CS315: Assignment-5

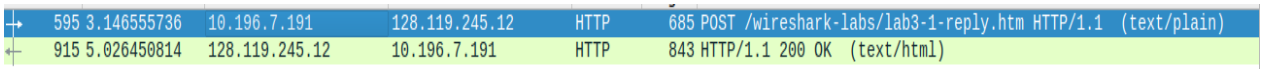
Name: Shahil Patel
Roll No.: 200010039

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".

Ans: Source IP address: ***10.196.7.191.***

Source TCP port number used to transfer the alice.txt file to gaia.cs.umass.edu: ***53950.***



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?

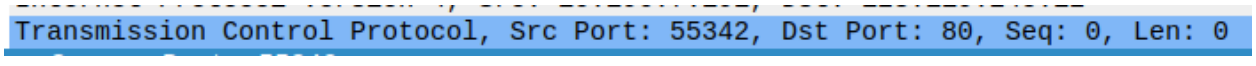
Ans: IP address of *gaia.cs.umass.edu:* ***128.119.245.12***

Port number to which it is sending and receiving TCP segments is: ***80***

Part 3: TCP Basics

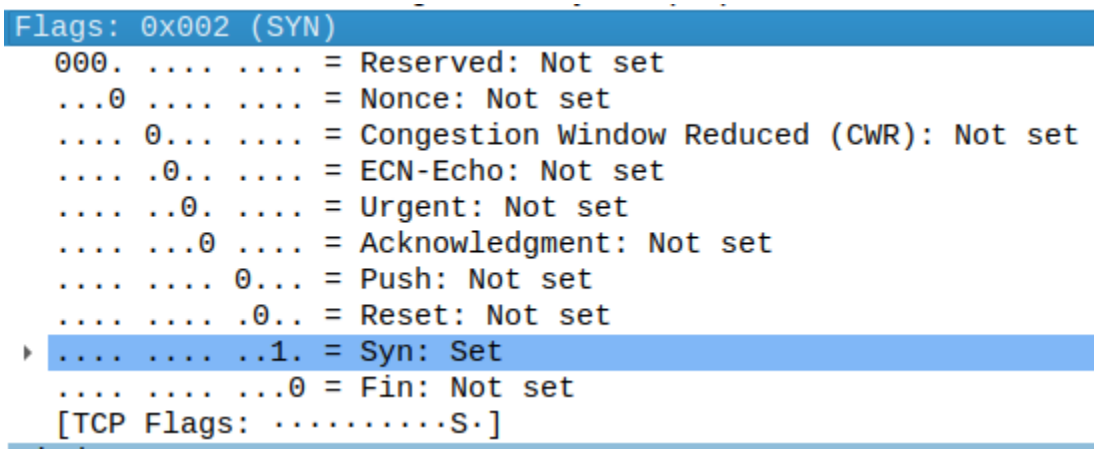
1. What is the *sequence number* of the TCP SYN segment that is used to initiate the TCP connection between the client computer and gaia.cs.umass.edu? What is it in this TCP segment that identifies the segment as a SYN segment?

Sol: 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 is: ***0***



The packet is SYN, because flag under SYN bit is set (i.e. 1)

Content of the flag:

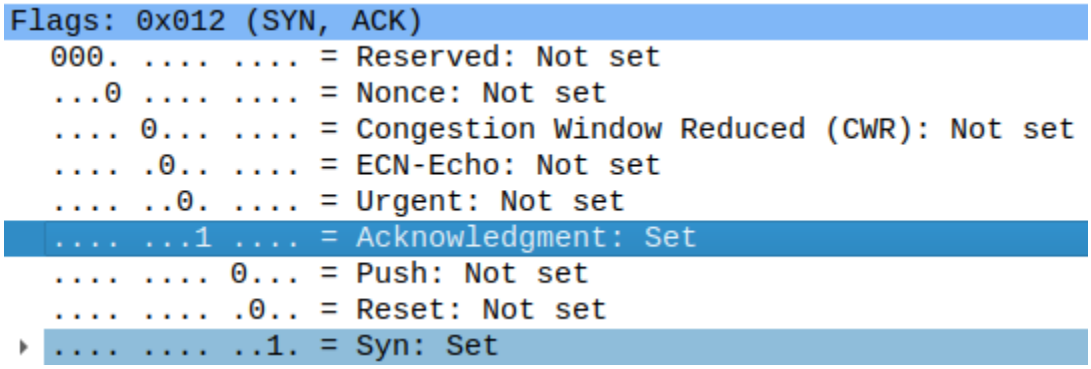


2. What is the *sequence number* of the SYNACK segment sent by gaia.cs.umass.edu to the client computer in reply to the SYN? What is it in the segment that identifies the segment as a SYNACK segment? What is the value of the Acknowledgement field in the SYNACK segment? How did gaia.cs.umass.edu determine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

Ans: The sequence number of the SYNACK segment sent by gaia.cs.umass.edu is: ***0***

The flags under SYN and Acknowledgement bit is set (i.e. 1)

The content of flags are:



The value of the Acknowledgment field in the SYNACK segment is: ***1***

The gaia.cs.umass.edu determines the value by adding 1 to the acknowledgement bit value obtained from the previous request made by client.

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?

Ans: The sequence number of TCP segment containing the header of the HTTP POST command is: 1

There are ***487 bytes*** of data contained in the payload (data) field of this TCP segment.

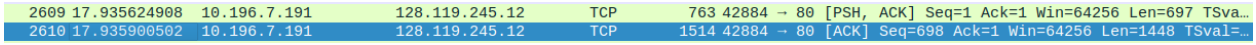
No, All of the data in the transferred file alice.txt would not fit into single segment.

```
Sequence Number: 1    (relative sequence number)
Sequence Number (raw): 257673299
[Next Sequence Number: 488    (relative sequence number)]
Acknowledgment Number: 1    (relative ack number)
Acknowledgment number (raw): 311588656
1000 .... = Header Length: 32 bytes (8)
Flags: 0x018 (PSH, ACK)
Window: 502
[Calculated window size: 64256]
[Window size scaling factor: 128]
Checksum: 0x4171 [unverified]
[Checksum Status: Unverified]
Urgent Pointer: 0
Options: (12 bytes), No-Operation (NOP), No-Operation (NOP), Timestamps
[Timestamps]
[SEQ/ACK analysis]
TCP payload (487 bytes)
```

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?
   ii. At what time was the ACK for this first data-containing segment received?
   iii. What is the RTT for this first data-containing segment?
   iv. What is the RTT value of the second data-carrying TCP segment and its ACK?

Ans: A. The time at which the first segment in the data transfer part of the TCP connection was sent is: ***17.935624908 seconds***

B. The time at which the ACK for the first data-containing segment was received is: ***17.935900502 seconds***



C. The RTT for this first data-containing segment is: ***0.000275594 seconds***

D. The RTT for the second segment is: ***0.000292243 seconds***

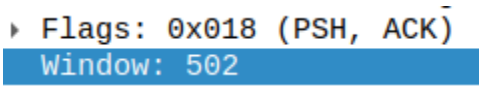
5. What is the length (header plus payload) of each of the first four data-carrying TCP segments?

Ans: The payload for the first 4 data-carrying TCP segments is:

- ***729***
- ***1480***
- ***1480***
- ***1480***

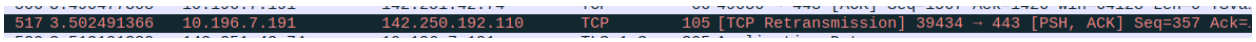
6. What is the minimum amount of available buffer space advertised to the client by gaia.cs.umass.edu among these first four data-carrying TCP segments? Does the lack of receiver buffer space ever throttle the sender for these first four data-carrying segments?

Ans: The minimum amount of available buffer space advertised to client by gaia.cs.umass.edu among this first 4 data-carrying TCP segments is: 502. No, The lack of receiver buffer space wouldn't throttle the sender for this first 4 data-carrying segments.



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

Ans: **Yes**, there are reretransmitted segments in the trace file. We checked for the TCP retransmitted information in the info column



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?

Ans: The receiver typically acknowledge ***1448*** bytes of data in an ACK among the first 10 data-carrying segments sent from the client to gaia.cs.umass.edu. No, there has not been a case where the receiver is ACKing every other received segment among these first ten data-carrying segments.

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

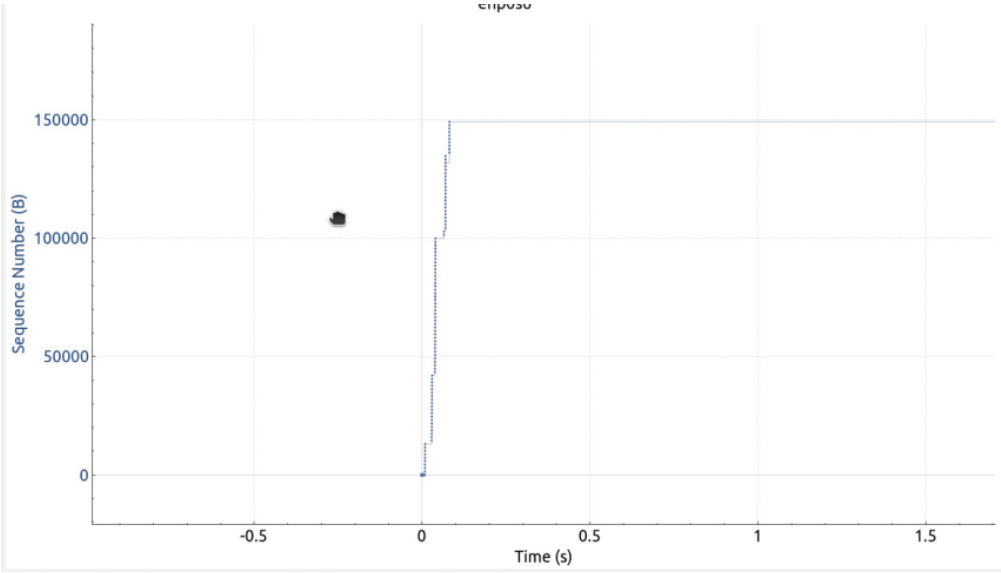
Ans: The throughput for the TCP connection is ***1841108.1410867*** bytes per unit time.

We calculated throughput using the formula = [Reassembled TCP length: *149377*] / [Time since first frame in this TCP stream: *0.081136040* seconds - Time since previous frame in this TCP stream: *0.000001751*].

Part 4: TCP congestion control in action

10. 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?

Ans: At packet 18, when there is an exponential increase, the TCP sluggish start phase starts. At packet 177, the TCP slow start phase ends. Since there are no segments with additive increment, congestion avoidance is not used.


150000
100000
50000
0
Sequence Number (B)
-0.5
0
0.5
1
1.5
Time (s)