CS315 : Computer Networks Lab

Assignment 5

Sourabh Bhosale (200010004)

January 31, 2023

1 Part 1: Capturing a bulk TCP transfer from your computer to a remote server

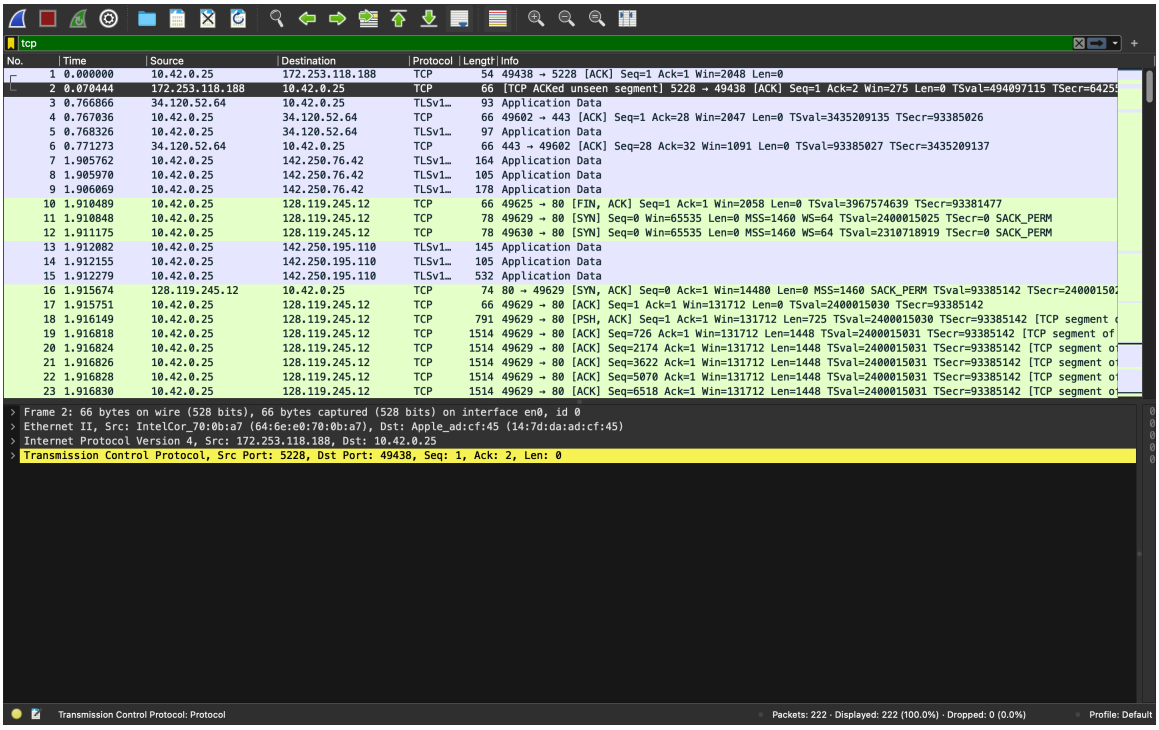


Figure 1: `Wireshark window : Part-1`

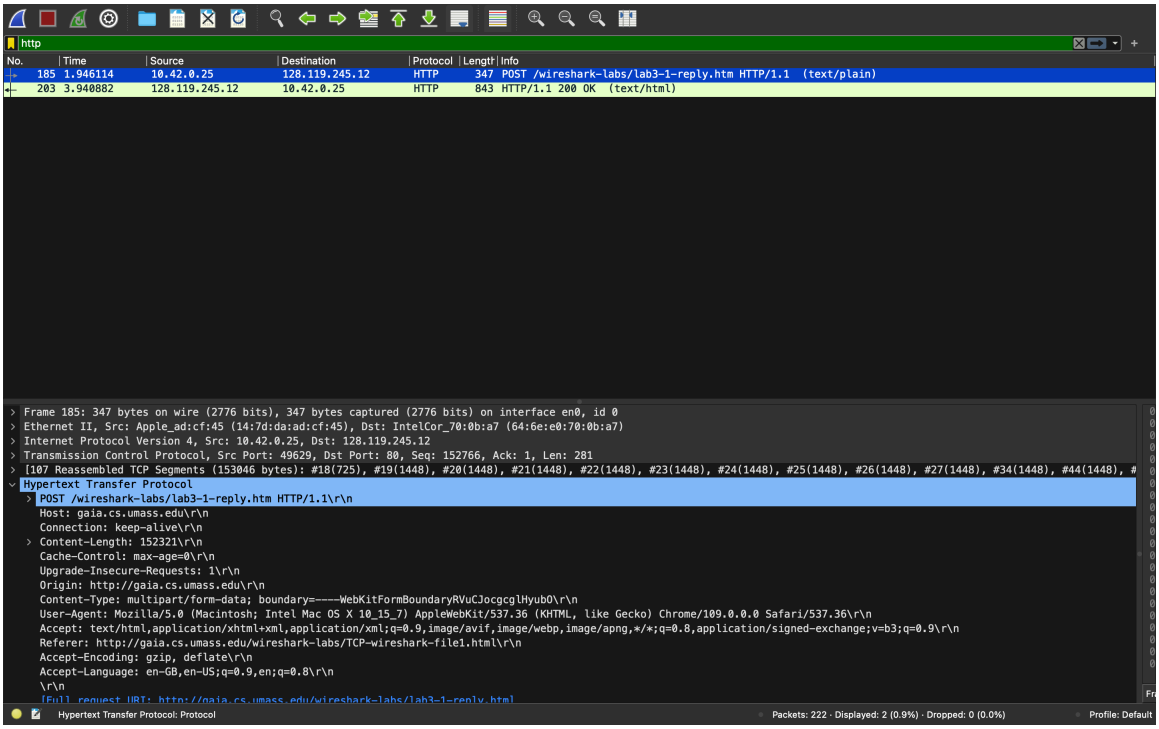
2 Part 2: A first look at the captured trace

1. What is the IP address and TCP port number used by the client computer (source) that is transferring the alice.txt 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"

IP address (source) : 10.42.0.25
TCP port number (source) : 49629

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?

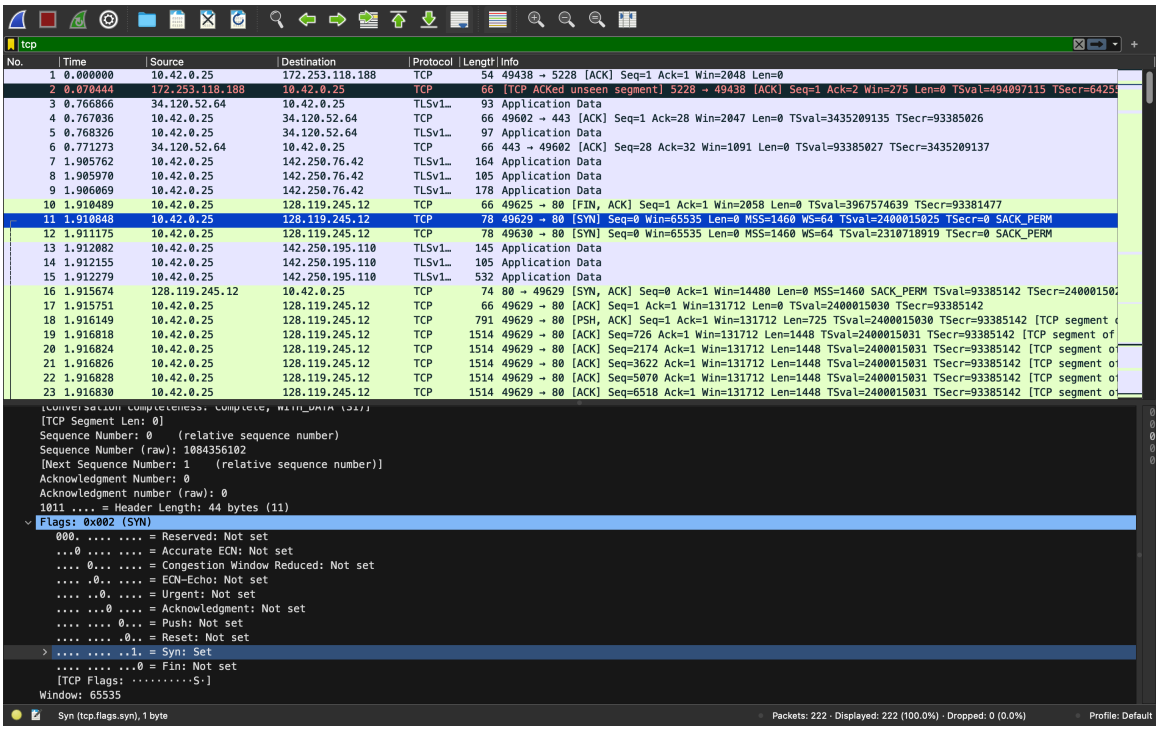
IP address (destination) : 128.119.245.12
TCP port number (destination) : 80



3 Part 3: TCP Basics

1. 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 this TCP segment that identifies the segment as a SYN segment?

Sequence number : 0. In the Flags section, the `Syn` flag is set to 1 which indicates that this segment is a SYN segment.



2. 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 it in the segment that identifies the segment as a SYNACK segment? What is the value of the Acknowledgement field in the SYNACK segment? How did `gaia.cs.umass.edu` determine that value?

Sequence number : 0. A segment will be identified as a SYNACK segment if both SYN flag and Acknowledgement in the segment are set to 1. The value of the acknowledgement field in the SYNACK segment is 1. We can say that, `gaia.cs.umass.edu` determines that value as the server adds 1 to the initial sequence number of SYN segment form the client computer. In this case, the initial sequence number of SYN segment from the client computer is 0, thus the value of the Acknowledgement field in the SYNACK segment is 1.

3. What is the sequence number of the TCP segment containing the header of the HTTP POST command? Note that in order to find the POST message header, you'll need to dig into the packet content field at the bottom of the Wireshark window, looking for a segment with the ASCII text "POST" within its DATA field. How many bytes of data are contained in the payload (data) field of this TCP segment? Did all of the data in the transferred file alice.txt fit into this single segment?

Sequence number : 152766. 281 bytes of data are contained in the payload (data) field of this TCP segment. No, all of the data in the transferred file alice.txt doesn't fit into this single segment.

4. Consider the TCP segment containing the HTTP "POST" as the first segment in the data transfer part of the TCP connection.

i. At what time was the first segment (the one containing the HTTP POST) in the data-transfer part of the TCP connection sent?

Time : 1.916149 seconds.

ii. At what time was the ACK for this first data-containing segment received?

Time : 1.919794 seconds.

iii. What is the RTT for this first data-containing segment?

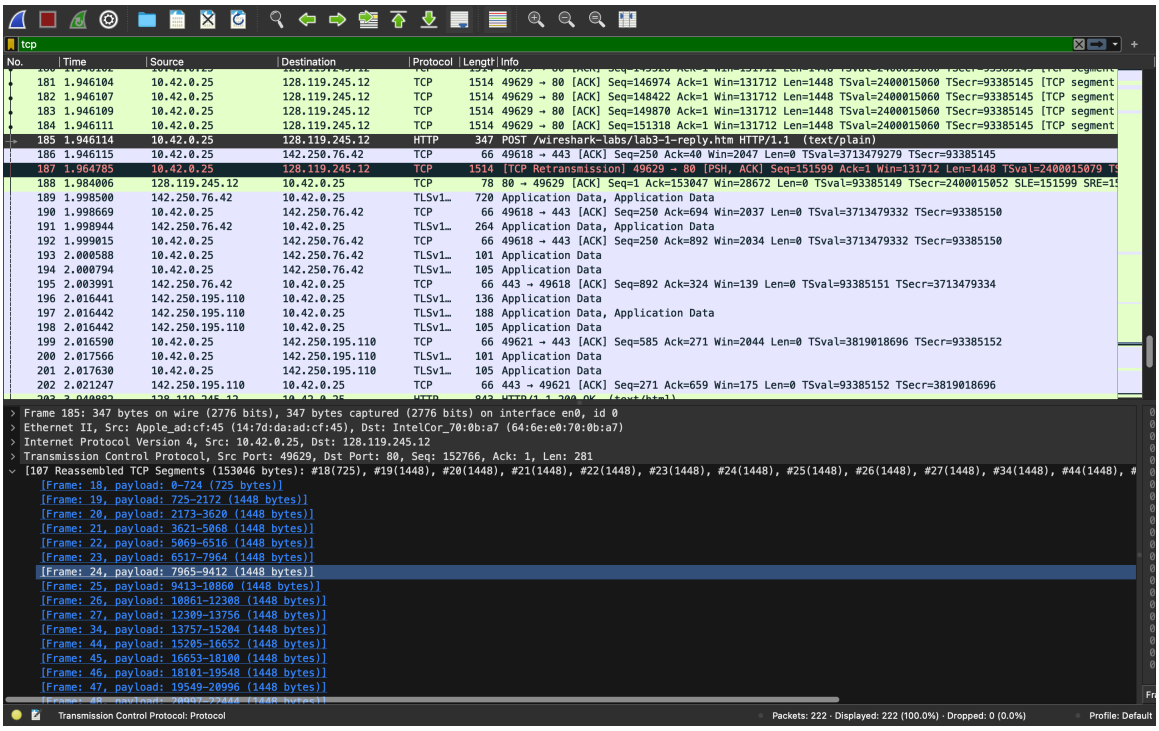
The RTT for this segment was: 0.003645000 seconds.

iv. What is the RTT value of the second data-carrying TCP segment and its ACK?

The RTT to this segment was: 0.005061000 seconds.

5. What is the length (header + payload) of each of the first four data-carrying TCP segments?

Length of each of the first four data-carrying TCP segments is (32 + 725) 757, (32 + 1448) 1480, (32 + 1448) 1480 and (32 + 1448) 1480 bytes respectively.



6. What is the minimum amount of available buffer space advertised to the client by `gaia.cs.umass.edu` among these first four data-carrying TCP segments? Does the lack of receiver buffer space ever throttle the sender for these first four data-carrying segments?

The minimum amount of available buffer space (receiver window) advertised at gaia.cs.umass.edu is 16000 bytes, which shows in the first acknowledgement from the server. This receiver window grows steadily. The sender is never throttled due to lacking of receiver buffer space by inspecting this trace.

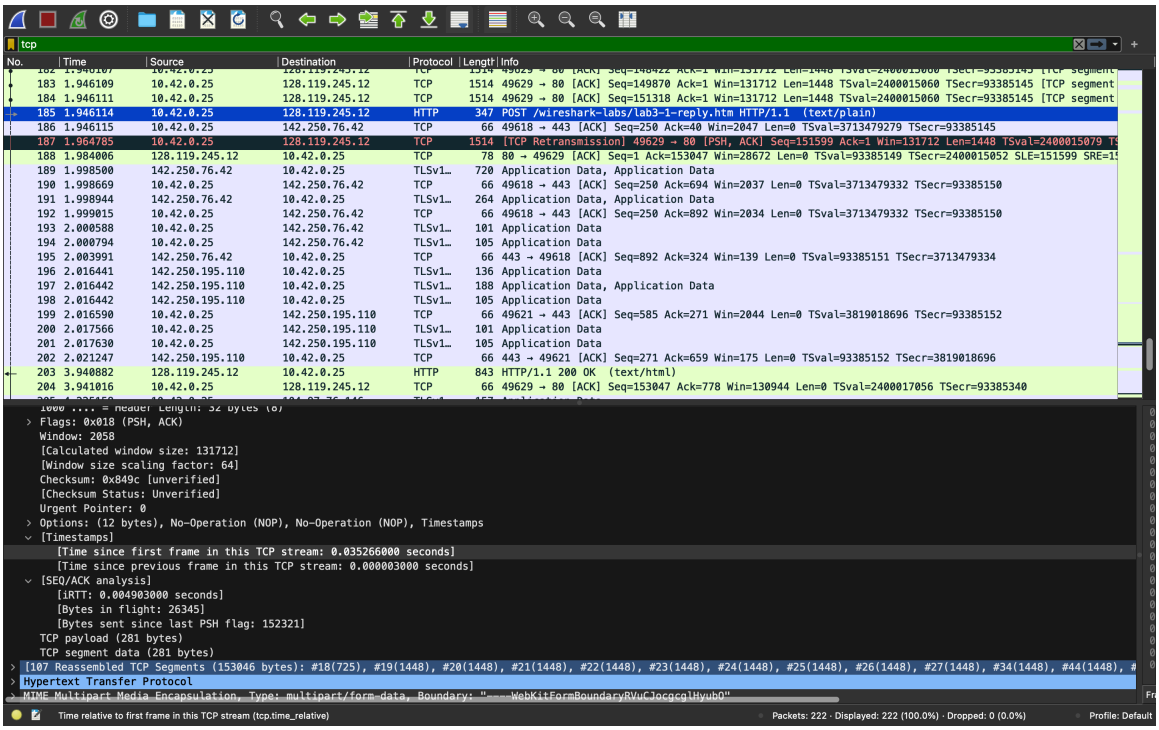
7. Are there any retransmitted segments in the trace file? What did you check for (in the trace) in order to answer this question?

There are no retransmitted segments in the trace file. 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 to the destination 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.

8. How much data does the receiver typically acknowledge in an ACK among the first ten data-carrying segments sent from the client to gaia.cs.umass.edu? Can you identify cases where the receiver is ACKing every other received segment among these first ten data-carrying segments?

The difference between the acknowledged sequence numbers of two consecutive ACKs indicates the data received by the server between thesetwo ACKs. The receiver typically acknowledges 1460 bytes in an ack. By inspecting the amount of acknowledged data by each ACK, I haven't seen any ACKing between every other received segment.

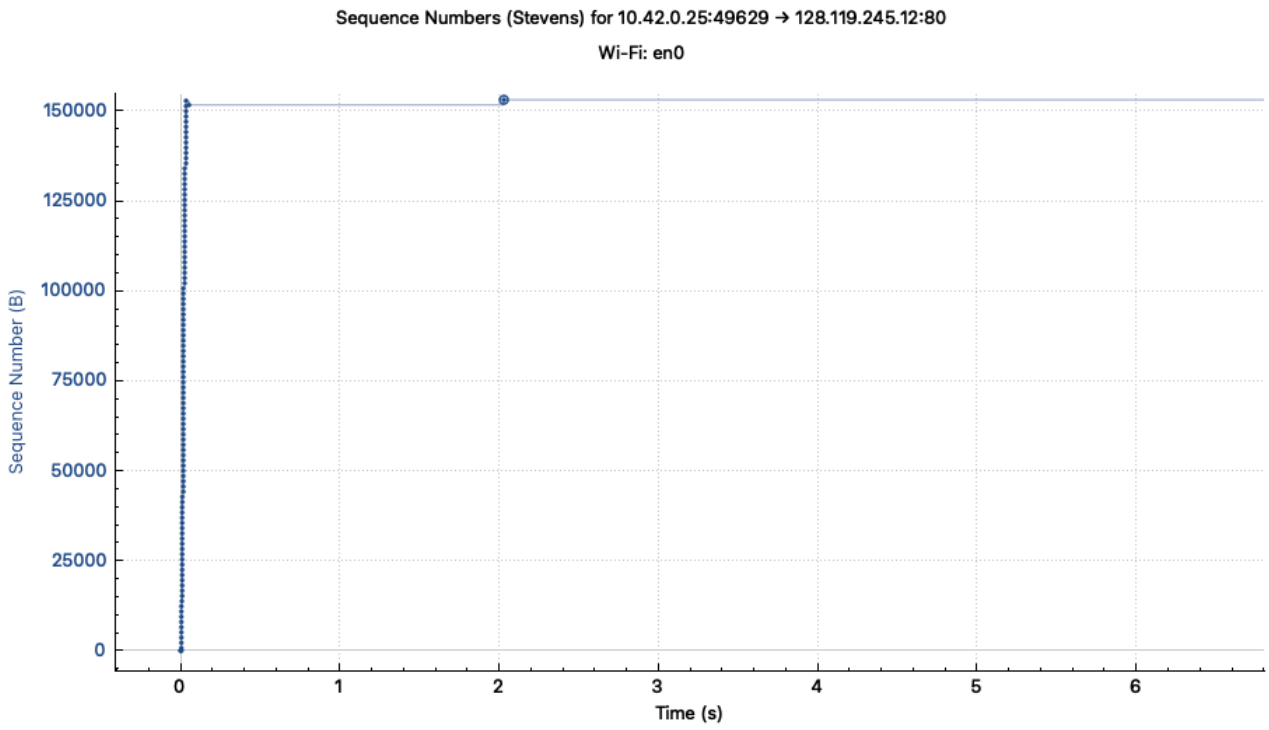
9. What is the throughput (bytes transferred per unit time) for the TCP connection? Explain how you calculated this value.



Total no. of bytes transferred is 153048 and the diffrence between first frame (0.035266000 seconds) and the last (0.000003000 seconds) came to be 0.035263 seconds. So the throughput will be 153048 bytes / 0.035263 seconds = 4340186.59785 bytes / seconds.

4 Part 4: TCP congestion control in action

1. 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?



Slow start phase begins at around 0 seconds and ends at around 0.04 seconds and then congestion takes over.