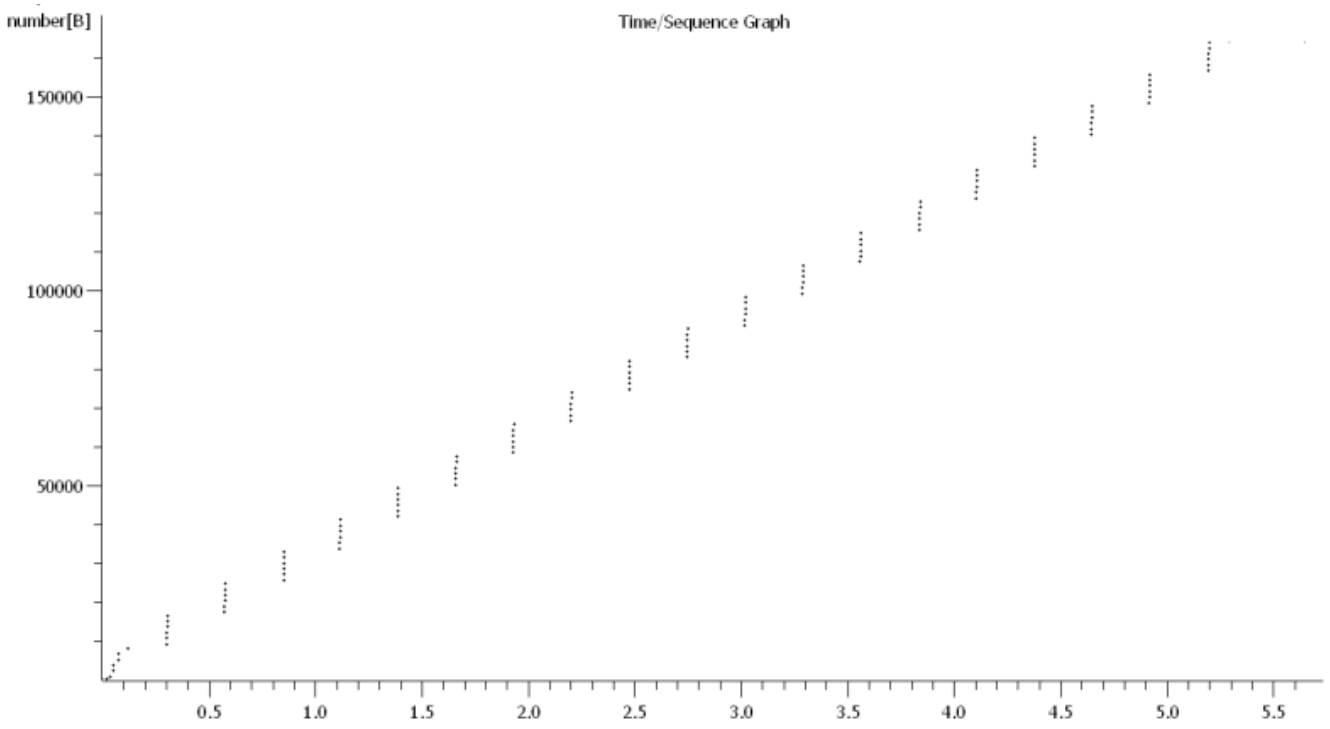
⊞ Frame 11 (1514 bytes on wire, 1514 bytes captured)
⊞ Ethernet II, Src: Actionte_8a:70:1a (00:20:e0:8a:70:1a), Dst: 192.168.1.1 (00:06:25:da:af:73)
⊞ Internet Protocol, Src: 192.168.1.102 (192.168.1.102), Dst: 128.119.245.12 (128.119.245.12)
⊞ Transmission Control Protocol, Src Port: 1161 (1161), Dst Port: http (80), Seq: 6406, Ack: 1, Len: 1460
⊟ Hypertext Transfer Protocol
Data (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?
ANS:
The minimum amount of buffer space advertised at gaia.cs.umass.edu for the entire trace is 5840 bytes, which shows in the first acknowledgement from the server. This receiver window grows steadily until a maximum receiver buffer size of 62780 bytes. The sender is never throttled due to lacking of receiver buffer space by inspecting this trace.



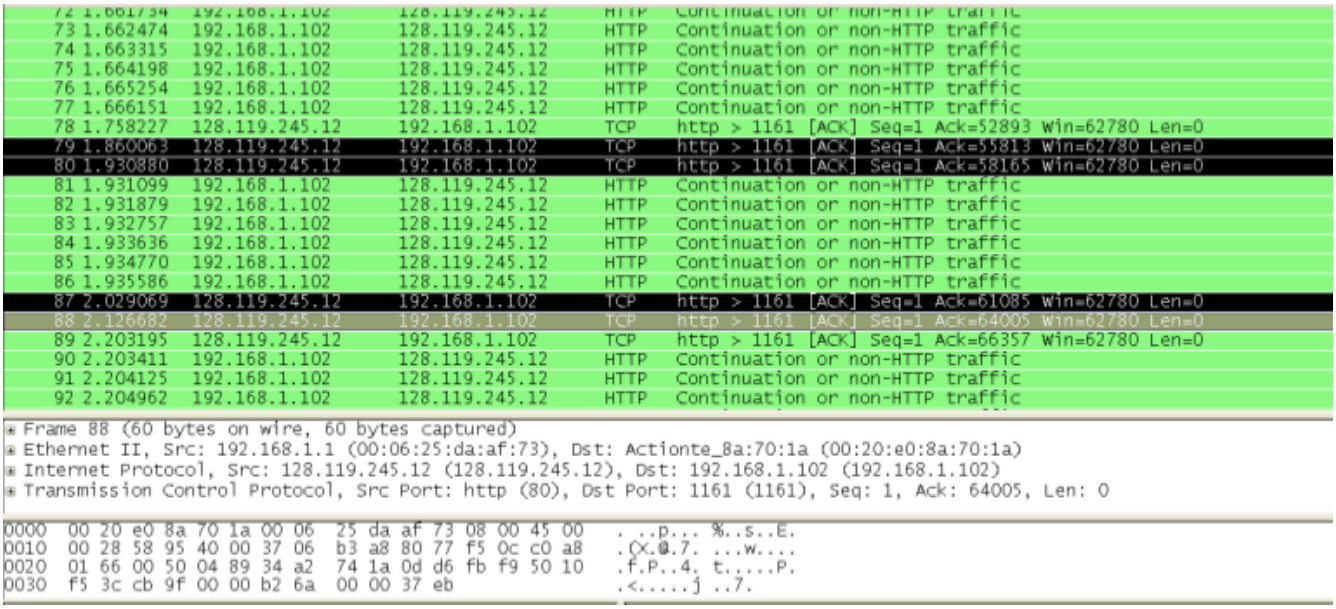
10. Are there any retransmitted segments in the trace file? What did you check for (in the trace) in order to answer this question?
ANS:

There are no retransmitted segments in the trace file. We can verify this by checking the sequence numbers of the TCP segments in the trace file. In the Time-Sequence-Graph of this trace, all sequence numbers from the source (192.168.1.102) to the destination (128.119.245.12) are increasing monotonically with respect to time. If there is a retransmitted segment, the sequence number of this retransmitted segment should be smaller than those of its neighboring segments.



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).
ANS:

Difference between the acknowledged sequence numbers of two consecutive ACKs indicates the data received by the server between these two ACKs. For example, segment of No. 80 acknowledged data with 2920 bytes = 1460*2 bytes.



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

Computation of TCP throughput largely depends on the selection of averaging time period. As a common throughput computation, in this question, we select the average time period as the whole connection time. The average throughput for this TCP connection is computed as the ratio between the total amount data and the total transmission time. The total amount data transmitted can be computed by the difference between the sequence number of the first TCP segment (1 byte for No. 4 segment) and the acknowledged sequence number of the last ACK (164091 bytes for No. 202 segment). Therefore, the total data are 164091 - 1 = 164090 bytes. The whole transmission time is the difference of the time instant of the first TCP segment (0.026477 second for No.4 segment) and the time instant of the last ACK (5.455830 second for No. 202 segment). The total transmission time is 5.455830 - 0.026477 = 5.4294 seconds. The throughput for the TCP connection is computed as 164090/5.4294 = 30.222 Kbyte/sec.

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 hich the measured data differs from the idealized behavior of TCP that we've studied in the text
ANS:
TCP Slow Start begins at the start of the connection, when the HTTP POST segment is sent out. The identification of the TCP slow start phase and congestion avoidance phase depends on the value of the congestion window size of this TCP sender. However, the value of the congestion window size cannot be obtained directly from the Time-Sequence-Graph (Stevens) graph. Nevertheless, we can estimate the lower bound of the TCP window size by the amount of outstanding data because the outstanding data is the amount of data without acknowledgement. We also know that TCP window is constrained by the receiver window size and the receiver buffer can act as the upper bound of the TCP window size. In this trace, the receiver buffer is not the bottleneck; therefore, this upper bound is not quite useful to infer the TCP window size. Hence, we focus on the lower bound of the TCP window size.