Wireshark Lab 4

TCP

Suraj Durgesht

(16EC06)

Instructor: Dr. Rajendra Hegadi

A first look at the captured trace

Que1. 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?

solution: Client computer (source)

IP address: 192.168.1.102 TCP port number: 1161

Que2. 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: Destination computer:

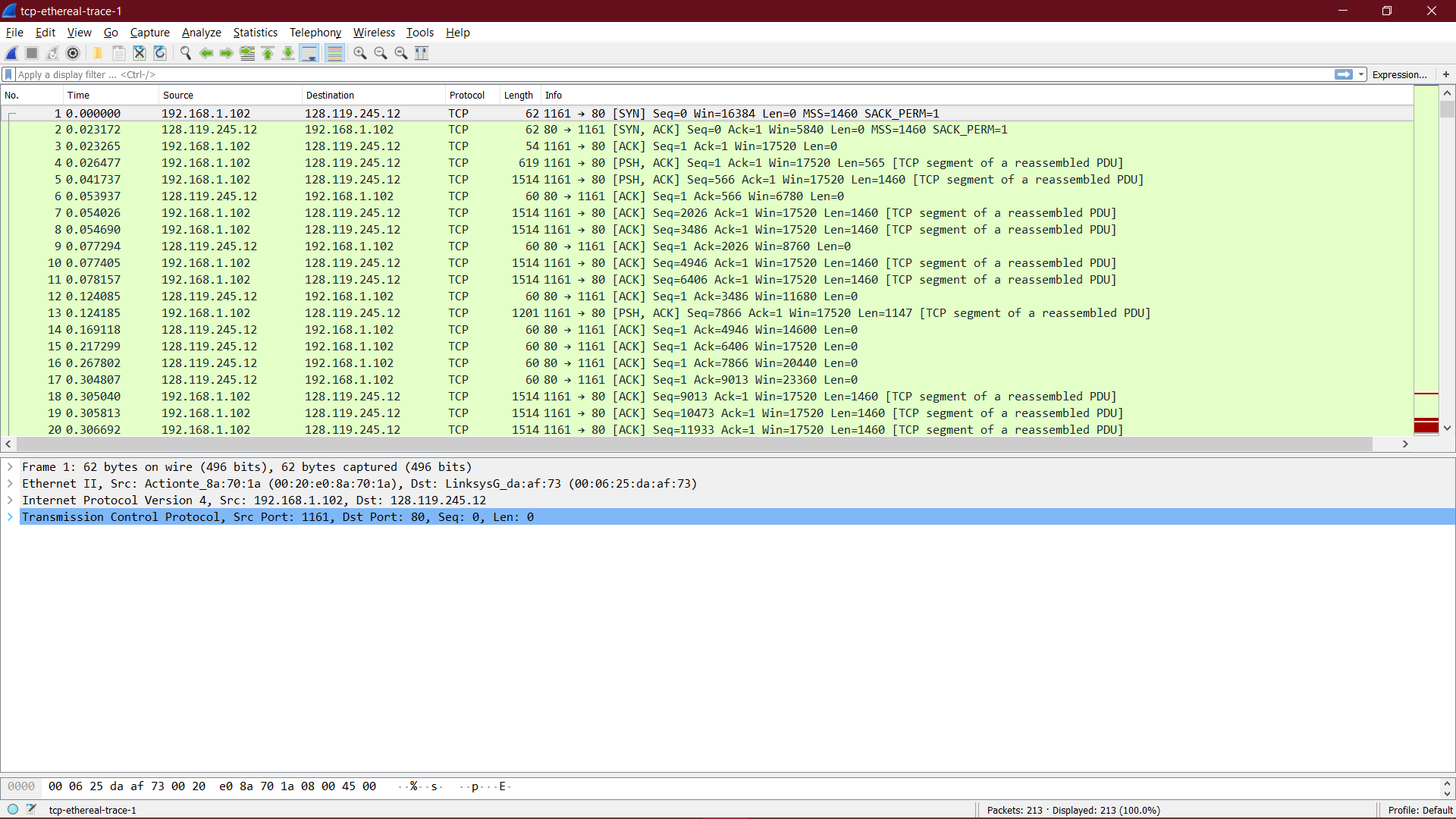
gaia.cs.umass.edu

IP address: 128.119.245.12 TCP port number: 80

Que3. 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:



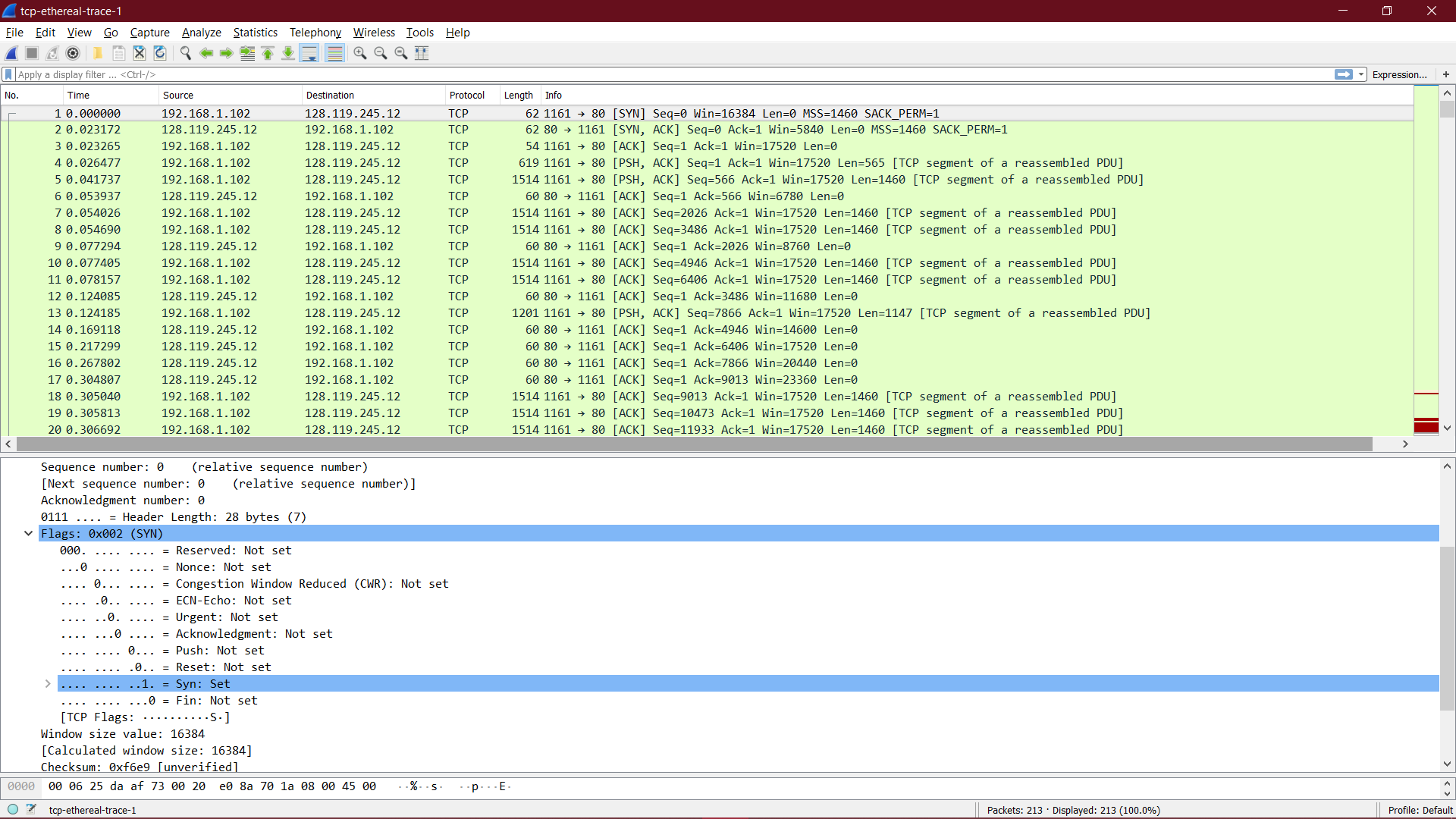
TCP Basics

Que4. 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:

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.

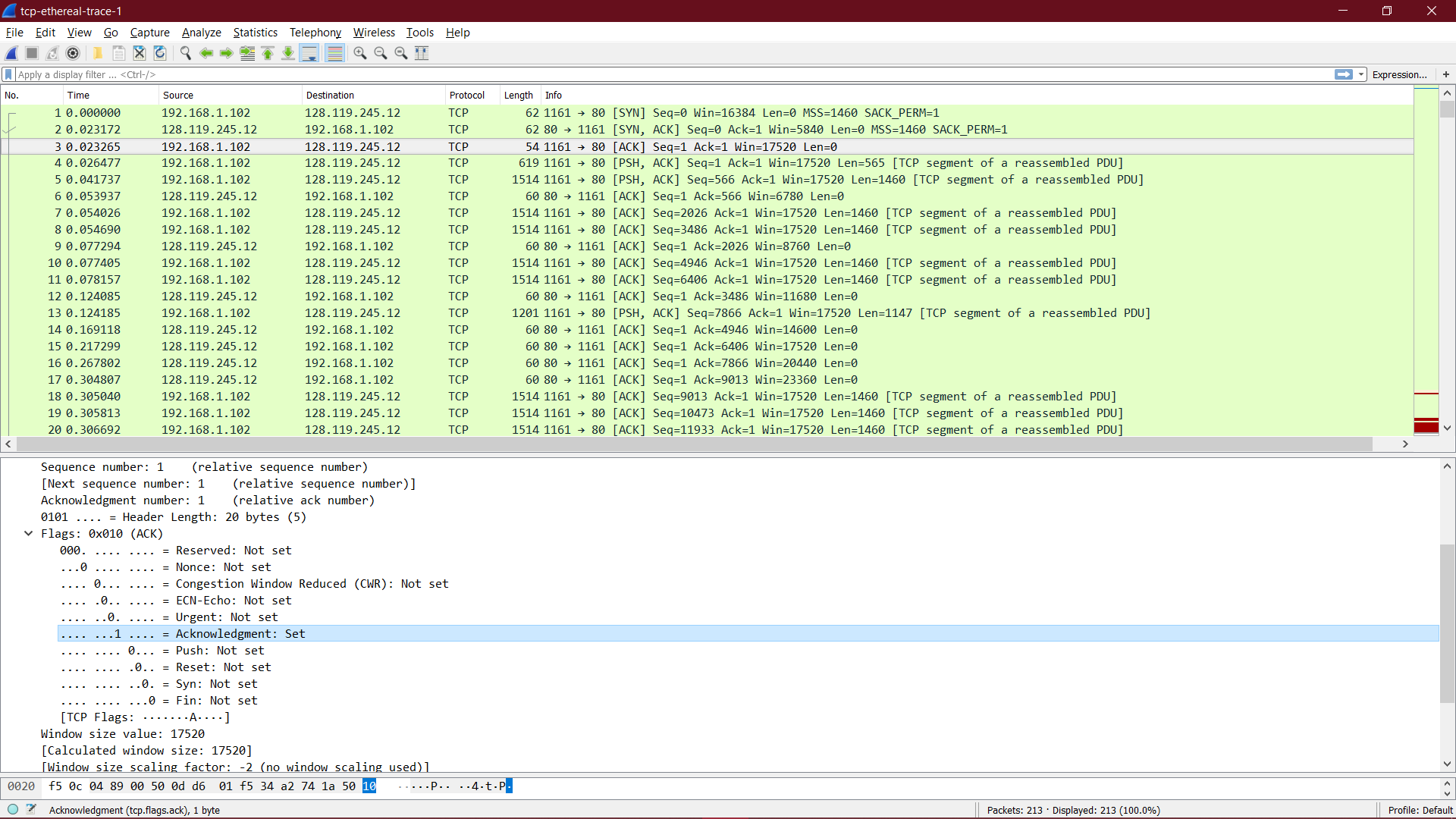


Que5. What is the sequence number of the SYNACK segment sent by gaia.cs.umass.edu

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:

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.



Que6. 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: No. 4 segment is the TCP segment containing the HTTP POST command. The sequence number of this segment has the value of 1.

Que7. 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 after the receipt of each ACK?

Solution: 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

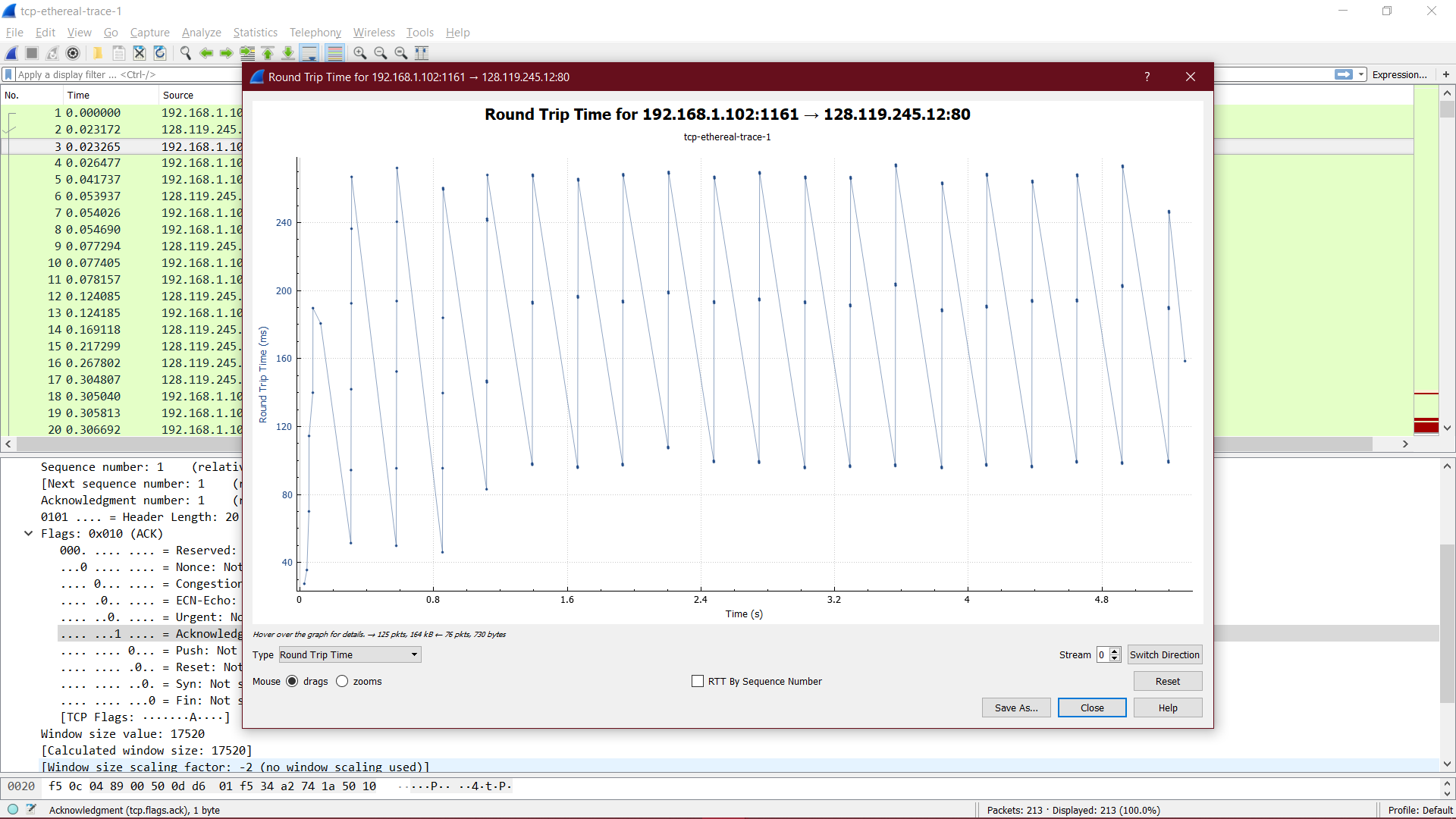
Segment 2 sequence number: 566

Segment 3 sequence number: 2026

Segment 4 sequence number: 3486

Segment 5 sequence number: 4946

Segment 6 sequence number: 6406



Que8. What is the length of each of the first six TCP segments?4

Solution: Length of the first TCP segment (containing the HTTP POST): 565 bytes

Length of each of the other five TCP segments: 1460 bytes (MSS).

Que9. 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 buffer space (receiver window) 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.

Que10. Are there any retransmitted segments in the trace file? What did you check for in order to answer this question?

Solution: There are no retransmitted segments in the trace file. We can verify this by

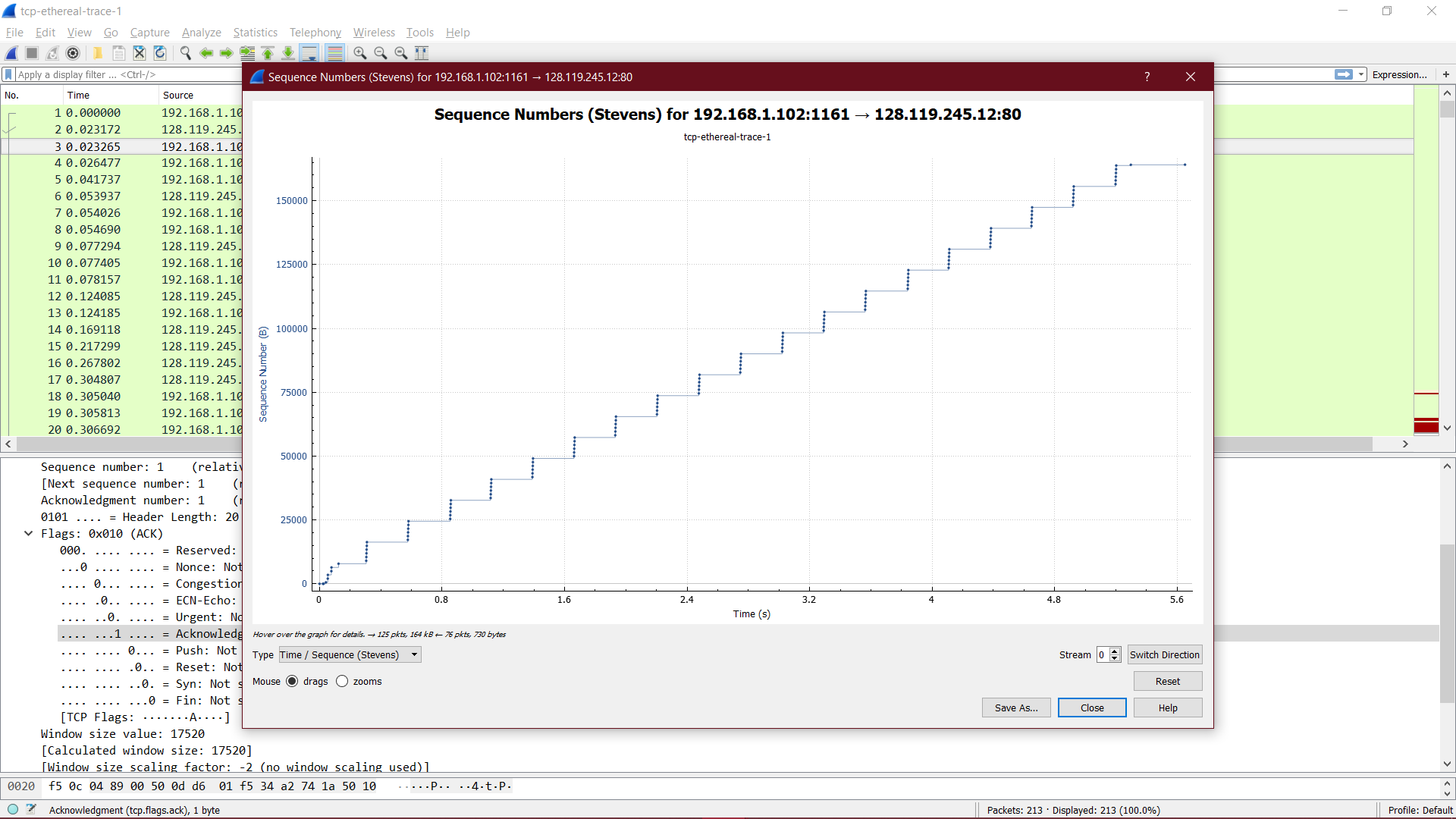
checking the sequence numbers of the TCP segments in the trace file. In the *Time-*

*Sequence-Graph (Stevens*) 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 neighbouring segments.



Que11. How much data does the receiver typically acknowledge in an ACK? Can you

identify cases where the receiver is ACKing every other received segment?

Solution: The difference between the acknowledged sequence numbers of two consecutive ACKs indicates the data received by the server between these two ACKs. By inspecting the

amount of acknowledged data by each ACK, there are cases where the receiver is ACKing every other segment. For example, segment of No. 80 acknowledged data with

2920 bytes = 1460\*2 bytes.

Que12. What is the throughput (bytes transferred per unit time) for the TCP connection?

Explain how you calculated this value.

Solution: The 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. Then, 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 (i.e. 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 (i.e., 0.026477 second for No.4 segment) and the time instant of the last ACK (i.e., 5.455830 second for No. 202 segment). Therefore, the total transmission time is 5.455830 - 0.026477 = 5.4294 seconds. Hence, the throughput for the TCP connection is computed as 164090/5.4294 = 30.222 KByte/sec.

TCP congestion control in action

Que13. 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 slow start phase begins

and ends, and where congestion avoidance takes over? Comment on ways in

which the measured data differs from the idealized behaviour of TCP that we’ve

studied in the text.

Solution:

By observing the plot, we can see that the slow-start phase only lasts for first 1-1.5 second. Afterwards, it seems that the TCP session is always in congestion avoidance state. In this case, we do not observe the expected linear increase behaviour, i.e. the TCP transmit window does not grow linearly during this phase. In fact, it appears that the sender transmits packets in batches of 6. This does not seem to be caused by flow control since the receiver advertised window is significantly larger than 5 packets. The reason for this behaviour might be due to the fact that the HTTP server has enforced a rate-limit of some sort.

