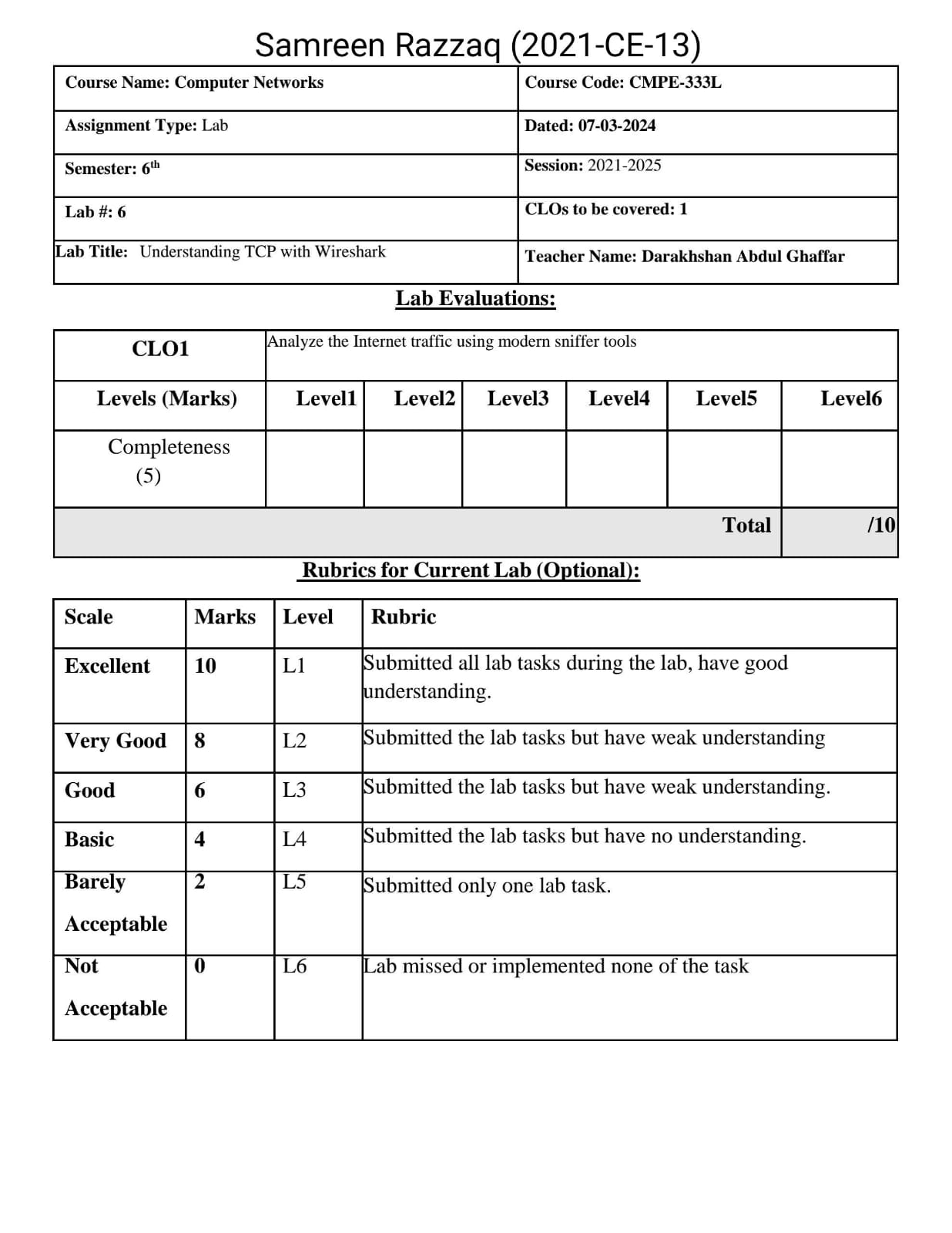
****

**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? To answer this question, it’s probably easiest to

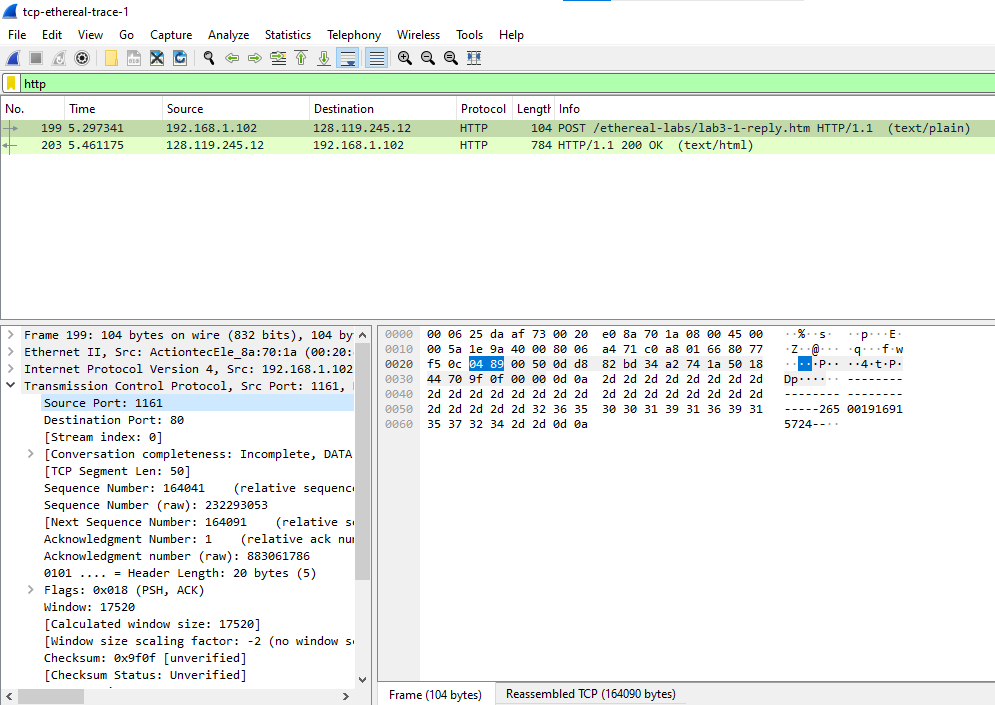
select an HTTP message and explore the details of the TCP packet used to carry this HTTP

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.

**Solution:**

The IP address is 192.168.1.102 and TCP port number is 1161.

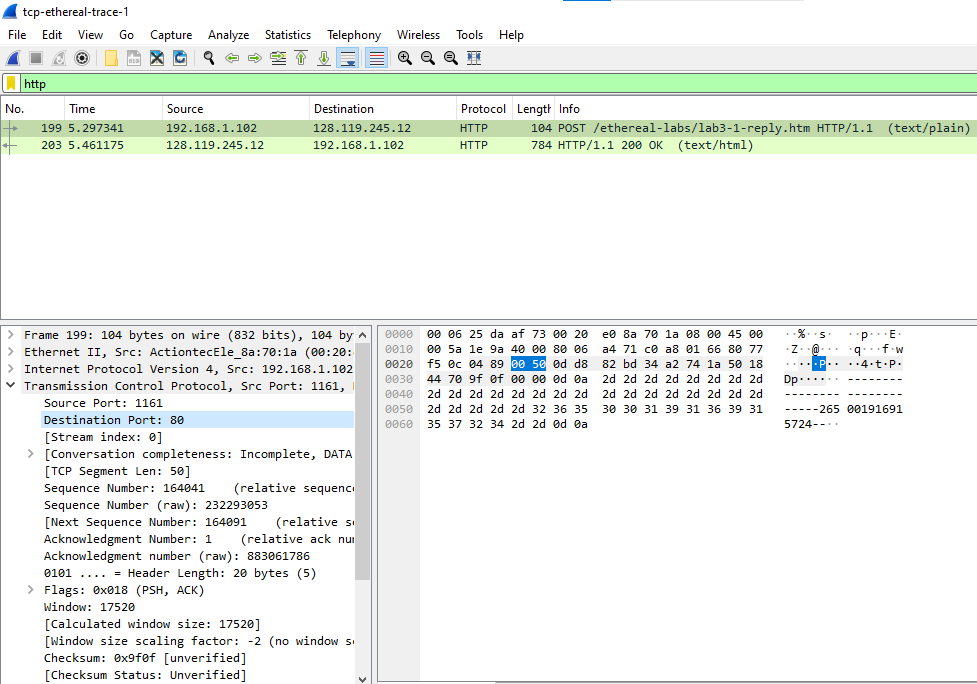


**2.** What is the IP address of gaia.cs.umass.edu? On what port number is it sending and

receiving TCP segments for this connection?

**Solution:**

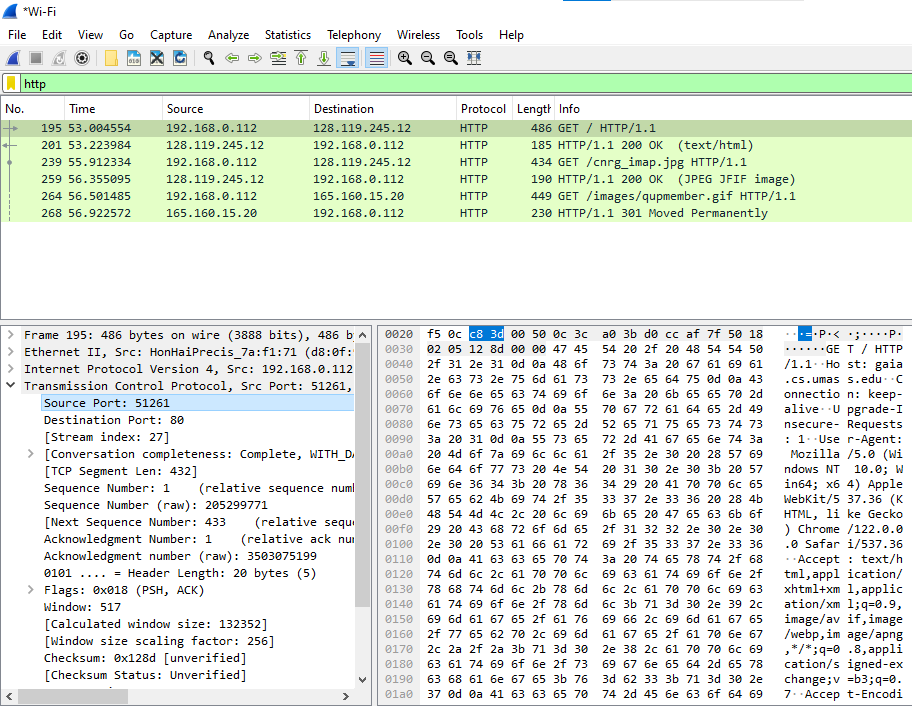
The IP address is 128.119.245.12 and port number is 80.



**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?

**Solution:**

The IP address is 192.168.0.112 and port number is 51261.

**TCP Basics**

**Answer the following questions for the TCP segments:**

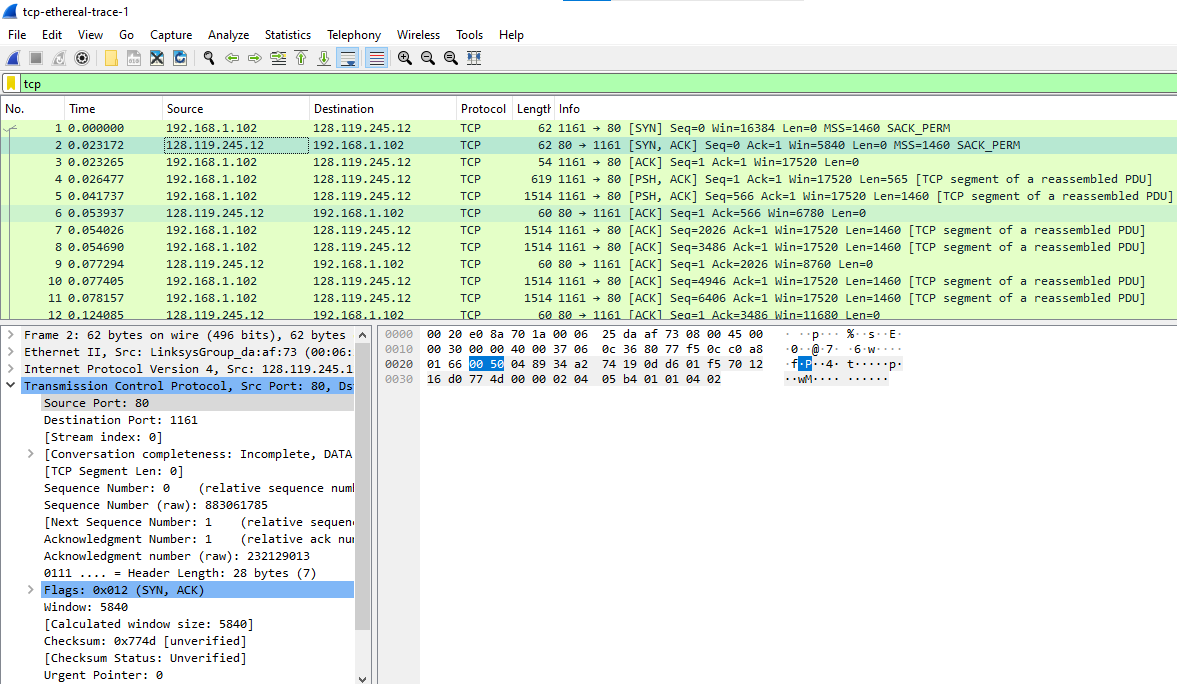
**4.** What is the sequence number of the TCP SYN segment that is used to initiate the TCP

connection between the client computer and gaia.cs.umass.edu? What is it in the segment

that identifies the segment as a SYN segment?

**Solution:**

The SYN\_ACK segment sent from gaia.cs.umass.edu to the client computer in response to the SYN has a sequence number of 0 and an acknowledgment field value of 1. This acknowledgment value is determined by adding 1 to the initial sequence number of the SYN segment from the client. If both the SYN and ACK flags are set to 1 in a segment, it is identified as a SYN\_ACK 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?

**Solution:**

The SYNACK segment's sequence number is 0, and its acknowledgment field holds the value 1. This acknowledgment value is derived from adding 1 to the initial sequence number. The presence of specific flags in the message indicates it is a SYN ACK message.

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

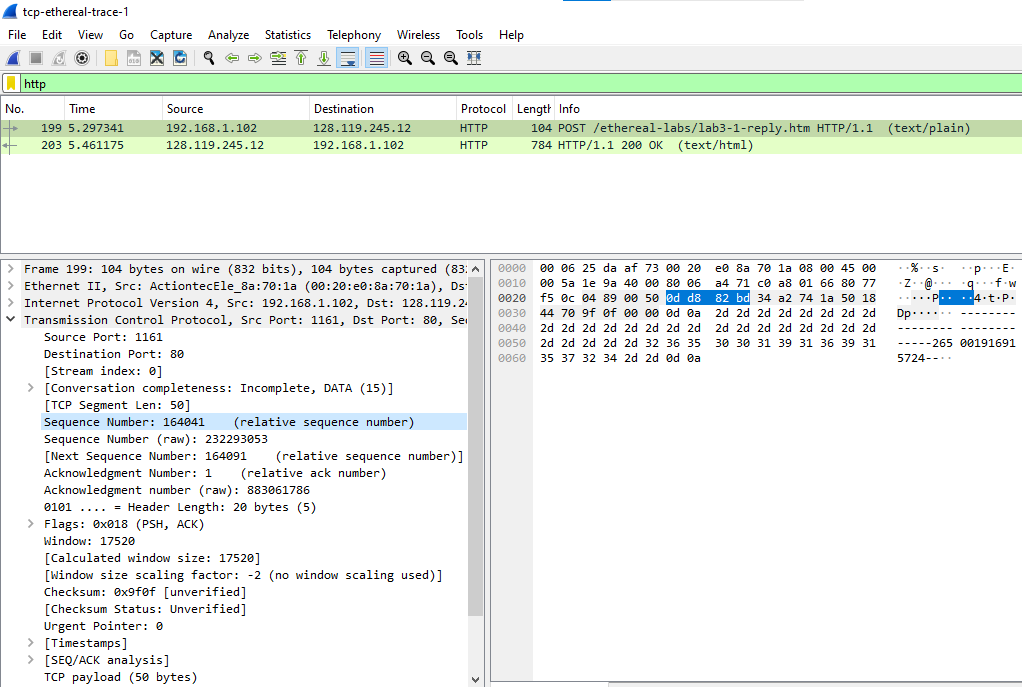
Note that in order to find the POST command, you’ll need to dig into the packet content

field at the bottom of the Wireshark window, looking for a segment with a “POST” within

its DATA field.

**Solution:**

The sequence number of the TCP segment containing the HTTP Post command is 164041.



**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.

**Solution:**

After receiving the ACK for segment 1, the Estimated Round-Trip Time (EstimatedRTT) is calculated as follows:

EstimatedRTT = RTT for Segment 1 = 0.02746 seconds.

After the ACK for segment 2 is received, the EstimatedRTT becomes:

EstimatedRTT = 0.875 \* 0.02746 + 0.125 \* 0.035557 = 0.0285.

Upon receiving the ACK for segment 3, the EstimatedRTT adjusts to:

EstimatedRTT = 0.875 \* 0.0285 + 0.125 \* 0.070059 = 0.0337.

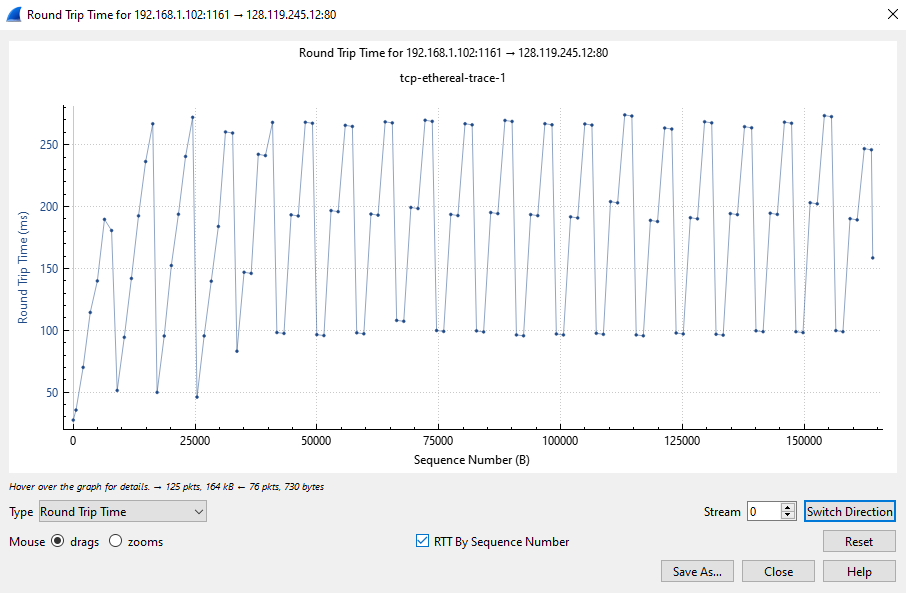
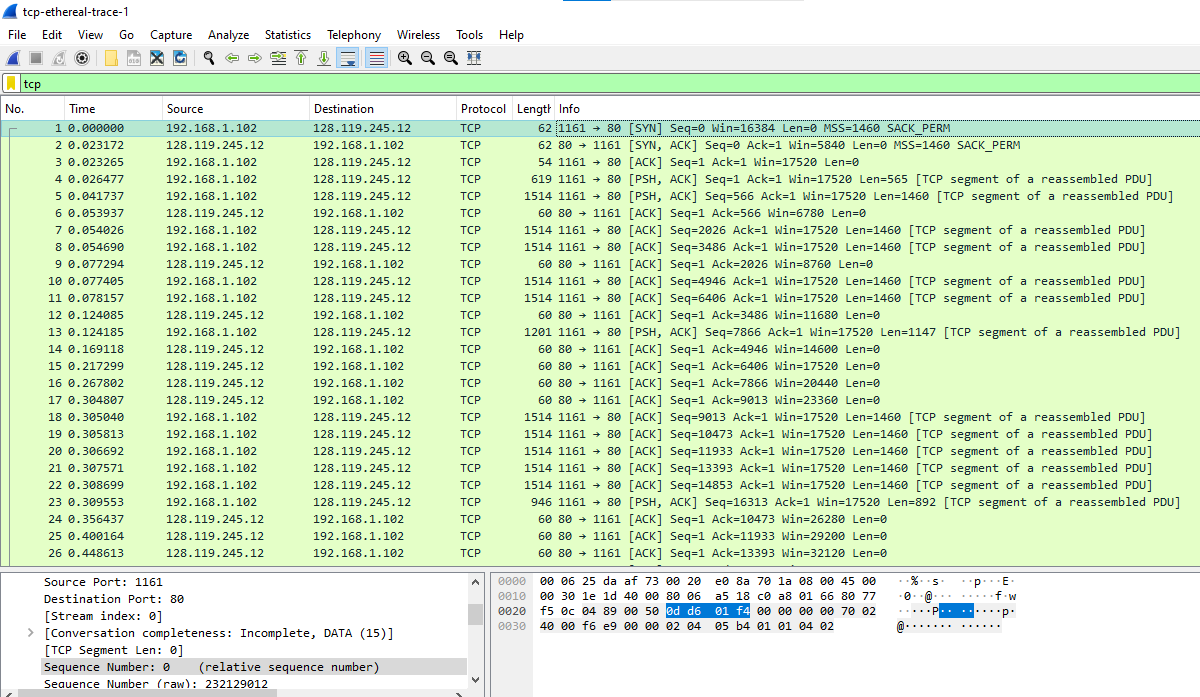
Following the receipt of the ACK for segment 4, the EstimatedRTT becomes:

EstimatedRTT = 0.875 \* 0.0337 + 0.125 \* 0.11443 = 0.0438.

Upon receiving the ACK for segment 5, the EstimatedRTT adjusts to:

EstimatedRTT = 0.875 \* 0.0438 + 0.125 \* 0.13989 = 0.0558.

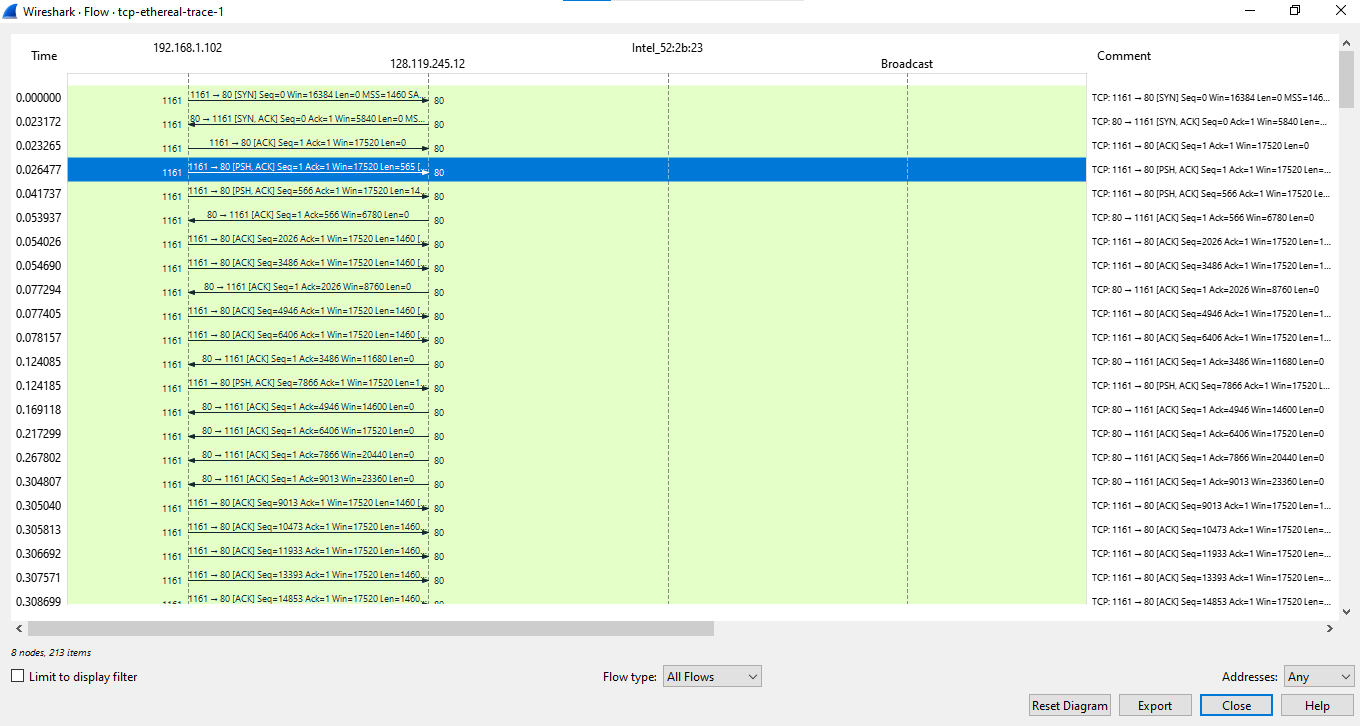
Finally, after the ACK for segment 6 is received, the EstimatedRTT becomes:

EstimatedRTT = 0.875 \* 0.0558 + 0.125 \* 0.18964 = 0.0725.

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

**Solution:**

The total size of the segments is 5848 bytes, with the sizes of each segment as follows: 565 bytes for the first segment, 1460 bytes for the second segment, 1147 bytes for the third segment, 892 bytes for the fourth, fifth, and sixth segments.

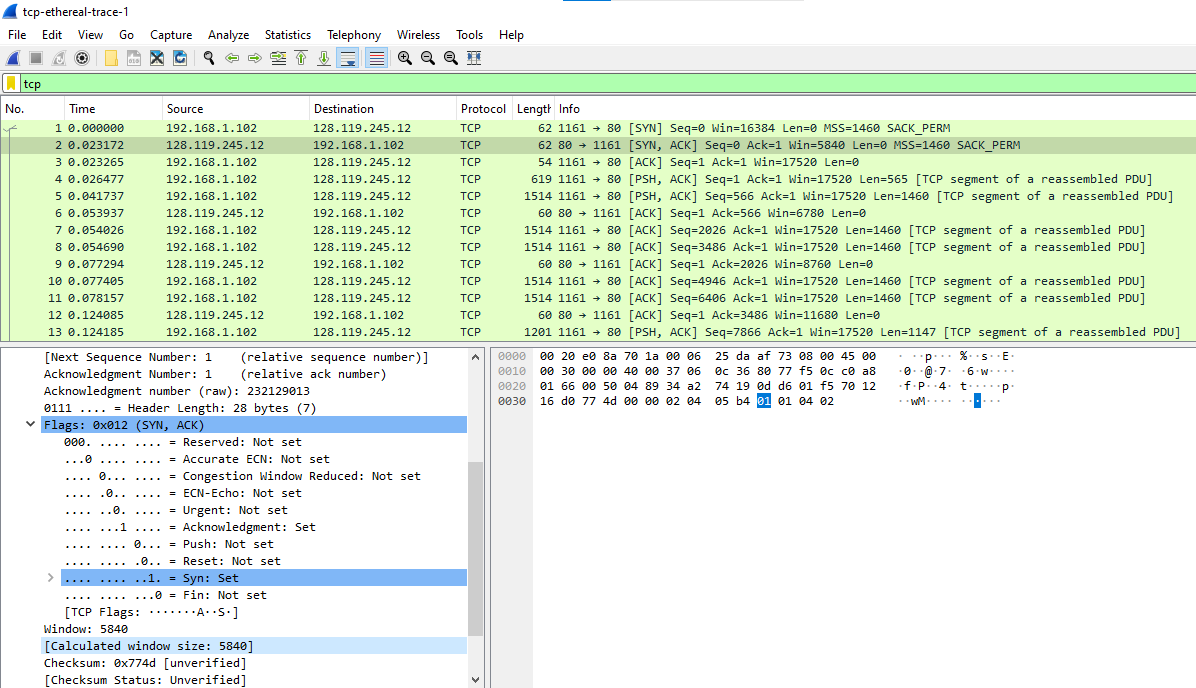


**9.** What is the minimum amount of available buffer space advertised at the received for the

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

**Solution:**

The minimum amount of available buffer space advertised by the receiver throughout the trace is 5480 bytes, as indicated in the first ACK received from the server.



**10.** Are there any retransmitted segments in the trace file? What did you check for (in the trace)

in order to answer this question?

**Solution:**

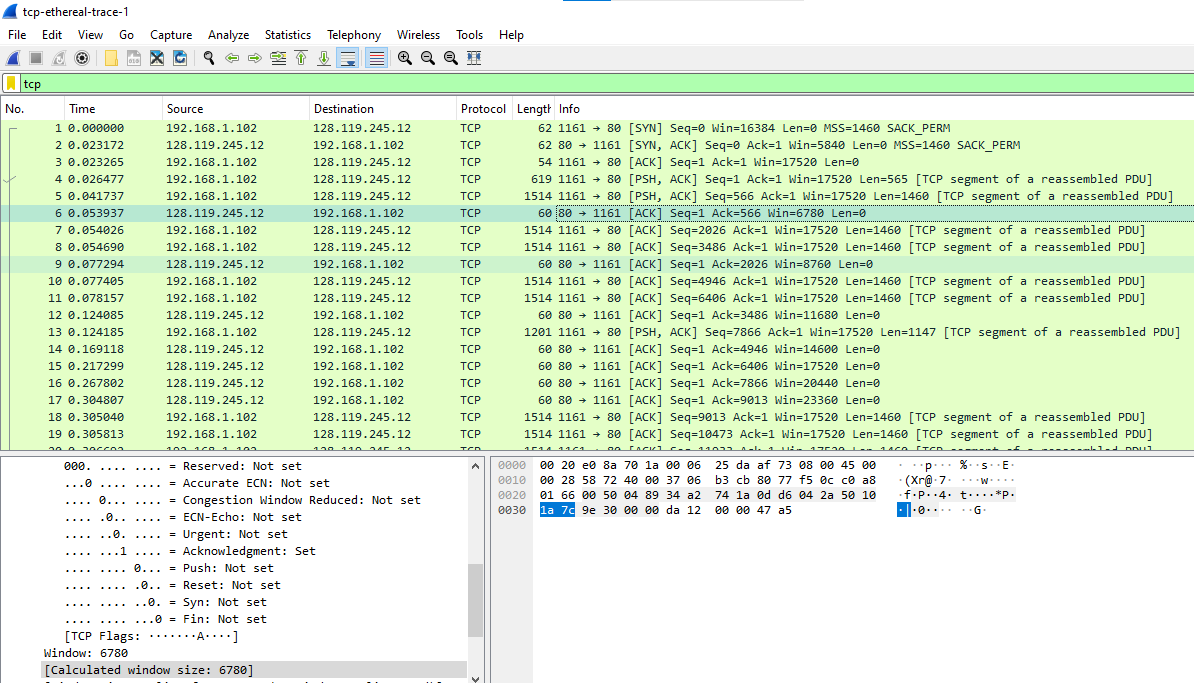
No segments are retransmitted in the trace file, as evidenced by the time sequence graph (Stevens), where all sequence numbers demonstrate a monotonically increasing pattern.

**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).

**Solution:**

The ACK numbers progress in a sequence of 566, 2026, 3486, 4946 and so on, increasing by 1460 each time. This consistent increase indicates that the receiver acknowledges 1460 bytes with each acknowledgment.

**12.** What is the throughput (bytes transferred per unit time) for the TCP connection? Explain

how you calculated this value.

**Solution:**

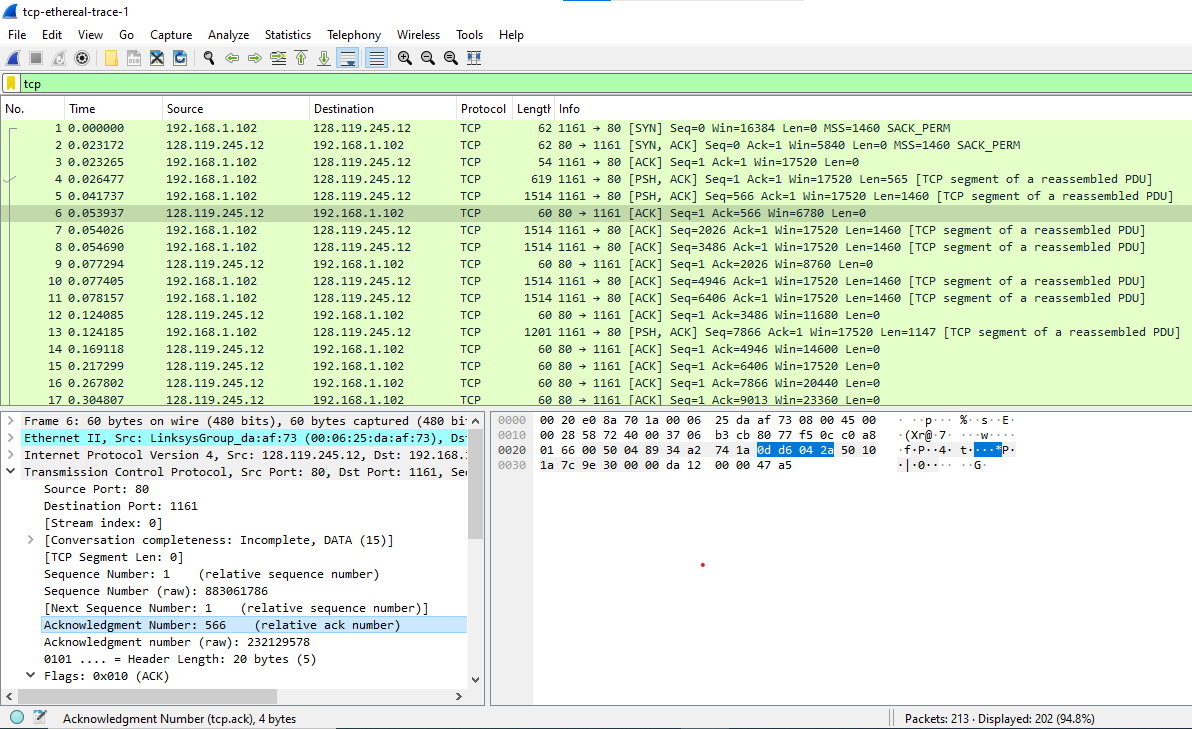
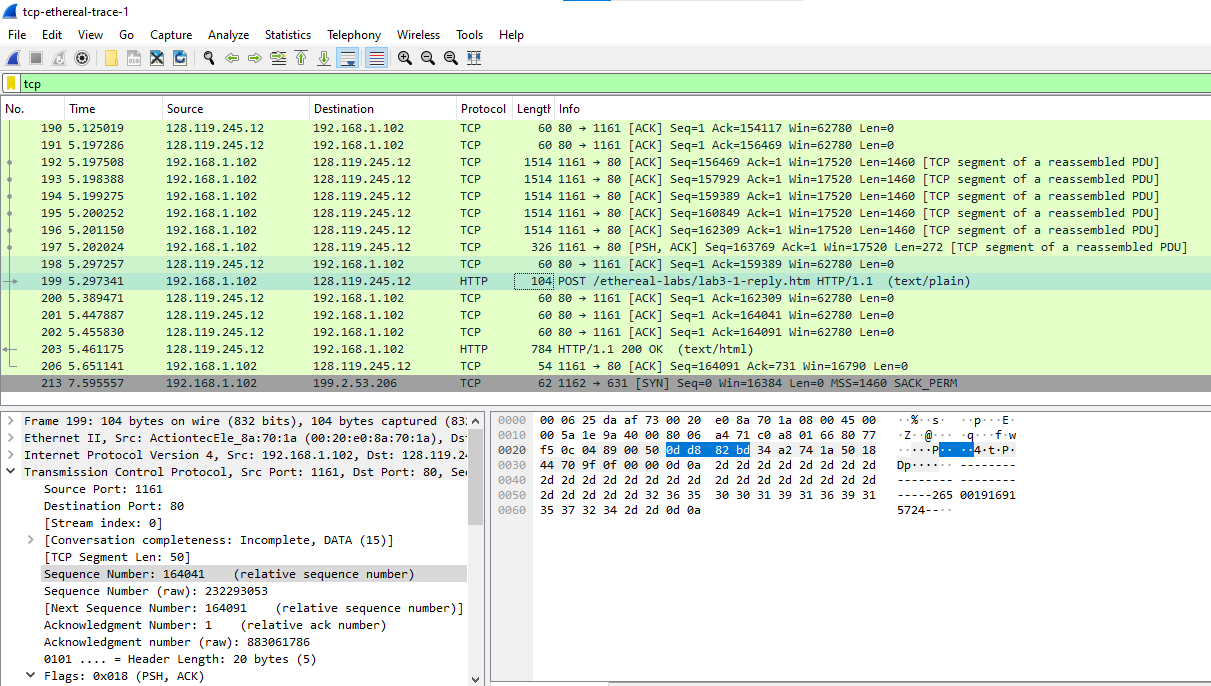
The transmission began at 0.053937 seconds with the first segment and concluded at 5.297341 seconds with the last segment.

The total amount of data transmitted is 164,041 bytes.

Time incurred = 5.297341-0.053937 =5.243404

Throughput =

Throughput =

Throughput = **0.673945 bytes** OR Throughput = **6.739 kbytes/sec**

**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.

**Solution:**

By examining the plot, it's evident that the slow-start phase persists only for the initial 1-1.5 seconds. Subsequently, the TCP session appears to remain in a congestion avoidance state. However, contrary to the anticipated linear growth behavior, the TCP transmit window does not exhibit a linear increase during this phase. Instead, it seems that the sender transmits packets in batches of 6. This behavior doesn't seem to stem from flow control since the receiver's advertised window is considerably larger than 5 packets. The observed pattern might be attributed to a rate-limit enforced by the HTTP server.