Wireshark Lab: TCP v6.0

Vandré Leal Cândido

September 1, 2017

1 A first look at the captured trace

Answer the following questions, by opening the Wireshark captured packet file tcpethereal-trace-1 in http://gaia.cs.umass.edu/wireshark-labs/wireshark-traces.zip (that is download the trace and open that trace in Wireshark; see footnote 2). Whenever possible, when answering a question you should hand in a printout of the packet(s) within the trace that you used to answer the question asked. Annotate the 3 printout to explain your answer. To print a packet, use File - Print, choose Selected packet only, choose Packet summary line, and select the minimum amount of packet detail that you need to answer the question.

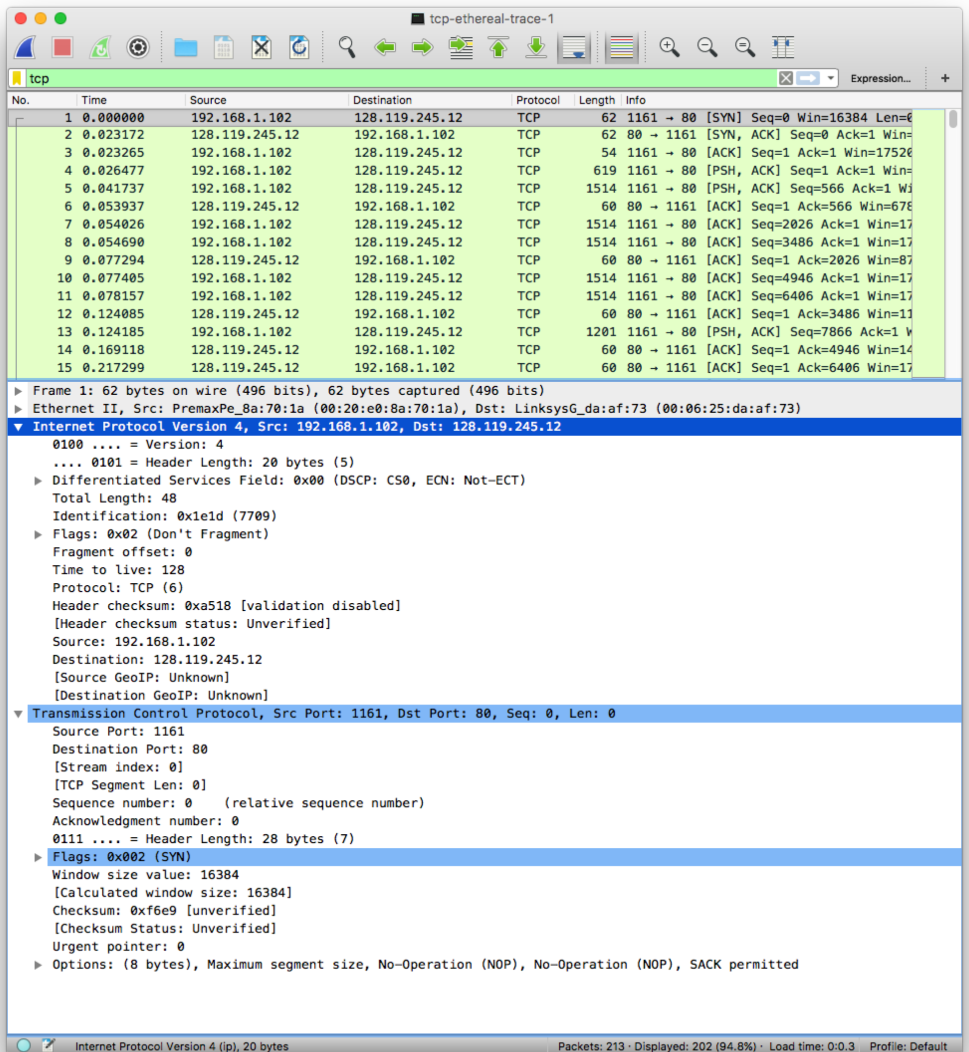
- *What is the IP address and TCP port number used by the client computer (source) that is transferring the file to gaia.cs.umass.edu? To answer this question, it's probably easiest to select an HTTP message and explore the details of the TCP packet used to carry this HTTP message, using the "details 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.*

  The IP address used by the client computer is 192.168.1.102 and the source TCP port number is 1161.

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

  The IP address of gaia.cs.umass.edu is 128.119.245.12 and the TCP port is 80.

Figure 1: tcpethereal-trace-1

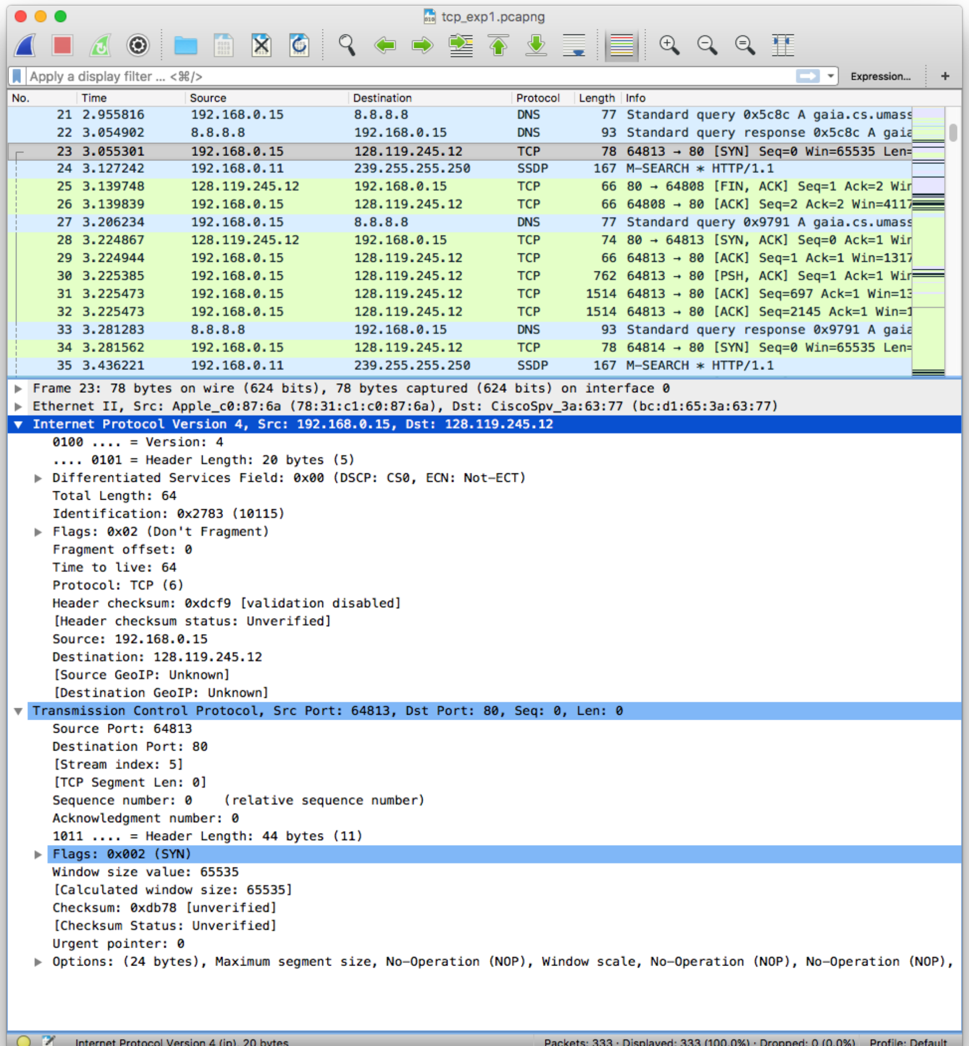


If you have been able to create your own trace, answer the following question:

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

  The IP address used by my computer is 192.168.0.15 and the TCP port is 64813.

Figure 2: tcplocal-trace-1

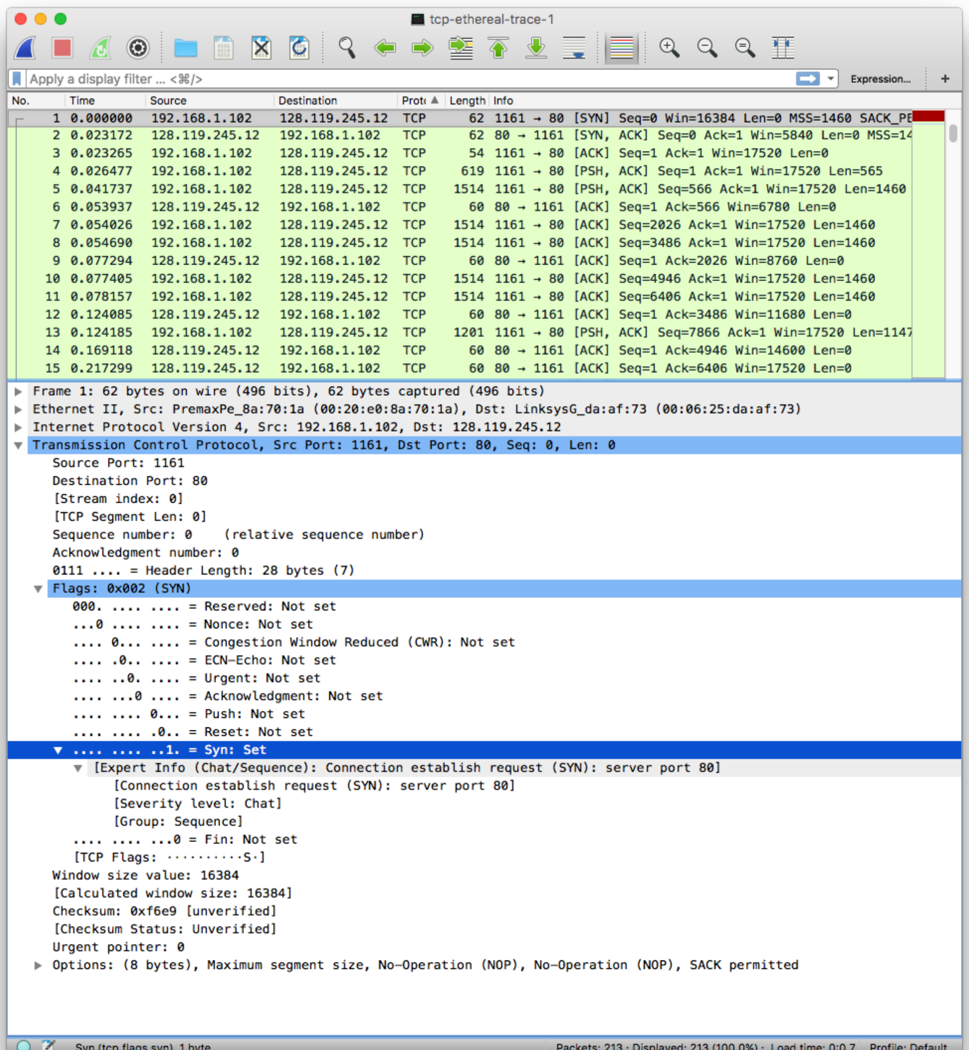


2 TCP Basics

- *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 used to initiate the TCP connection is 0. The **SYN** flag highlighted below indicates that the segment is a SYN segment.

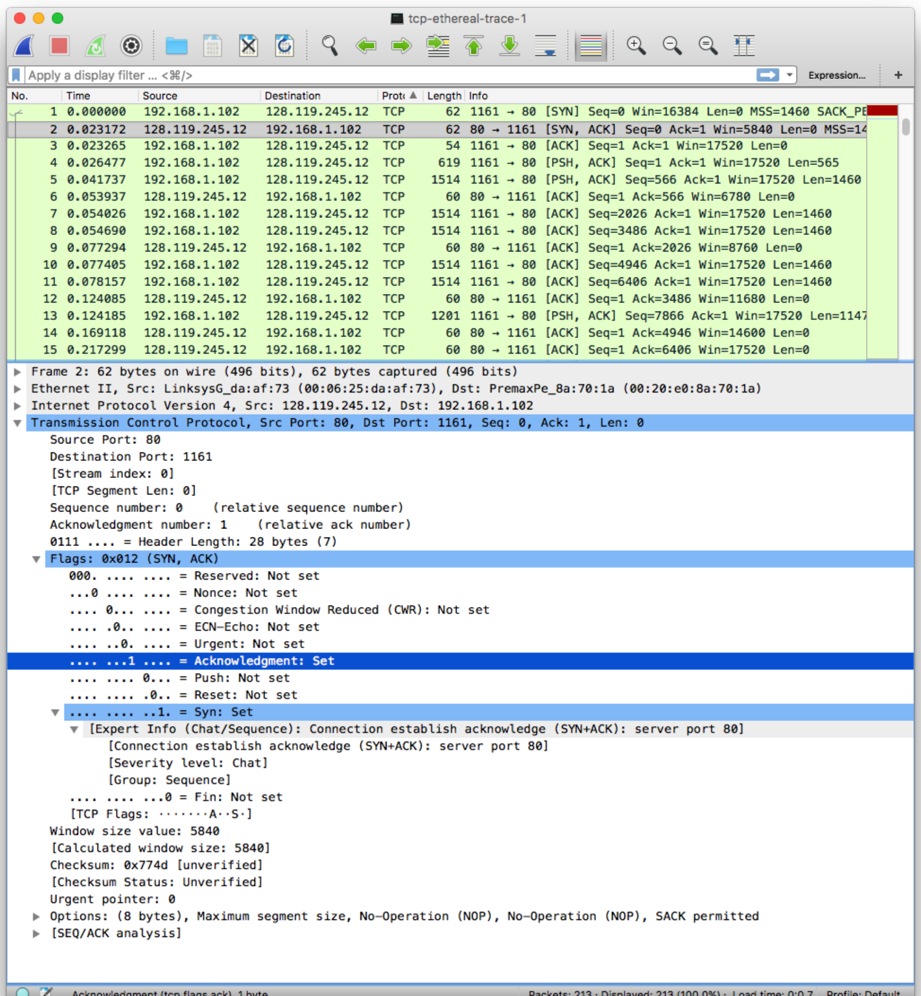
Figure 3: tcplocal-trace-1 (Question 04)



- *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 of the SYNACK segment sent by gaia.cs.umass.edu to the client computer is 0. The value of the ACK filed in the SYNACK segment is 1 as highlighted below. gaia.cs.umass.edu determined the value by adding 1 to the initial sequence number of SYN segment (0). Both the **SYN** and the **Acknoledgment** flags are set to 1, which indicates that this is a SYNACK segment.

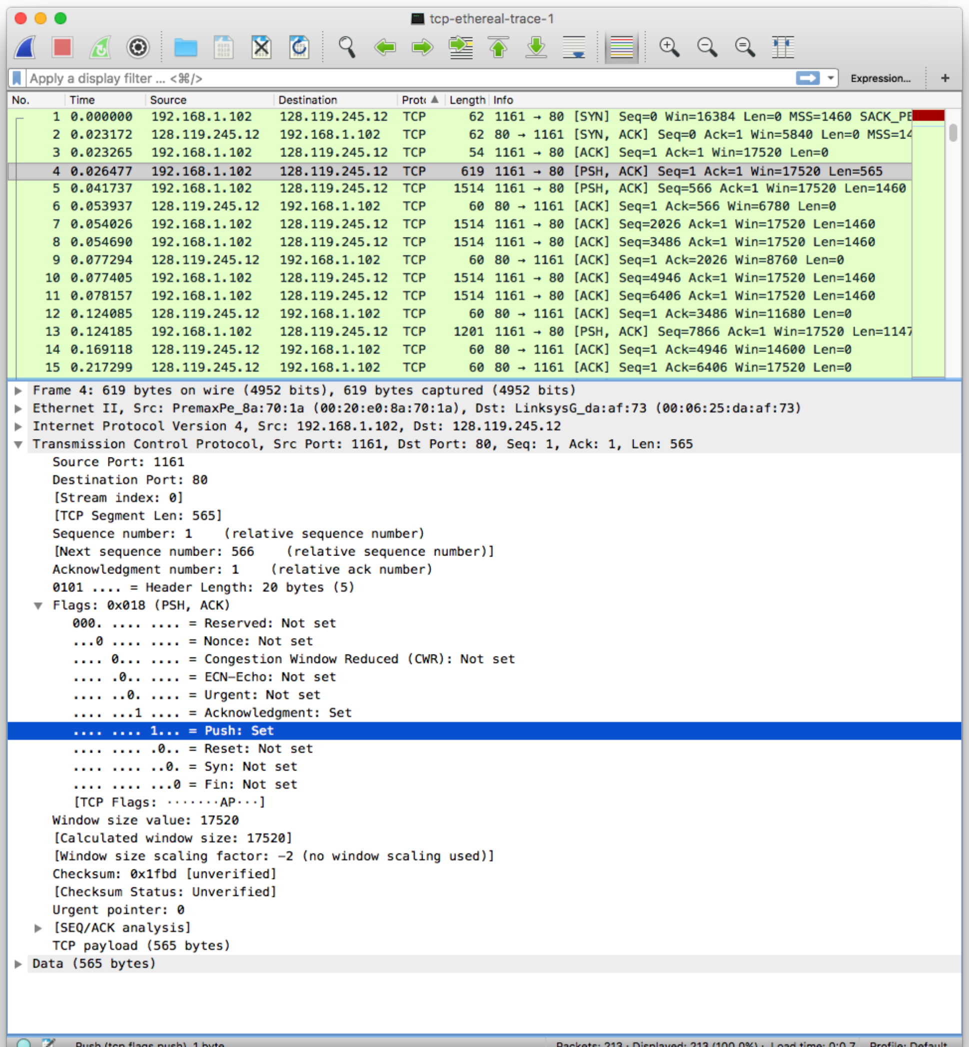
Figure 4: tcplocal-trace-1 (Question 05)



- *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.*

  The sequence number of the TCP segment containing the POST command is 1. The segment is highlighted below and has the flag **Push** set to 1 indicating that the client is sending data.

Figure 5: tcplocal-trace-1 (Question 06)



- *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.*

  The sequence numbers of the first six segments in the TCP connection are 1, 566, 2026, 3486, 4946 and 6406 (frames 4, 5, 7, 8, 9 and 10) in the trace. The ACK received for each one the segments are frames 6, 9, 12, 14, 15 and 16, respectively.

Table 1: RTT values (seconds)

Segment	Frame	Sent time (seconds)	ACK received (seconds)	RTT (seconds)
1	4	0.026477	0.053937	0.02746
2	5	0.041737	0.077294	0.035557
3	7	0.054026	0.124085	0.070059
4	8	0.054690	0.169118	0.11443
5	10	0.077405	0.217299	0.13989
6	11	0.078157	0.267802	0.18964

The EstimatedRTT equation on page 239 is

*EstimatedRTT = 0.875 * EstimatedRTT + 0.125 * SampleRTT*

Segment 1: *EstimatedRTT = 0.02746 secs*

Segment 2: *EstimatedRTT = 0.875 * 0.02746 + 0.125 * 0.035557 = 0.0285 secs*

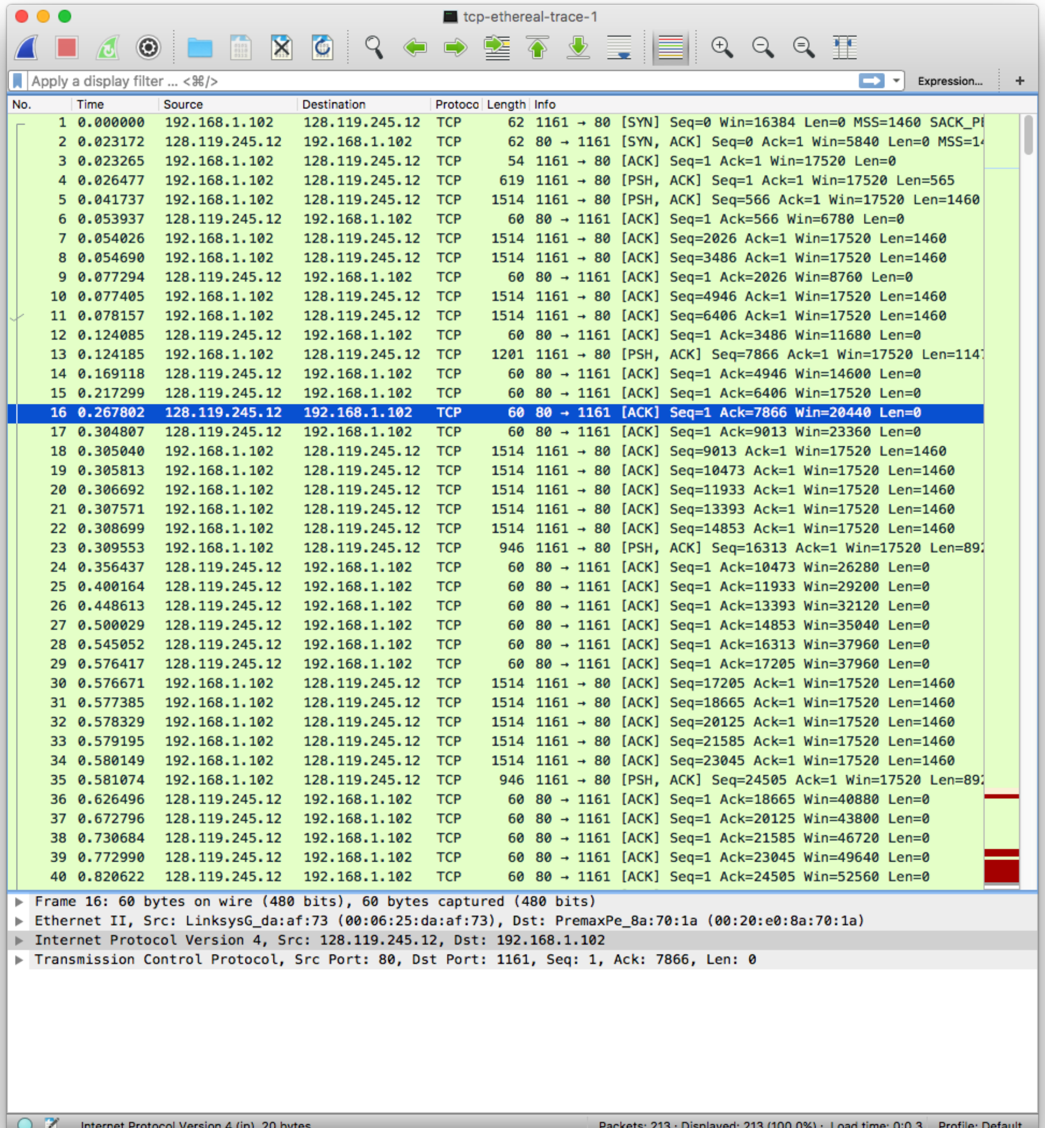
Segment 3: *EstimatedRTT = 0.875 * 0.0285 + 0.125 * 0.070059 = 0.0337 secs*

Segment 4: *EstimatedRTT = 0.875 * 0.0337+ 0.125 * 0.11443 = 0.0438 secs*

Segment 5: *EstimatedRTT = 0.875 * 0.0438 + 0.125 * 0.13989 = 0.0558 secs*

Segment 6: *EstimatedRTT = 0.875 * 0.0558 + 0.125 * 0.18964 = 0.0725 secs*

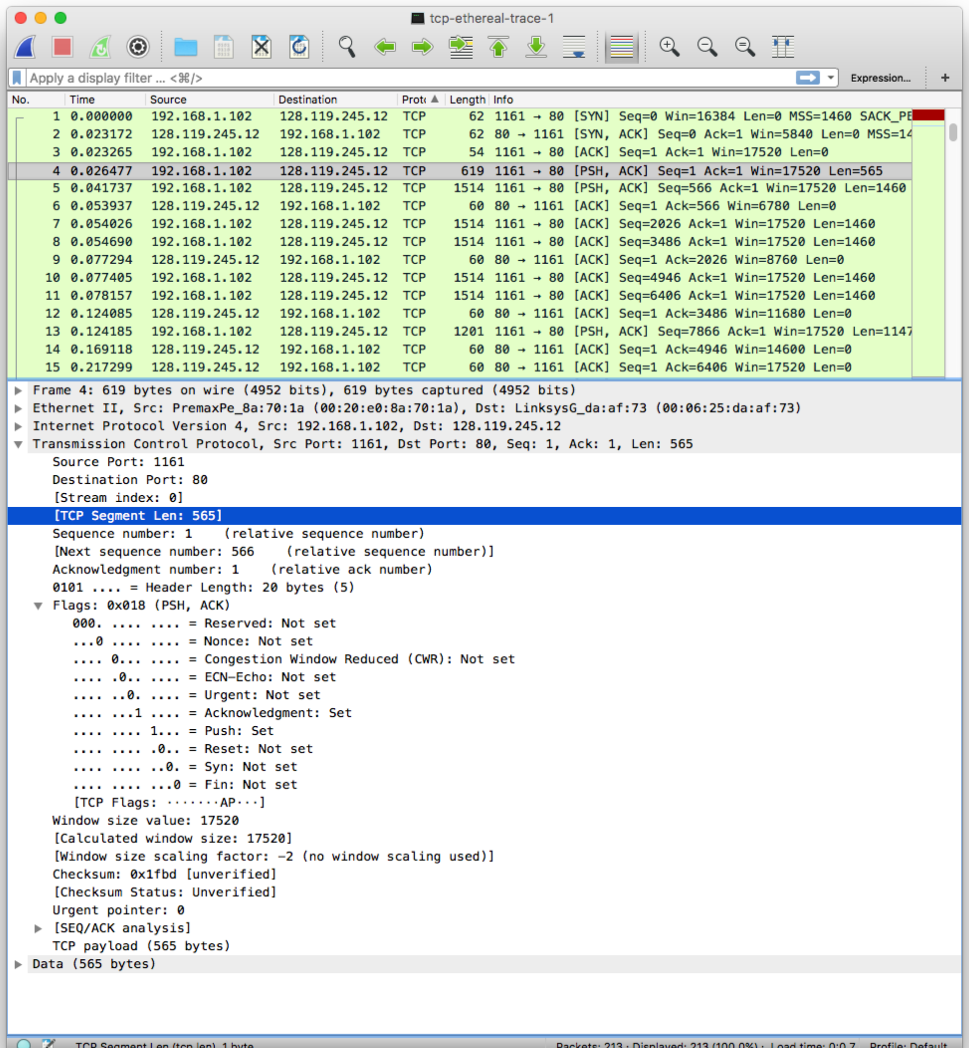
Figure 6: tcplocal-trace-1 (Question 07)



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

  The length of the first TCP segment (including the HTTP POST) is 565 bytes. The length of each one of the other five segments is 1460 bytes. This value is included within the TCP section as highlighted below but can also be quickly identified by checking the value of **Len** in the column **Info** for each one of the segments in the trace.

Figure 7: tcplocal-trace-1 (Question 08)



- *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?*

  The minimum amount of buffer space advertised at the received (gaia.cs.umass.edu) for the entire trace is 5840 bytes as indicated by the field **Window size value** in the first ACK. The window size value grows until it reaches the value 62780 bytes as indicated in the last POST segment. By inspecting the trace it doesn't seem like the sender ever throttle due to lack of buffer space.

Figure 8: tcplocal-trace-1 (Question 09 - First ACK)

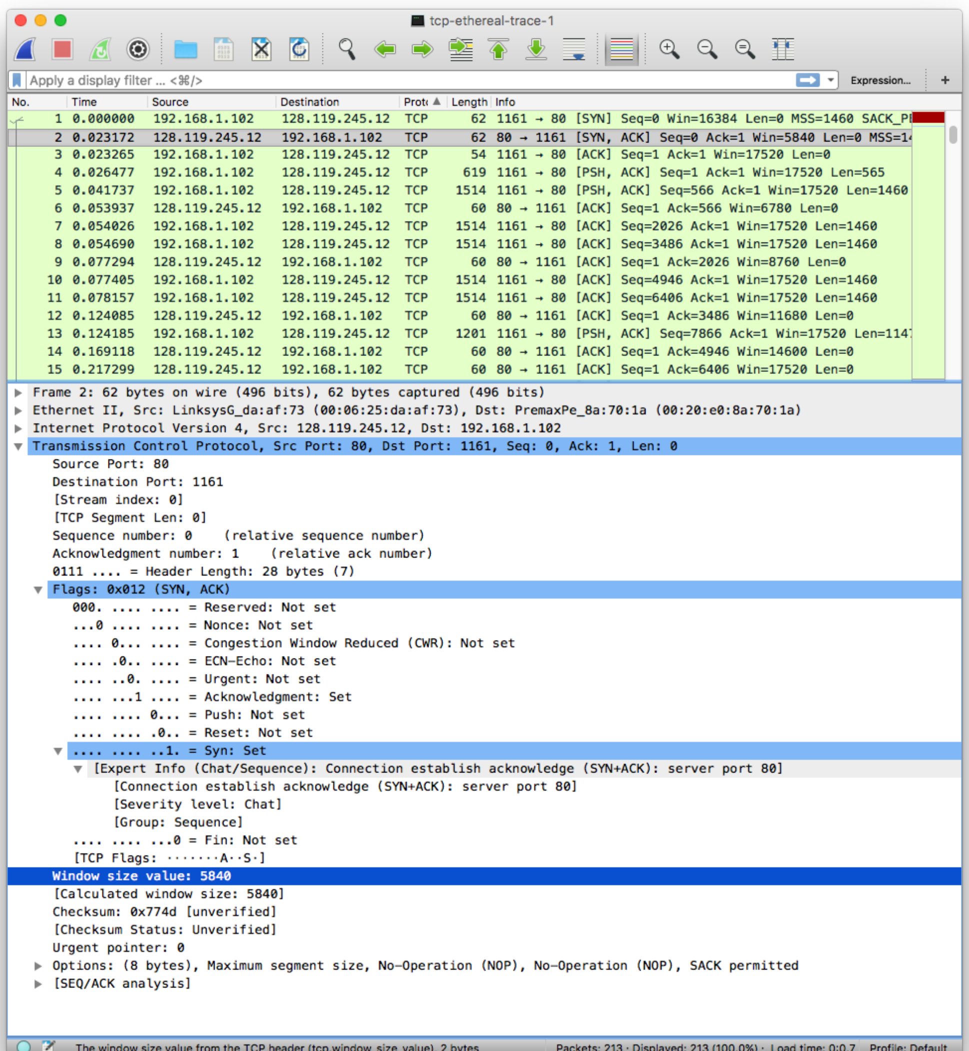
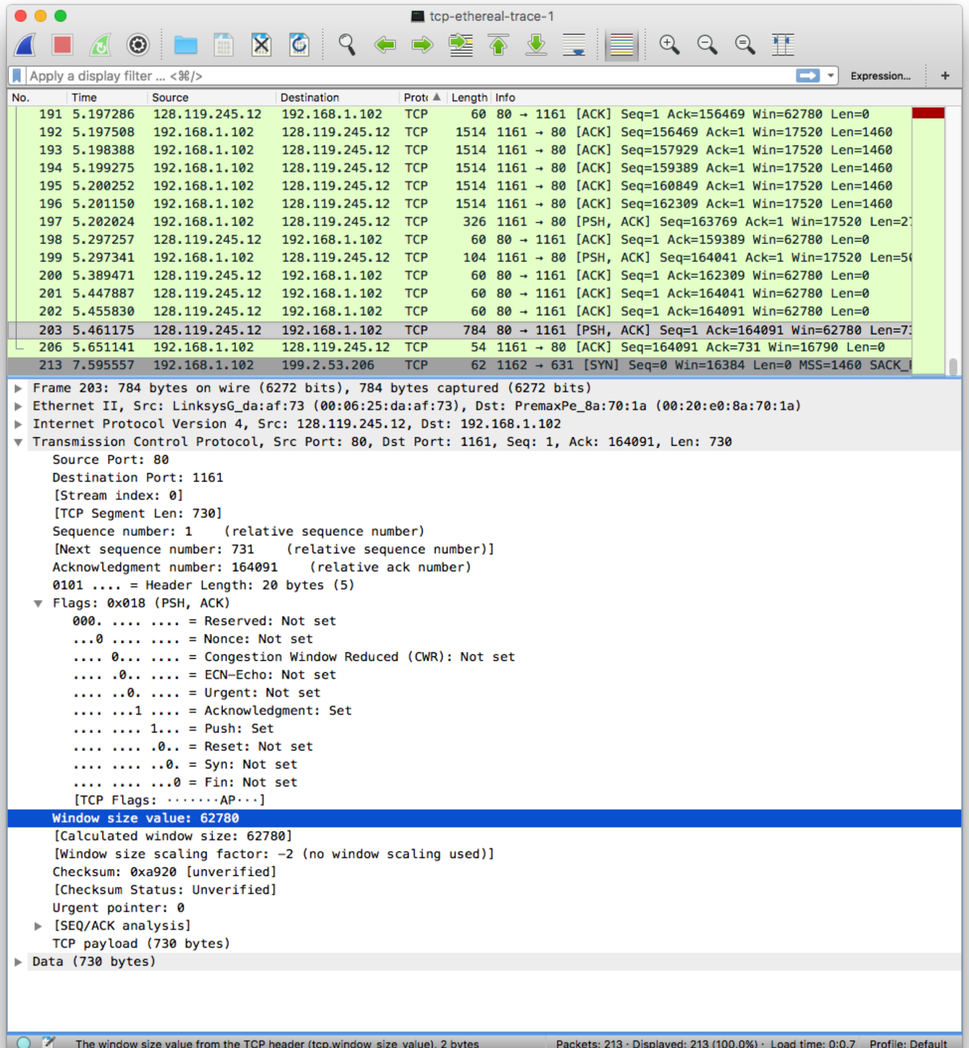


Figure 9: tcplocal-trace-1 (Question 09 - Last POST segment)



- *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 in the trace file. There aren't any wireshark frames highlighted as retransmitted [TCP Retransmission]. Additionally, all sequence numbers increases as packets are sent. When a segment is retransmitted, the sequence number of the segment is smaller than the sequence number of its nearby segments.

- *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).*

  The receiver typically acknowledge 1460 bytes in an ACK.

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

  The average throughput is computed as the ratio between the total amount data and the total transmission time.

  Total amount data:
  Sequence number of the last ACK minus the seq number of the first TCP segment
  164091 - 1
  164090 bytes.

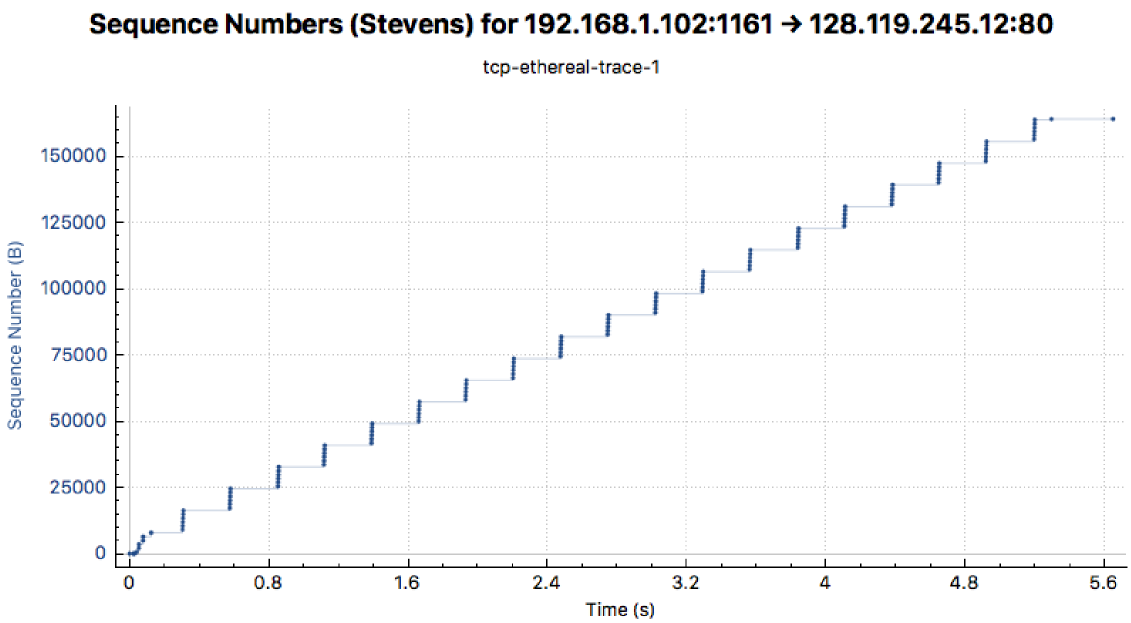
  Total transmission time:
  Time instant of the last ACK minus the time instant of the first TCP segment
  5.455830 - 0.026477
  5.4294 seconds.

  Therefore, the throughput for the TCP connection is 164090/5.4294 = **30.222 Kbs/sec**.

3 TCP congestion control in action

- *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.*

Figure 10: tcplocal-trace-1 (Question 13 - Time-Sequence-Graph(Stevens))



- *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*

  There's no way to determine the end of the slow start phase and the start of the congestion avoidance since the client is not sending enough data to force a congestion state. The client stops transmitting data before any congestion.