

Computer Networks CSE 5344

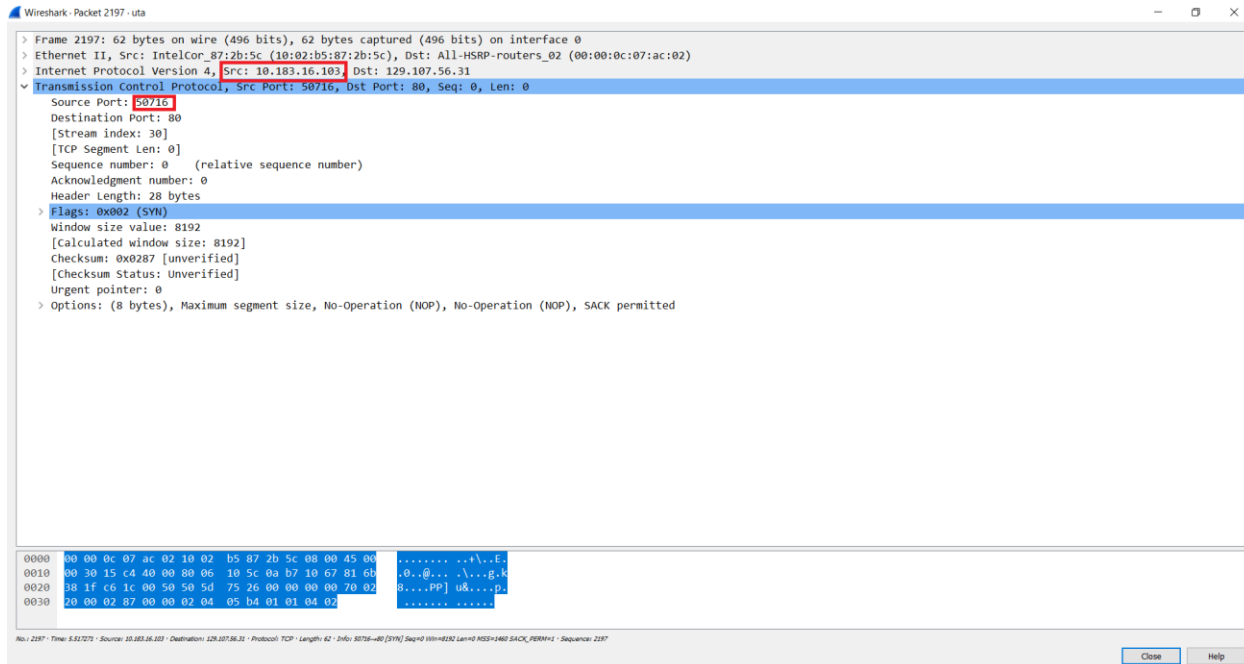
Project 2: Transmission Control Protocol Analysis Using Wireshark

Name: Ritesh Deshmukh

Problem Set 1

1. What is the IP address and TCP port number used by your client computer (source) to browse the page uta.edu.

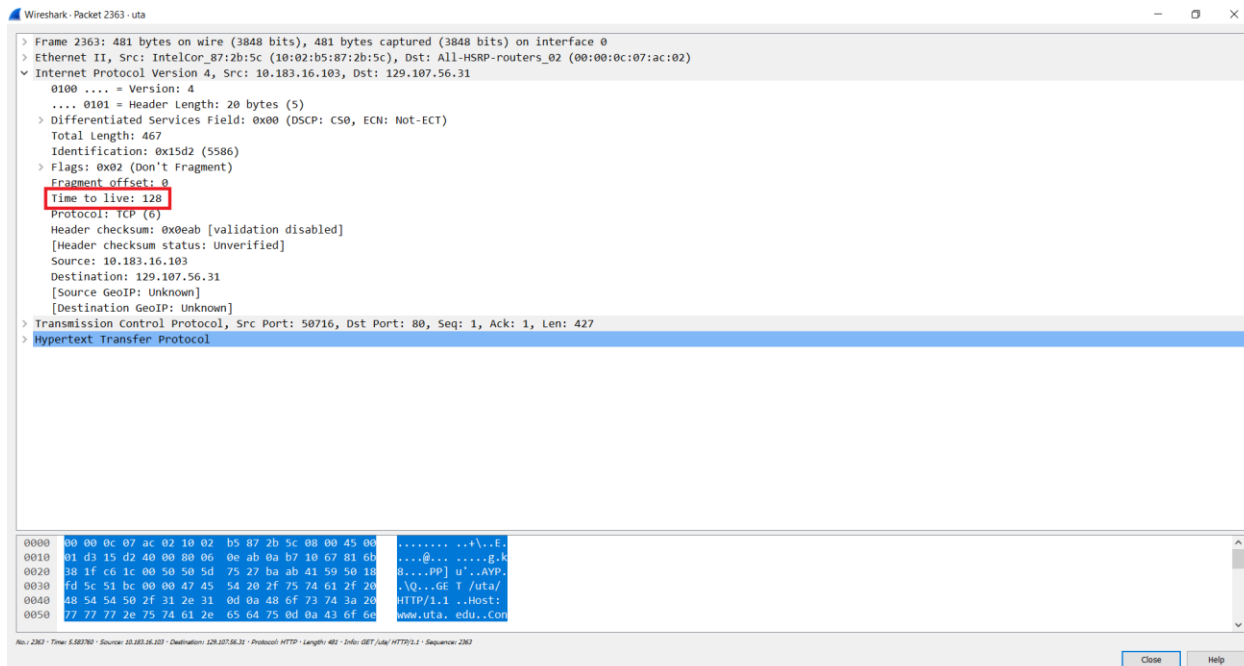
-> IP Address = 10.183.16.103 Port number = 50716



Use the 'GET' message to answer the following questions.

2. What is the TTL value that is used in this communication?

-> 128



3. Did you Use IPV4 or IPV6 for communication?

-> IPV4

4. Does your optional field has some particular information or not.

-> There is not information present in the field specified.

5. Is the Packet Fragmented

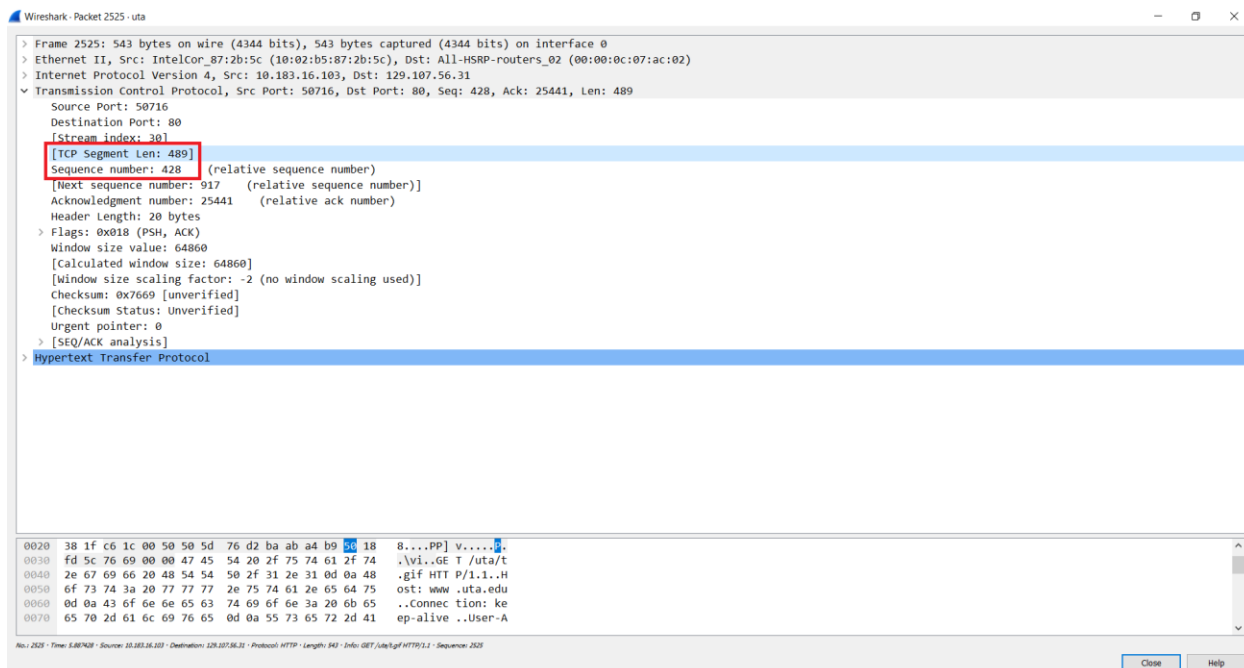
-> Since the Fragment Offset is 0, the packet is not fragmented.

6. What is the TCP segment length?

-> 489

7. What is the Sequence Number of TCP segment (you can use the relative sequence number).

-> 428



8. Calculate the acknowledgement number based on the two questions above. Verify your solution with the Wireshark values.

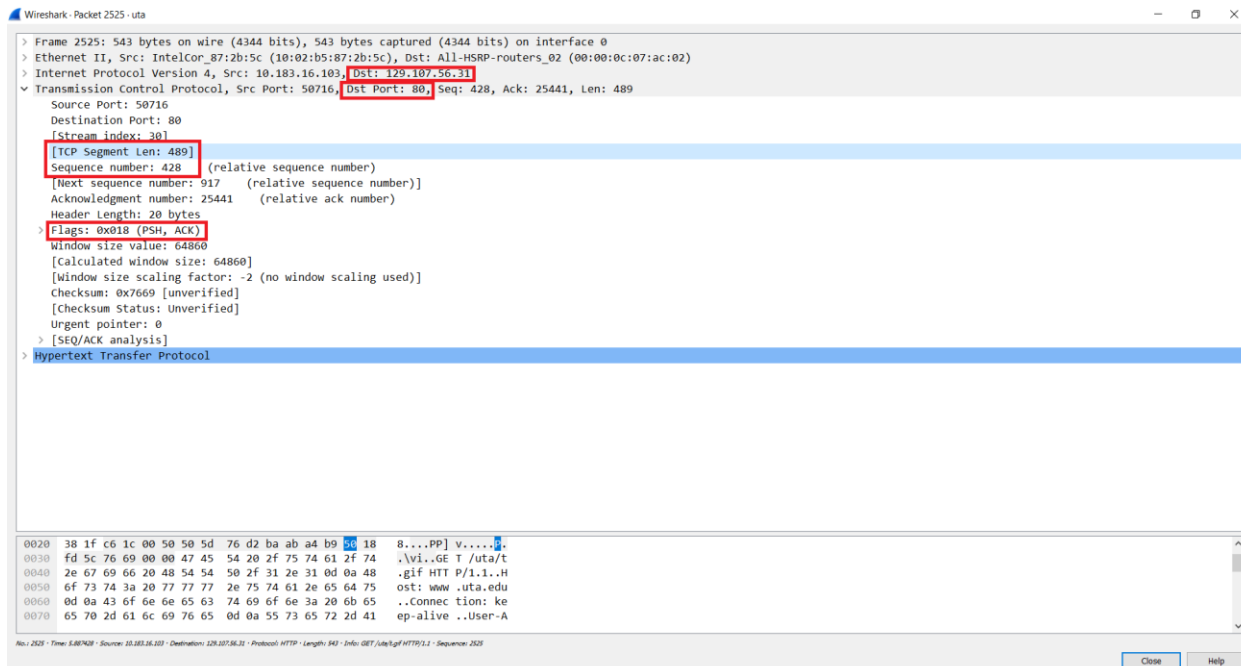
-> 917

9. What are the fields in the TCP Flags. No need to give any values but give the field names given in Wireshark

- > (PSH, ACK)

10. What is the IP address of uta.edu? On what port number is it sending and receiving TCP segments for this connection?

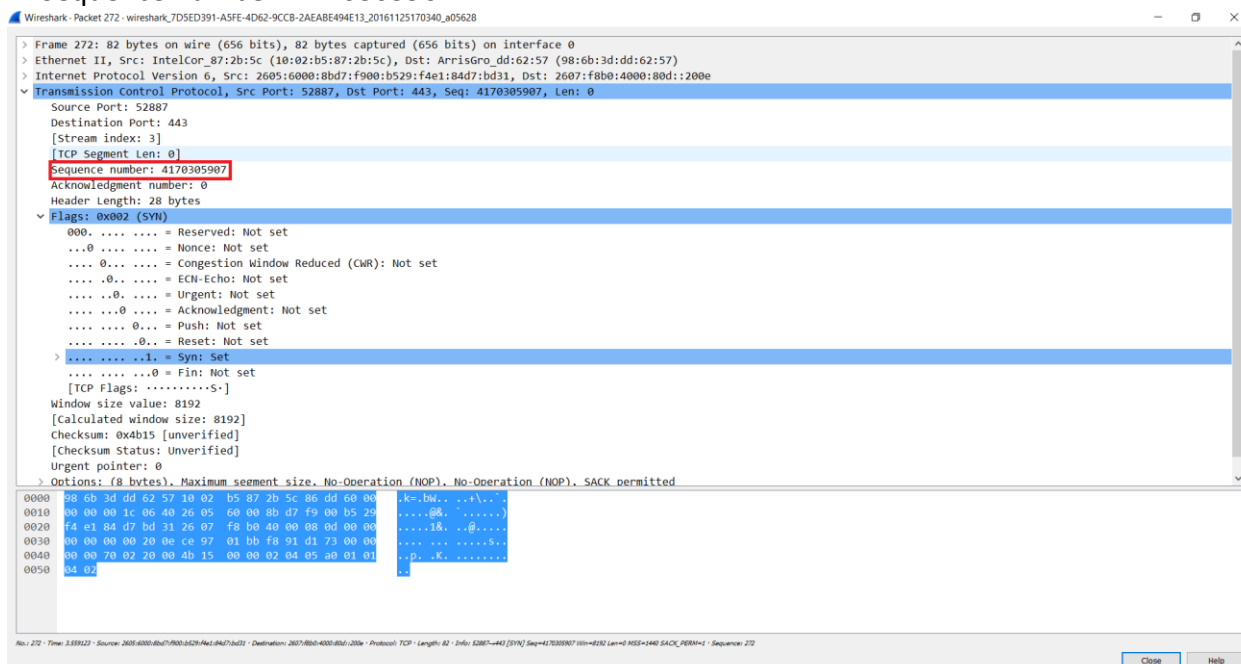
-> IP Address = 129.107.56.31 Port Number = 80



Problem Set 2

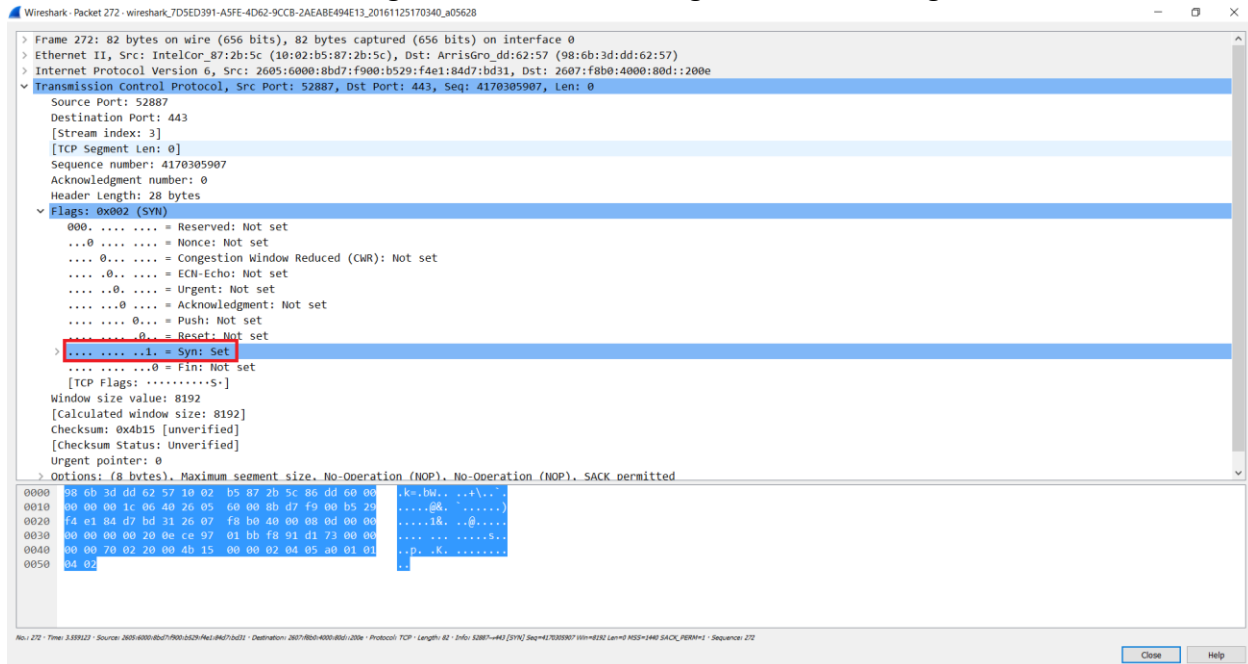
1. What is the sequence number (absolute) of the TCP SYN segment that is used to initiate the TCP connection between the client computer and youtube.com?

-> Sequence number: 4170305907



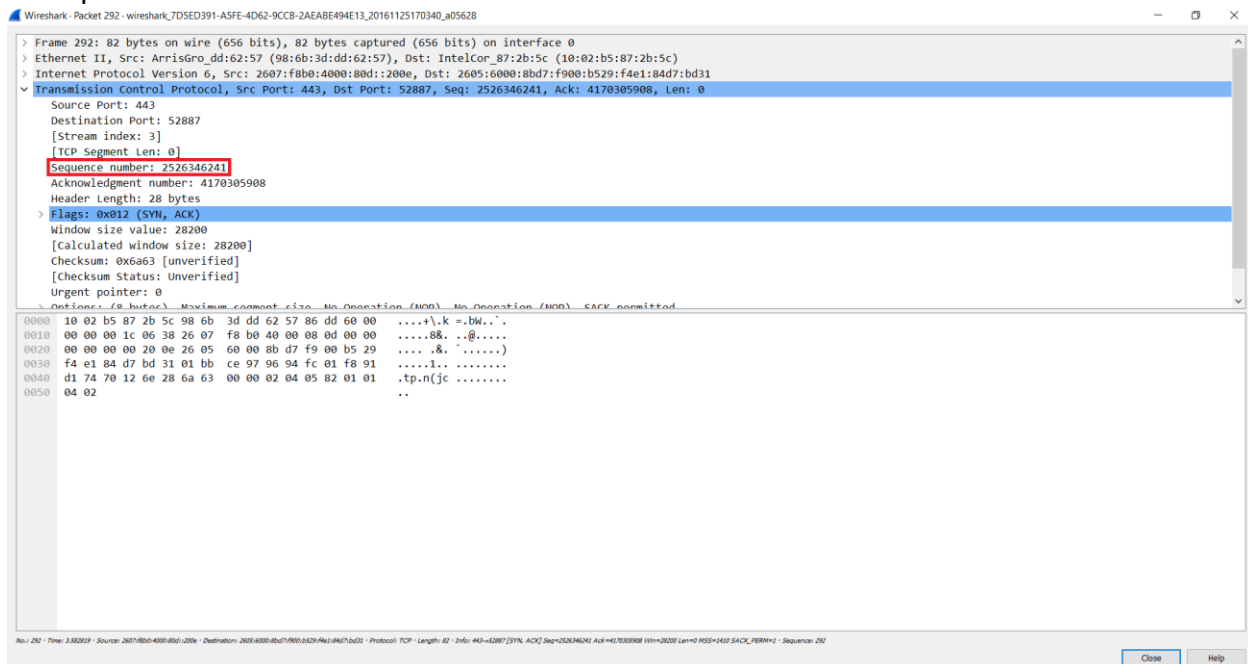
2. What is it in the segment that identifies the segment as a SYN segment?

-> The SYN bit is Set in the flags that identifies the segment as a SYN Segment.



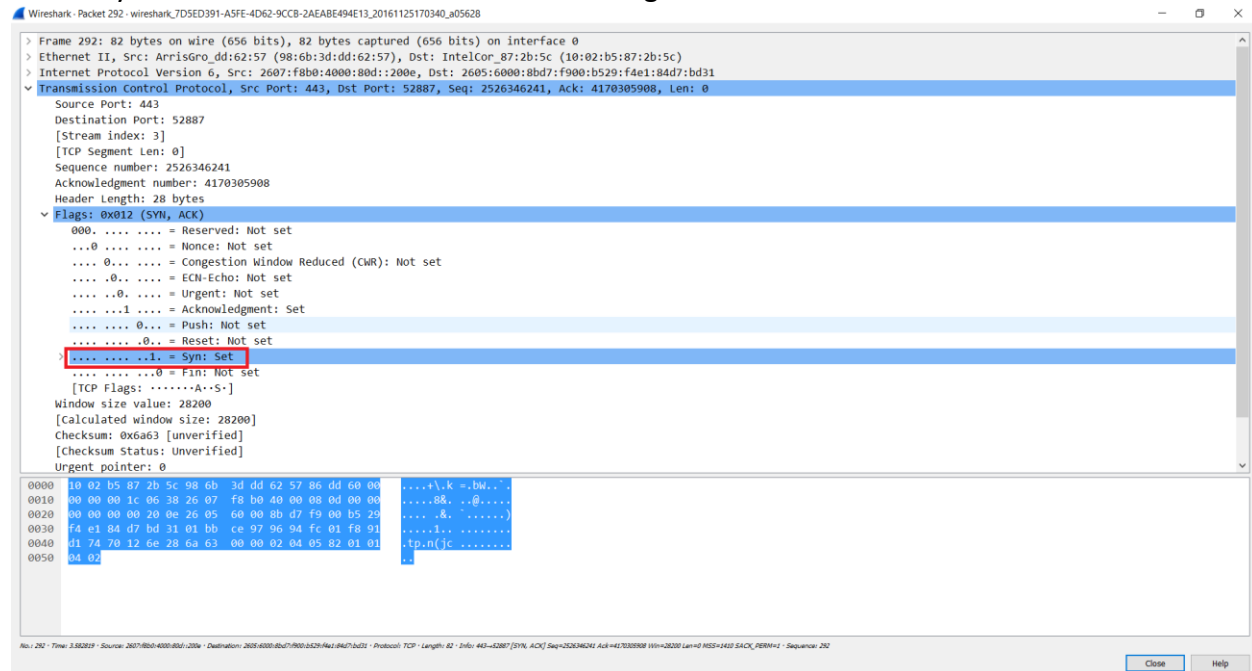
3. What is the sequence number of the SYNACK segment sent by youtube.com to the client computer in reply to the SYN? What is the value of the Acknowledgement field in the SYNACK segment?

-> Sequence number: 2526346241



4. How did youtube.com determine that value? What is it in the segment that identifies the segment as a SYNACK segment?

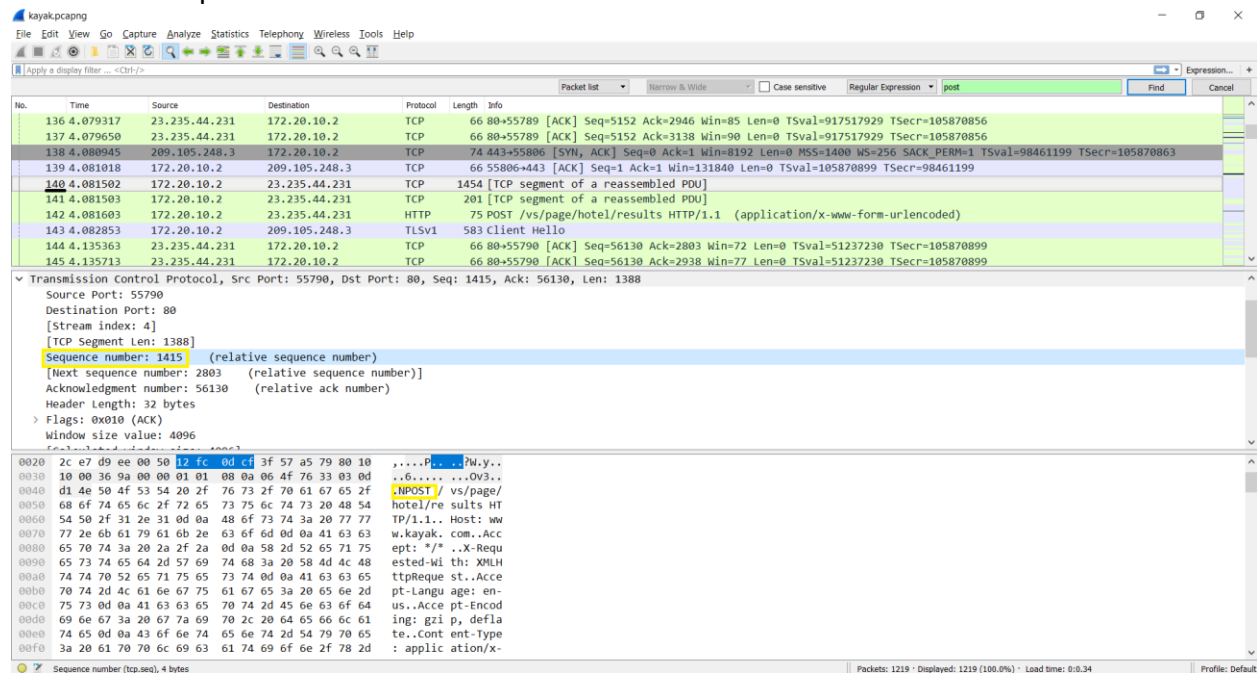
-> The Syn and the ack bits are both set in the flags.



Problem Set 3

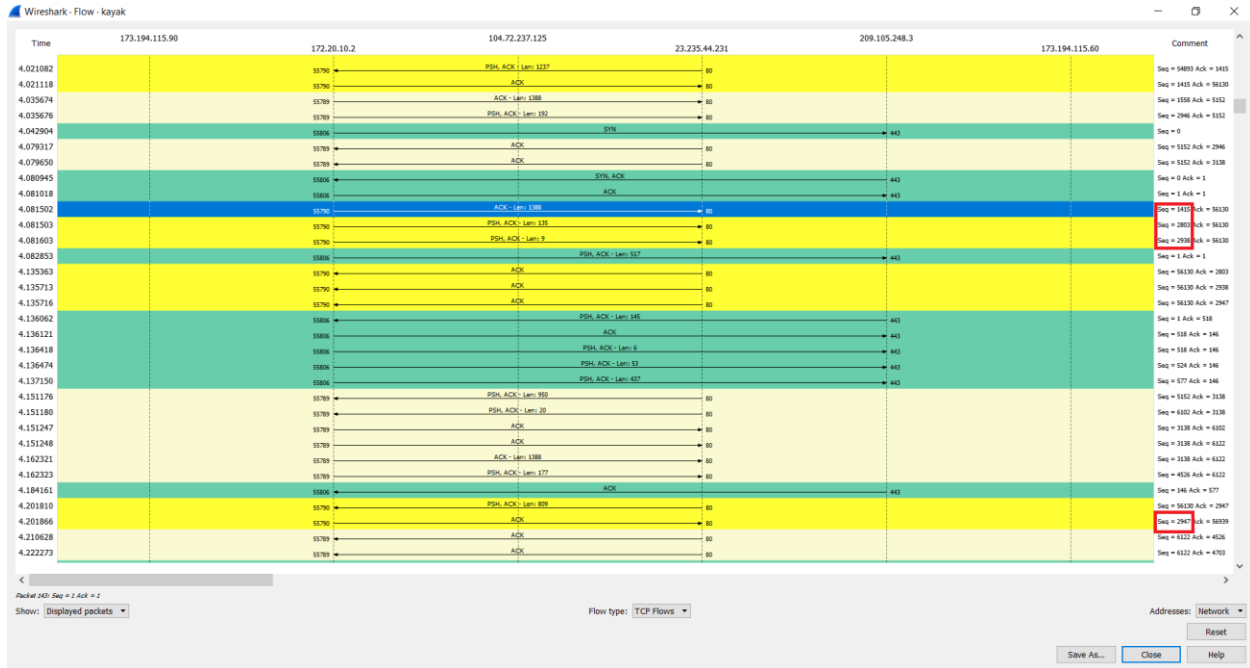
1. What is the sequence number of the TCP segment containing the first HTTP POST command?

-> The sequence number of the TCP segment containing the first HTTP POST command is on No. 140 and its sequence number is 1415

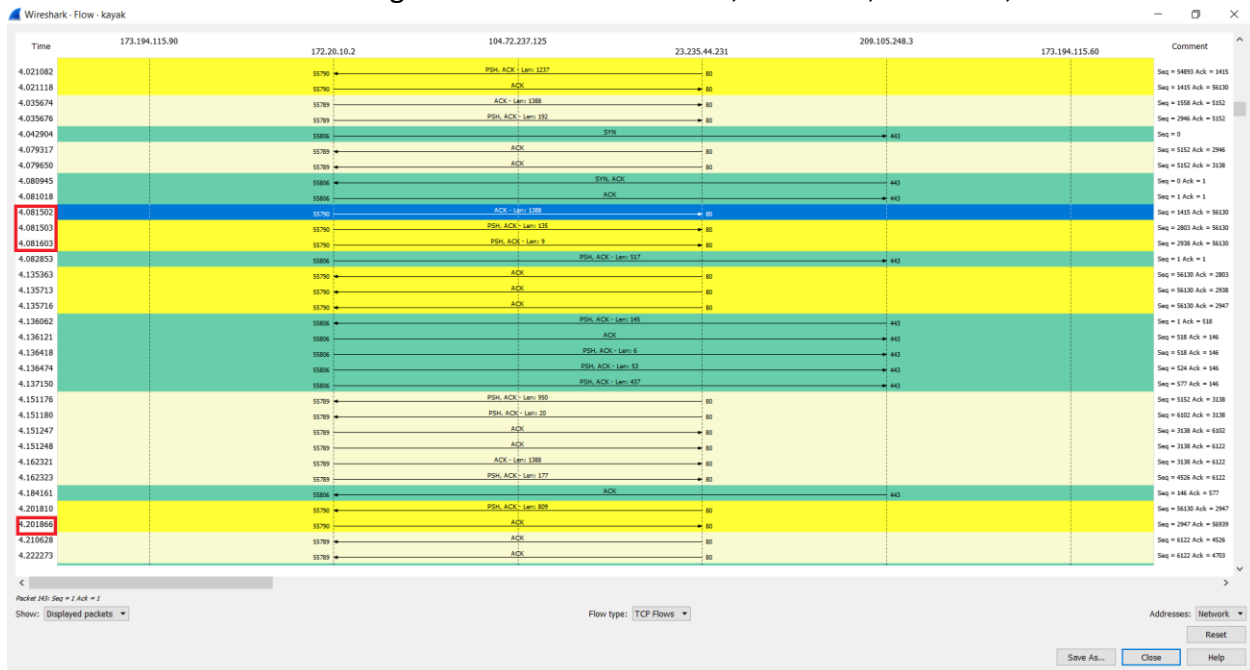


2. Consider the TCP segment containing the HTTP POST as the first segment in the TCP connection.

i) What are the sequence numbers of the first four segments in the TCP connection (including the segment containing the HTTP POST)? Sequence numbers: 1415, 2803, 2938, 2947

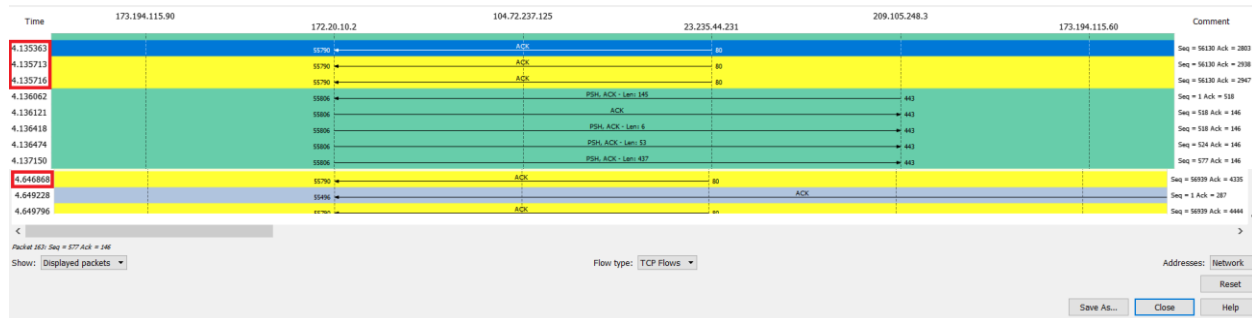


ii) At what time was each segment sent?
Time at which each segment was sent: 4.081502, 4.081503, 4.081603, 4.201866



iii) When was the ACK for each segment received?

Time at which ACK for each segment was received: 4.135363, 4.135713, 4.135716, 4.646468



iv) Given the difference between when each TCP segment was sent, and when its acknowledgement was received,

Difference: 0.053861, 0.05421, 0.054113, 0.445002

v) what is the RTT value for each of the four segments?

RTT value: 0.053861000, 0.054210000, 0.054113000, 0.076515000

vi) What is the EstimatedRTT value (see Section 3.5.3, page 239 in text) after the receipt of each ACK?

Estimated RTT: 0.053861, 0.05421, 0.054113, 0.398941

vii) 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.

3. What is the length of each of the first four TCP segments?

Length: 140: - 1454, 141: - 201, 142: - 75, 160: - 66

4. What is the minimum amount of available buffer space advertised at the receiver for the entire trace?

Minimum available buffer space at the receiver: 28960 bytes

Wireshark packet capture showing a TCP connection. The packet list shows a SYN packet (No. 21) and an ACK packet (No. 34) with a window size of 1460. The packet details for packet 34 show the 'Window' field as 1460. The packet bytes show the TCP header and data.

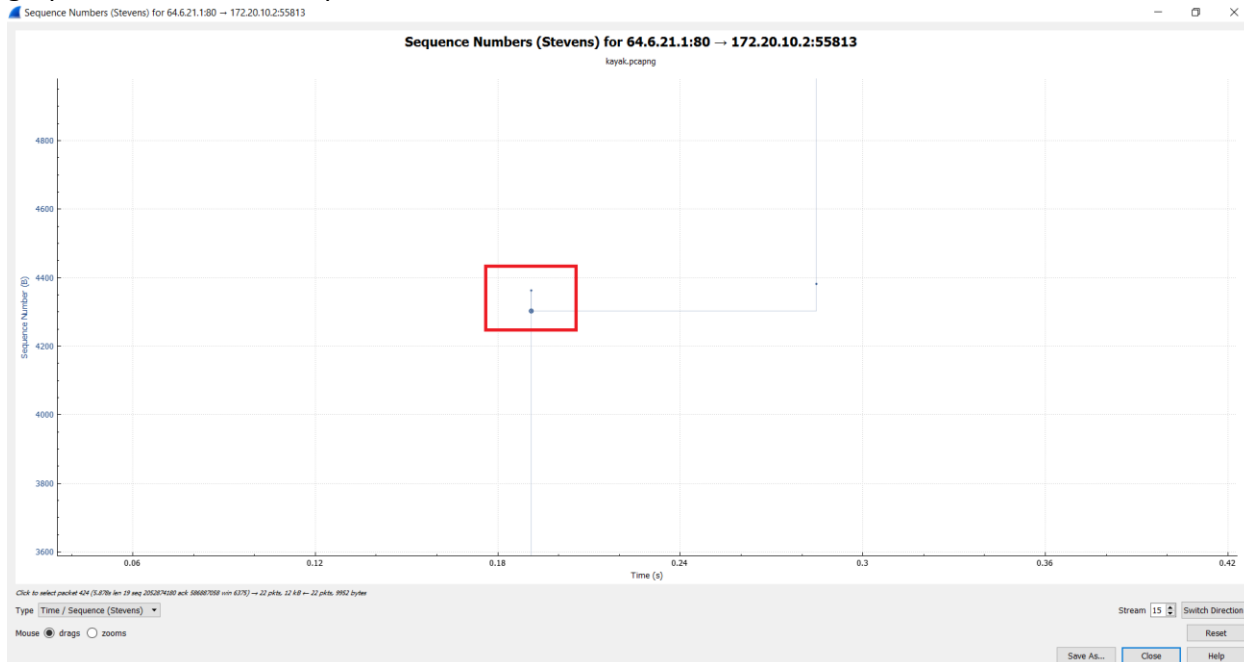
5. Does the lack of receiver buffer space ever throttle the sender?

Since the maximum buffer space goes to 131040, the lack of receiver buffer space does not throttle the sender.

Wireshark packet capture showing a TCP connection. The packet list shows a SYN packet (No. 443) and an ACK packet (No. 512) with a window size of 131040. The packet details for packet 512 show the 'Window' field as 131040. The packet bytes show the TCP header and data.

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

The retransmitted segments in the trace file are. I checked for the Time Sequence(Stevens) graph in TCP Stream Graphs.



7. How much data does the receiver typically acknowledge in an ACK? Can you identify cases where the receiver is ACK-ing every other received segment (see Table 3.2 on page 247 in the text)?

-> The receiver typically acknowledges 65535 bytes.

Following are the numbers, where the case asked took place = 841, 845, 846, 848

kayakpcapng

File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help

Apply a display filter: <Ctrl>F

No.	Time	Source	Destination	Protocol	Length	Info
833	9.131288	64.6.21.1	172.20.10.2	TCP	1454	[TCP segment of a reassembled PDU]
834	9.131380	172.20.10.2	64.6.21.1	TCP	66	55826->443 [ACK] Seq=98620814 Ack=1590579844 Win=65535 Len=0 TSval=10587582
835	9.131510	64.6.21.1	172.20.10.2	TLSv1..	271	Application Data
836	9.131562	172.20.10.2	64.6.21.1	TCP	66	55826->443 [ACK] Seq=98620814 Ack=1590580049 Win=65535 Len=0 TSval=10587582
837	9.131049	64.6.21.1	172.20.10.2	TLSv1..	1287	Application Data
838	9.132081	172.20.10.2	64.6.21.1	TCP	66	55826->443 [ACK] Seq=98620814 Ack=1590581270 Win=65535 Len=0 TSval=10587582
839	9.219836	129.107.56.31	172.20.10.2	TCP	1434	[TCP segment of a reassembled PDU]
840	9.219842	129.107.56.31	172.20.10.2	TCP	158	[TCP segment of a reassembled PDU]
841	9.219934	129.107.56.31	129.107.56.31	TCP	66	55825->80 [ACK] Seq=3630394512 Ack=3550338103 Win=65535 Len=0 TSval=1058759
842	9.220038	129.107.56.31	172.20.10.2	TCP	1434	[TCP segment of a reassembled PDU]
843	9.220391	129.107.56.31	172.20.10.2	TCP	158	[TCP segment of a reassembled PDU]
844	9.220394	129.107.56.31	172.20.10.2	TCP	1434	[TCP segment of a reassembled PDU]
845	9.220441	129.107.56.31	129.107.56.31	TCP	66	55825->80 [ACK] Seq=3630394512 Ack=3550339563 Win=65535 Len=0 TSval=1058759
846	9.220492	129.107.56.31	129.107.56.31	TCP	66	55825->80 [ACK] Seq=3630394512 Ack=3550340931 Win=65535 Len=0 TSval=1058759
847	9.221162	129.107.56.31	172.20.10.2	TCP	158	[TCP segment of a reassembled PDU]
848	9.221242	129.107.56.31	129.107.56.31	TCP	66	55825->80 [ACK] Seq=3630394512 Ack=3550341023 Win=65535 Len=0 TSval=1058759
849	9.222009	129.107.56.31	172.20.10.2	TCP	67	[TCP segment of a reassembled PDU]
850	9.222067	129.107.56.31	129.107.56.31	TCP	66	55825->80 [ACK] Seq=3630394512 Ack=3550341024 Win=65535 Len=0 TSval=1058759
851	9.265345	129.107.56.31	172.20.10.2	TCP	1434	[TCP segment of a reassembled PDU]
852	9.265349	129.107.56.31	172.20.10.2	TCP	157	[TCP segment of a reassembled PDU]
853	9.265422	129.107.56.31	129.107.56.31	TCP	66	55825->80 [ACK] Seq=3630394512 Ack=3550342483 Win=65535 Len=0 TSval=1058759
854	9.266362	129.107.56.31	172.20.10.2	TCP	1434	[TCP segment of a reassembled PDU]
855	9.266365	129.107.56.31	172.20.10.2	TCP	1434	[TCP segment of a reassembled PDU]
856	9.266367	129.107.56.31	172.20.10.2	TCP	250	[TCP segment of a reassembled PDU]

> Frame 841: 66 bytes on wire (528 bits), 66 bytes captured (528 bits) on interface 0
 > Ethernet II, Src: Apple 8b:6e:80 (6c:40:08:8b:6e:80), Dst: fa:cf:9c:21:5f:64 (fa:cf:9c:21:5f:64)
 > Internet Protocol Version 4, Src: 172.20.10.2, Dst: 129.107.56.31
 > Transmission Control Protocol, Src Port: 55825, Dst Port: 80, Seq: 3630394512, Ack: 3550338103, Len: 0

```

0000  fa cf 9c 21 5f 64 6c 40 08 8b 6e 80 08 00 45 00  ...l d!@ ...n...E.
0010  00 34 6e d5 40 00 00 06 5c 4e ac 14 0a 02 81 6b  .4n.@.@. \N....k
0020  38 1f da 11 00 50 d8 63 6c 90 d3 9d dc 37 80 10  8....P.c 1....7..
0030  ff ff ad b1 00 00 01 01 08 0a 06 4f 89 ca de 7a  .......O...Z
0040  1b ab
  
```

kayak

Packets: 1219 · Displayed: 1219 (100.0%) · Load time: 0:0.24

Profile: Default

8. What is the throughput (bytes transferred per unit time) for the TCP connection (Just consider a single connection)? Think on how to calculate the throughput!

Wireshark - Conversations - kayak

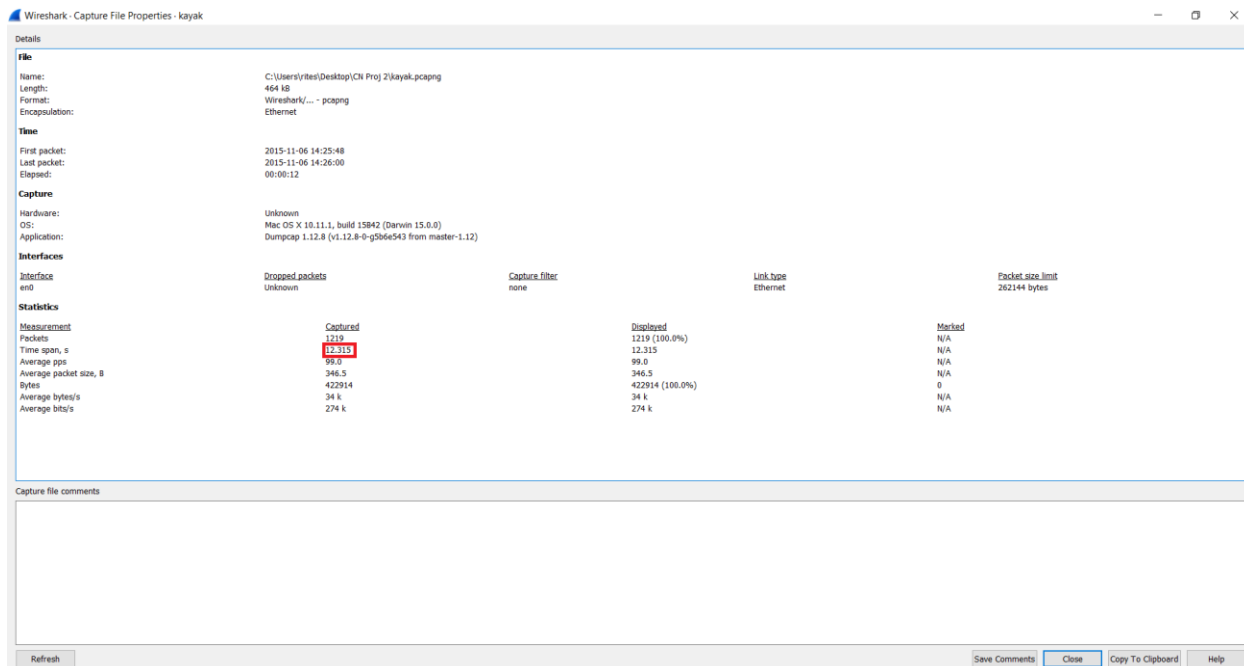
Ethernet 1IPv4 29IPv6TCP 61UDP 25

Address A	Address B	Packets	Bytes	Packets A → B	Bytes A → B	Packets B → A	Bytes B → A	Rel Start	Duration	Bits/s A → B	Bits/s B → A
6c40088b6e80	fa:cf:9c:21:5f:64	1,219	422 k	637	163 k	582	259 k	0.000000	12.3150	106 k	168 k

☐ Name resolution☐ Limit to display filter☐ Absolute start time

CopyFollow Stream...Graph...CloseHelp

Conversation Types ▼



Throughput = $163/12.315 = 13.2358$ Kbytes/sec

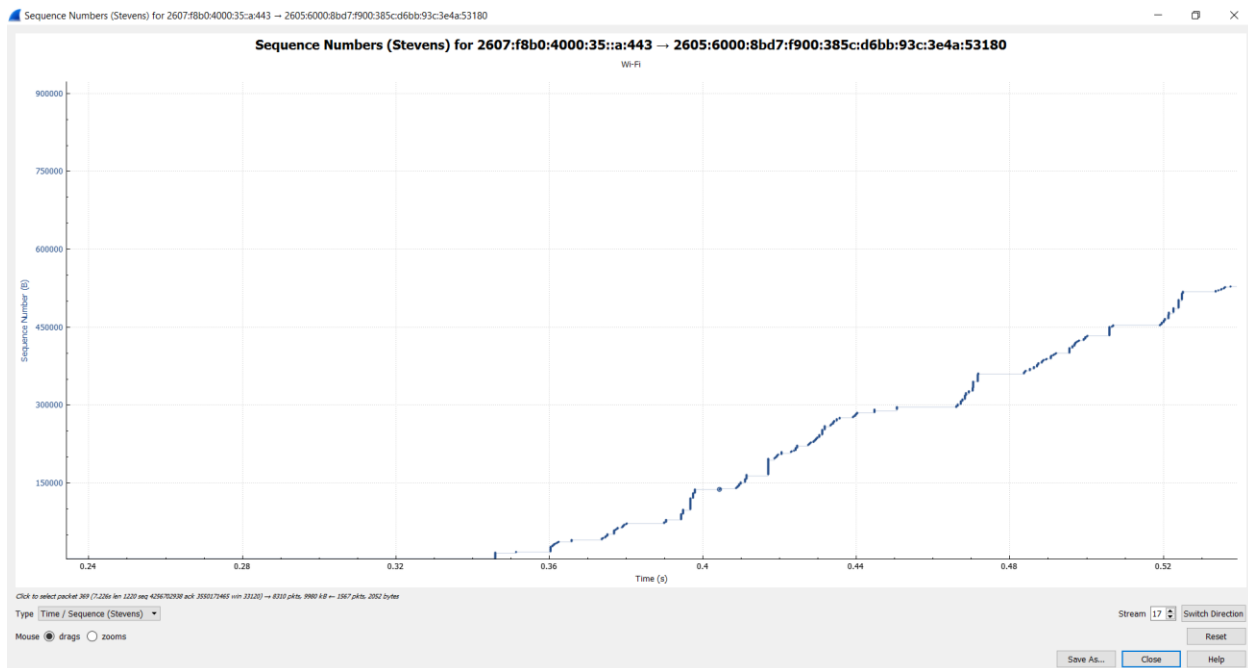
9. Explain how you calculated this value.

-> The throughput was calculated using the amount of data sent from sender to receiver and the time span provided. Dividing the two will give you the value.

Problem Set 4:

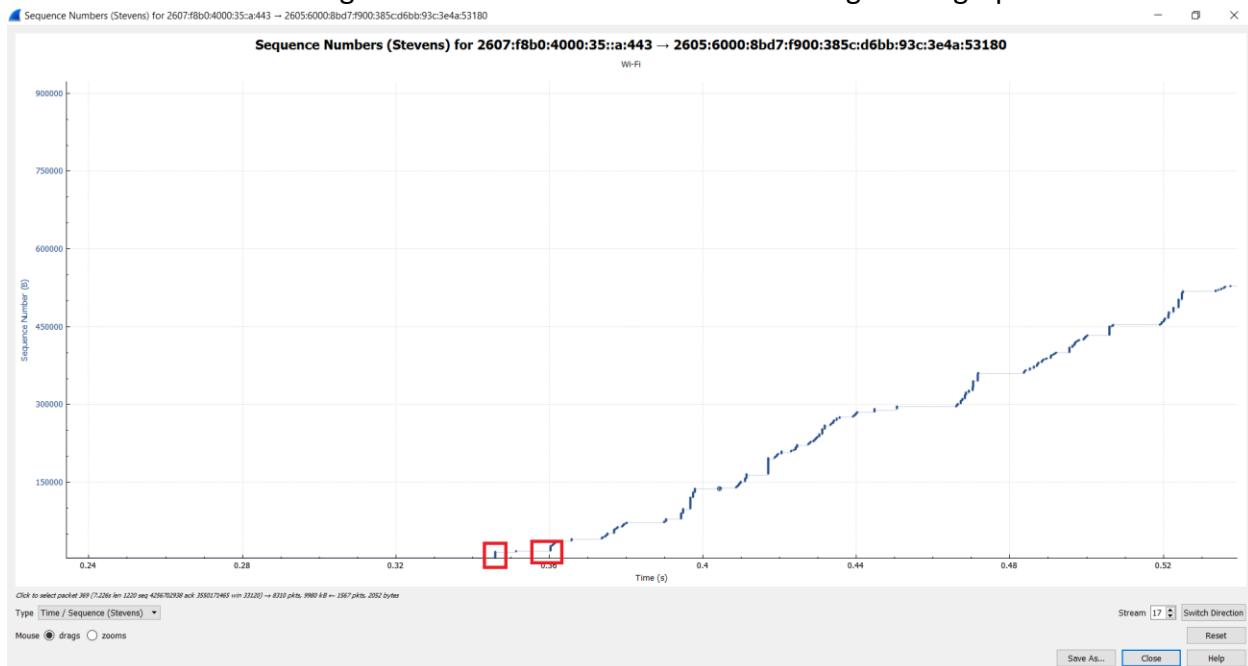
Use the Time-Sequence-Graph(Stevens) plotting tool to view the sequence number versus time plot of segments being sent from youtube.com to your computer.

Answer each of three questions below for the trace that you have gathered when you transferred a file to your computer from youtube.com.



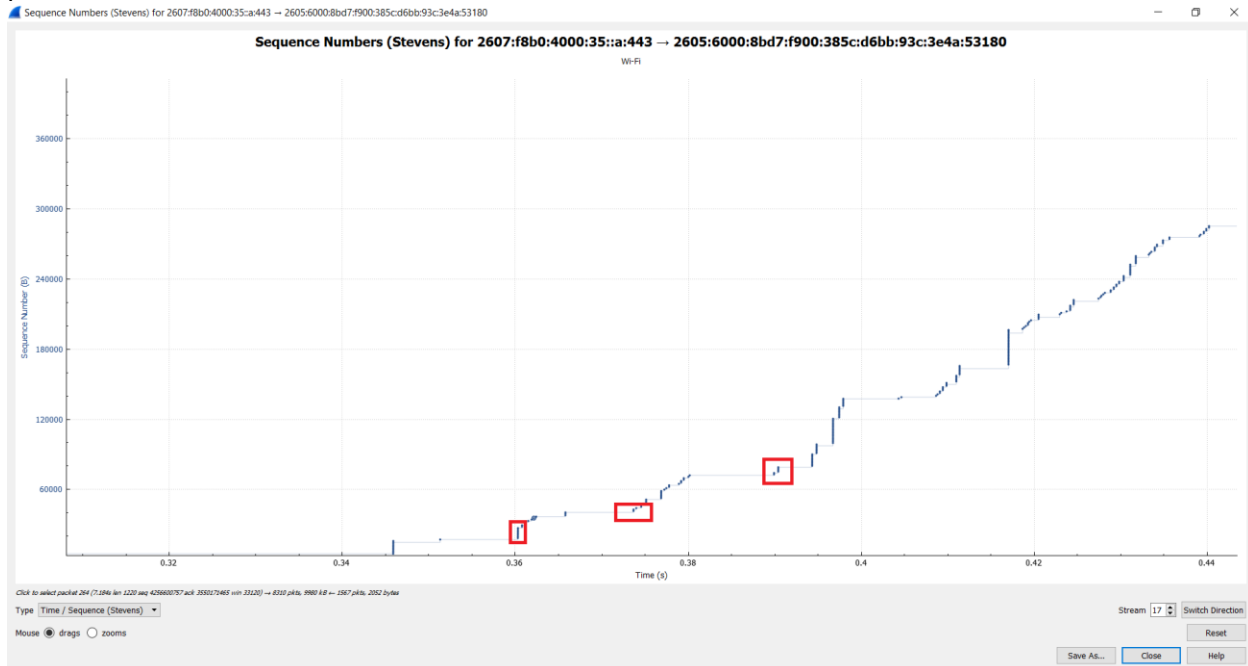
1. Can you identify where TCP's slow-start phase begins and ends.

-> The TCP's slow start begins at 0.345 and ends at 0.36 according to the graph.



2. Where congestion avoidance takes over? Highlight these areas.

-> Congestion avoidance takes place at places around 0.36, 0.375, 0.39 and after each time packets are sent. Not all places are marked on the graph where the congestion avoidance takes place.



3. Comment on ways in which the measured data differs from the idealized behavior of TCP that we've studied in the text.

-> After viewing the measured data, we can say that the scenarios in the real world are very different than what is studied. In the real world, there is packet loss every now and then due to the window size difference, so it is clearly established that the level of efficiency always differs.