**Computer Networks CSE 5344 Project 2**

**Transmission Control Protocol Analysis using Wireshark**

**Name: Sushmitha Nagarajan**

**Student ID: 1001556348**

**Objective:**

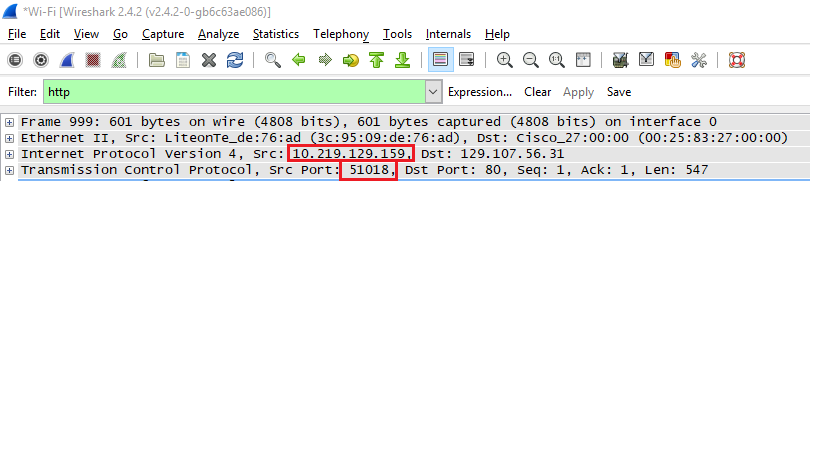
**Problem Set 1:**

1. What is the IP address and TCP port number used by your client computer (source) to browse the page uta.edu? Use the 'GET' message to answer the following questions.

Ans:

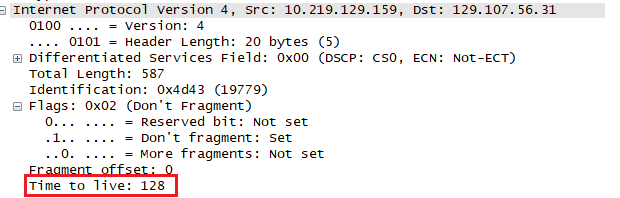
IP address used by the client: 10.219.129.159

TCP Port number used by the client: 51018



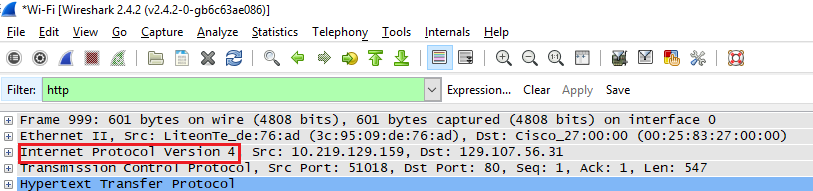
1. What is the TTL value that is used in this communication?

Ans. TTL: 128 secs



1. Did you use IPV4 or IPV6 for communication?

Ans. The client uses IPV4 for communication.

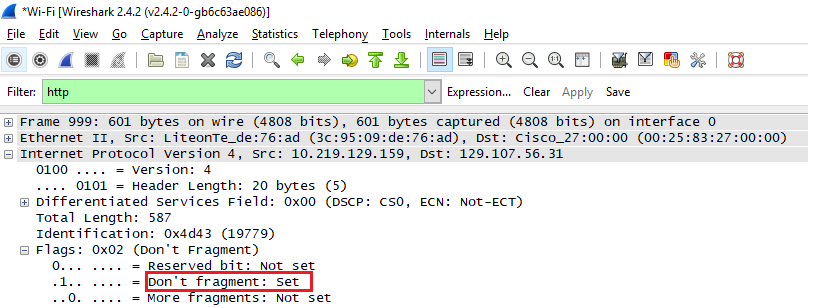


4 Do your optional field has some information or not?

Ans: The optional field is not available in HTTP segment.

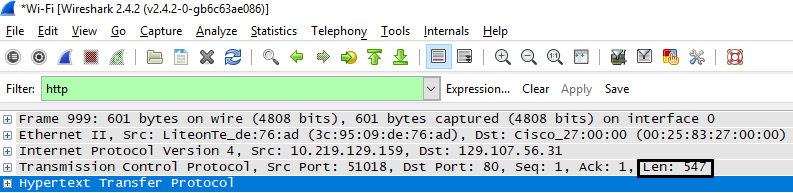
1. Is the Packet Fragmented

Ans. Packet is not fragmented as the ‘Don’t Fragment’ packet is Set.



1. What is the TCP segment length?

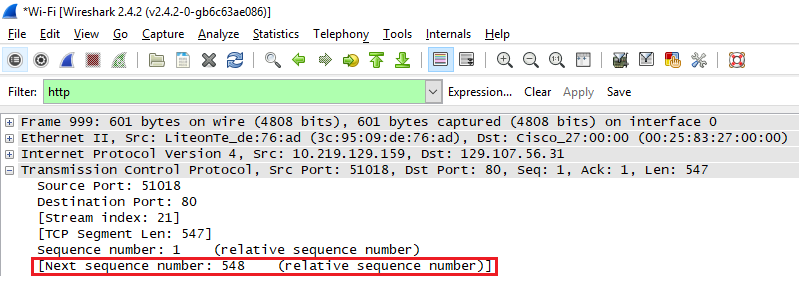
Ans. The length of the TCP segment is 547 bytes.



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

Ans. Sequence Number of TCP segment: 1

Next sequence number: 548

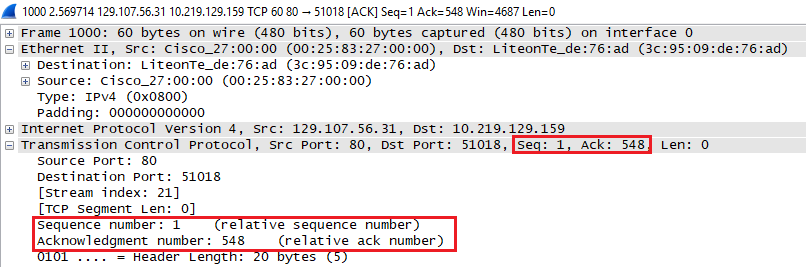


1. Calculate the acknowledgement number based on the two questions above. Verify your solution with the Wireshark values.

Ans. From the previous question, we can derive that: Initial Sequence number = 1, ACK = 1,

Acknowledgment number = (ACK number of HTTP segment + TCP Segment length) = 1+547= 548.

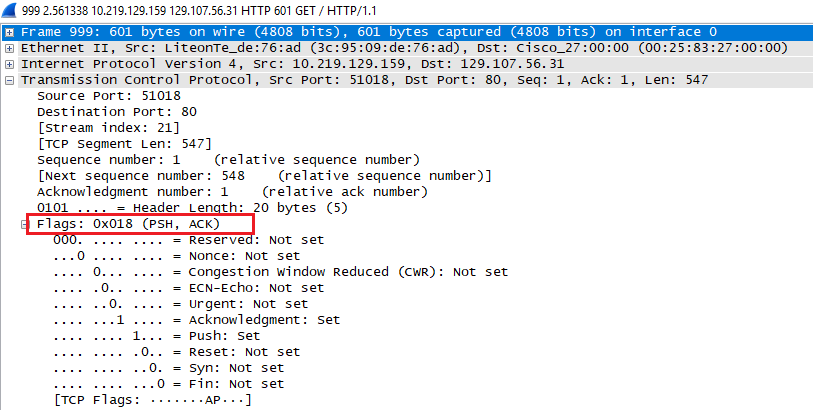
So, ACK number should be equal to Next Sequence number i.e. 548.



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

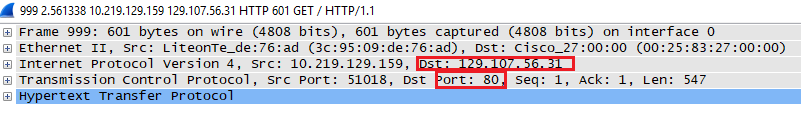
Ans. TCP Flags are: PSH and ACK bit are set

* Reserved
* Nonce
* Congestion Window Reduced
* ECN- Echo
* Urgent
* Acknowledgment
* Push
* Reset
* Syn
* Fin



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

Ans. IP address of uta.edu: 129.107.56.31, Port number used by uta.edu to send TCP segments: 80

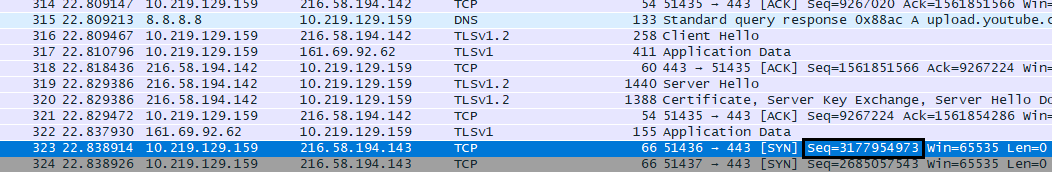


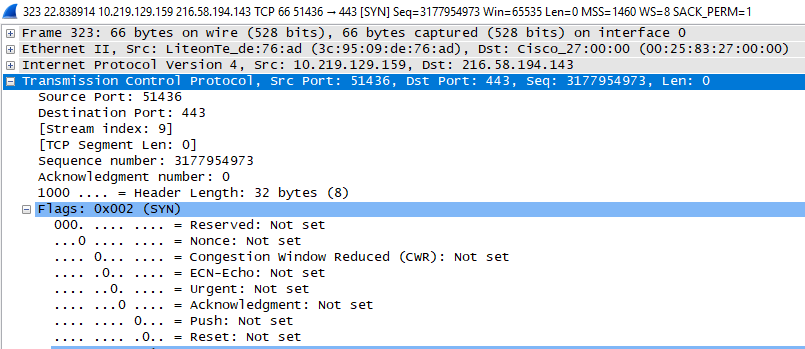
**Section 2: Analyzing the Connection Parameters in TCP**

**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?

Ans. The absolute sequence number of the of the TCP SYN segment is **3177954973**.

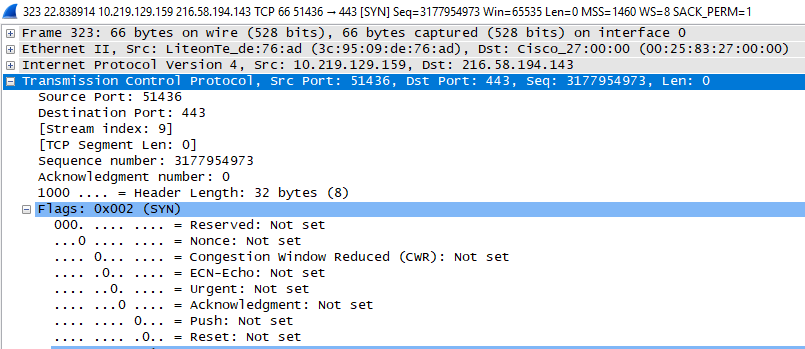




2. What is it in the segment that identifies the segment as a SYN segment?

Ans. Flag in the TCP protocol depicts that it is a SYN bit.

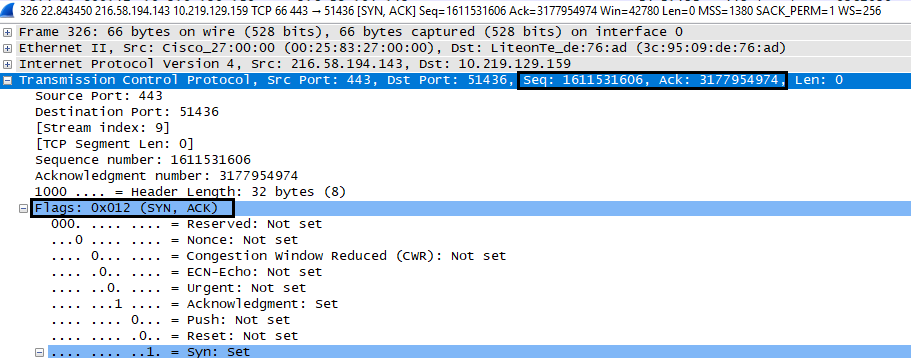
**Flags – 0x002 (SYN)**

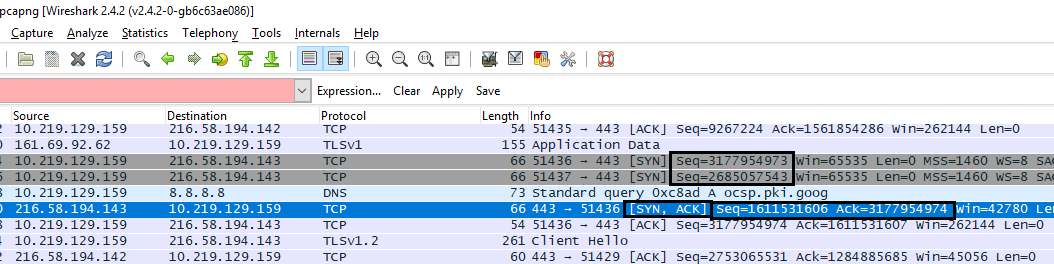


3. What is the sequence number of the SYNACK segment sent by youtube.com to the client computer in reply to the SYN? What is the value of the Acknowledgement field in the SYNACK segment?

Ans. **Sequence number** of the SYNACK segment sent by youtube.com to the client computer - 1611531606

**Acknowledgement number** in the SYNACK segment – 3177954974



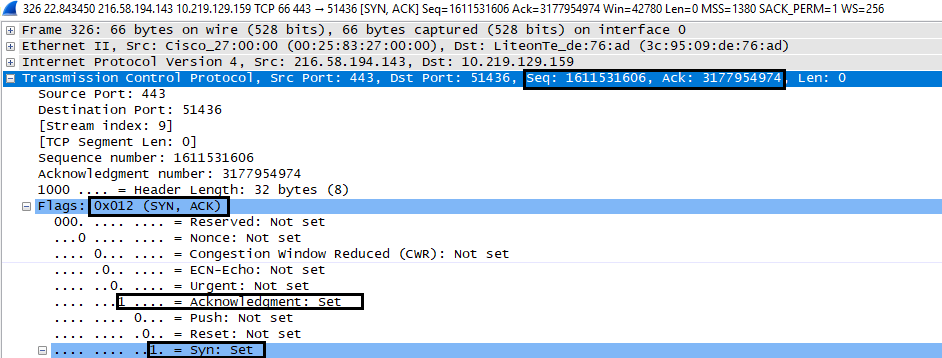


4. How did youtube.com determine that value? What is it in the segment that identifies the segment as a SYNACK segment?

Ans. Youtube.com determines the value using Flag bit.

The segment that identifies SYNACK segment in the Flag bit.

**Flag – 0x012 (SYN,ACK)**

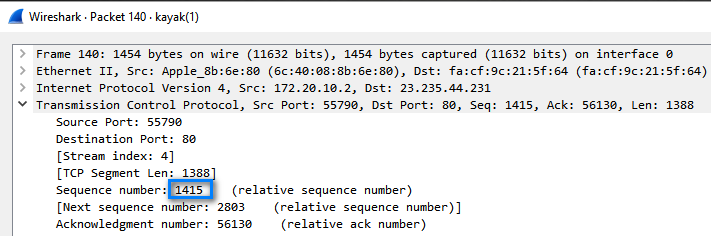
****

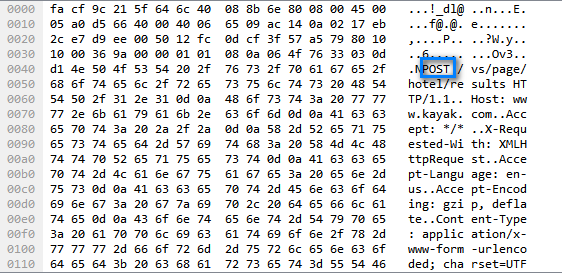
**Section 3: Analysis of the trace provided**

**Problem Set 3:**

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

Ans. Relative Sequence number for the TCP segment containing the first HTTP POST command 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)?

Ans. TCP segment having HTTP POST: (140)

TCP Segment 1: (140)

Sequence number: 1415 (Relative), Next Sequence number: (Relative)

TCP Segment 2: (141)

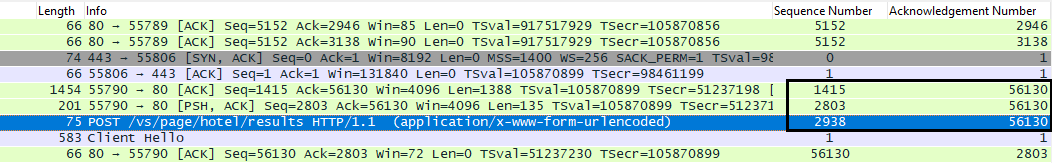
Sequence number: 2803 (Relative), Next Sequence number: (Relative)

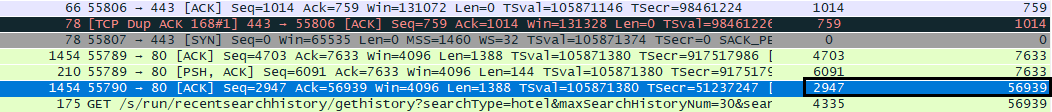
TCP Segment 3: (142)

Sequence number: 2938 (Relative), Next Sequence number: (Relative)

TCP Segment 4: (174)

Sequence number: 2947 (Relative), Next Sequence number: (Relative)





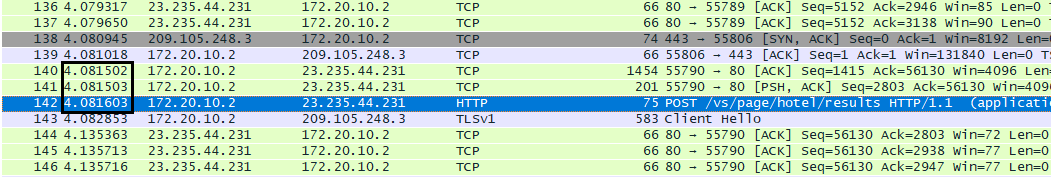
ii) At what time was each segment sent?

Ans. Time for each segment is as follows:

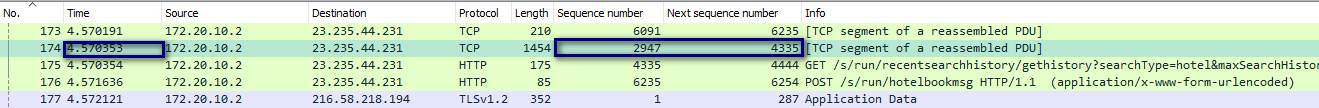
TCP segment having HTTP POST: 4.081502 secs

TCP Segment 1: 4.081503 secs

TCP Segment 2: 4.081603 secs



TCP Segment 3: 4.570353 secs

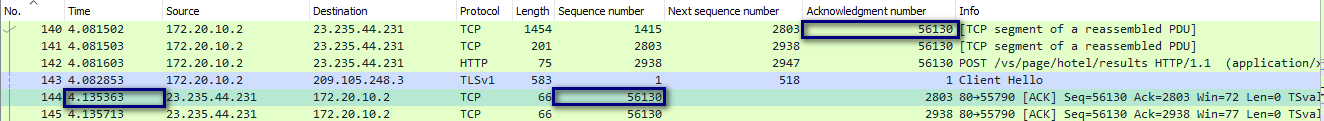


iii) When was the ACK for each segment received?

Ans.

**TCP containing HTTP Post (140):** Sequence number: 1415, ACK: 56130

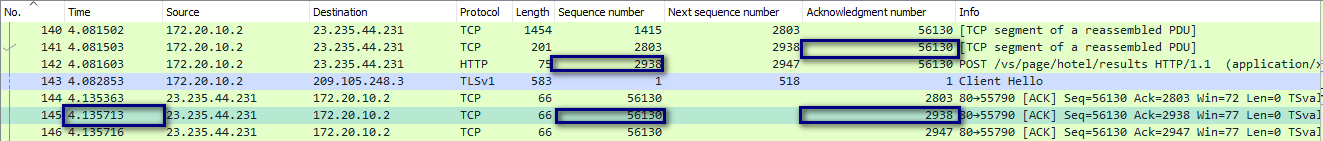
**ACK for HTTP (144)**: Sequence number: 56130 ACK: 2803, Time: 4.135363 secs



**TCP Segment 1 (141):** Sequence number: 2803, ACK: 56130

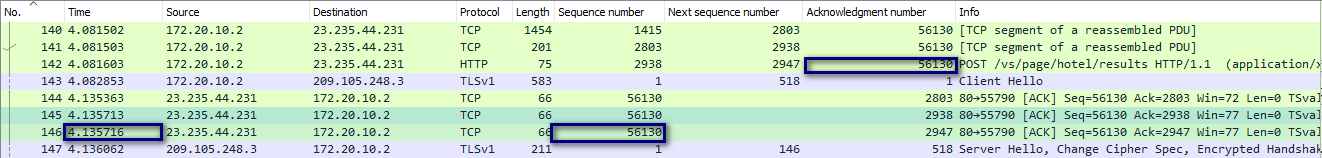
**ACK for TCP Segment 1(145):** Sequence number: 56130, ACK: 2938, Time: 4.135713 secs

We have considered this packet as the next sequence number of the ACK packet is equal to sequence number of the next TCP segment.



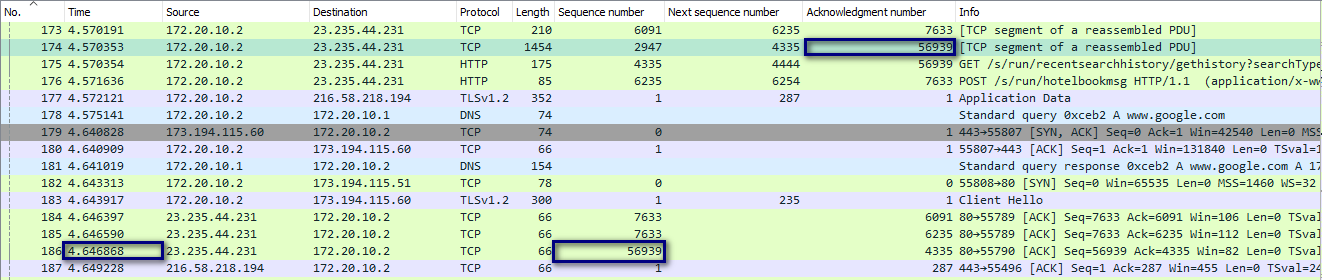
**TCP Segment 2 (142):** Sequence number: 2938, ACK: 56130

**ACK for TCP Segment 2 (146):** Sequence number: 56130, ACK: 2947, Time: 4.135716 Secs



**TCP Segment 3 (174):** Sequence number: 4335, ACK: 56939

**ACK for TCP Segment 3:** Sequence number: 56939, ACK: 4335, Time: 4.646868 Secs



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

Ans.

|  |  |  |  |
| --- | --- | --- | --- |
| Segment | Time (secs) | Time (ACK received) (secs) | Difference (Time(ACK)- Time) (secs) |
| TCP Segment with HTTP post | 4.081502 | 4.135363 | 0.053861 |
| TCP Segment 1 | 4.081503 | 4.135713 | 0.05421 |
| TCP Segment 2 | 4.081603 | 4.135716 | 0.054113 |
| TCP Segment 3 | 4.570353 | 4.646868 | 0.076518 |

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

Ans.

|  |  |  |  |
| --- | --- | --- | --- |
| Segment | Time (secs) | Time (ACK received) (secs) | Sample RTT (Time(ACK)- Time) (secs) |
| TCP Segment with HTTP post | 4.081502 | 4.135716 | 0.053861 |
| TCP Segment 1 | 4.081503 | 4.135713 | 0.05421 |
| TCP Segment 2 | 4.081603 | 4.135716 | 0.054113 |
| TCP Segment 3 | 4.570373 | 4.646868 | 0.076518 |

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

Ans. Estimated RTT of first segment is equal to sample RTT

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Segment | Time (secs) | Time (ACK received) (secs) | Sample RTT (secs) | Estimated RTT (secs) |
| TCP Segment with HTTP post | 4.081502 | 4.135716 | 0.53861 | 0.053861 |
| TCP Segment 1 | 4.081503 | 4.135713 | 0.05421 | 0.053904 |
| TCP Segment 2 | 4.081603 | 4.135716 | 0.054113 | 0.053930 |
| TCP Segment 3 | 4.570373 | 4.646868 | 0.076518 | 0.056754 |

vii) Assume that the value of the EstimatedRTT is equal to the measured RTT for the first segment, and then is computed using the Estimated RTT equation on page 239 for all subsequent segments.

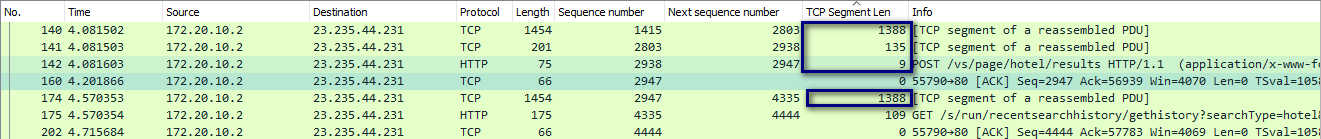
Ans.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Segment | Time (secs) | Time (ACK received) (secs) | Sample RTT (secs) | Estimated RTT (secs) |
| TCP Segment with HTTP post | 4.081502 | 4.135716 | 0.53861 | 0.053861 |
| TCP Segment 1 | 4.081503 | 4.135713 | 0.05421 | 0.053904 |
| TCP Segment 2 | 4.081603 | 4.135716 | 0.054113 | 0.053930 |
| TCP Segment 3 | 4.570373 | 4.646868 | 0.076518 | 0.056754 |

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

Ans.

|  |  |  |  |
| --- | --- | --- | --- |
| Segment | Length (bytes) | Column number | Sequence number |
| TCP Segment with HTTP post | 1388 | 140 | 1415 |
| TCP Segment 1 | 135 | 141 | 2803 |
| TCP Segment 2 | 9 | 142 | 2938 |
| TCP Segment 3 | 1388 | 174 | 4335 |

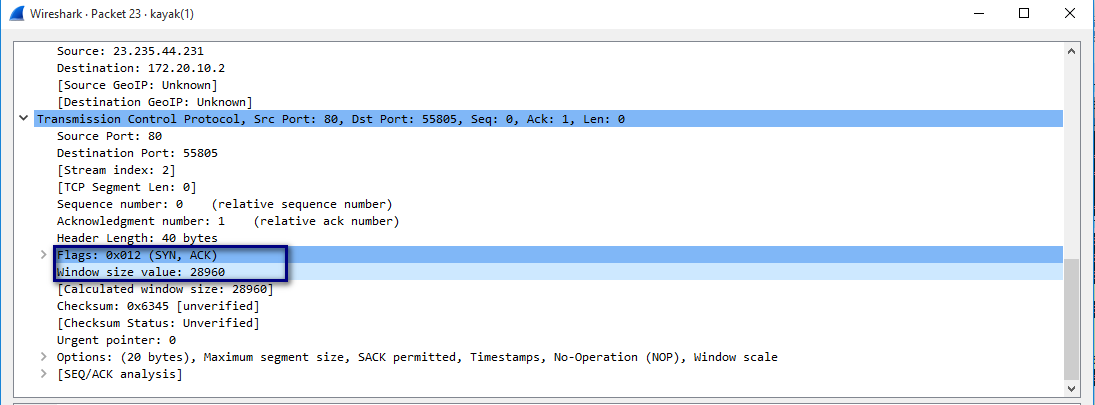


4. What is the minimum amount of available buffer space advertised at the receiver for the entire trace?

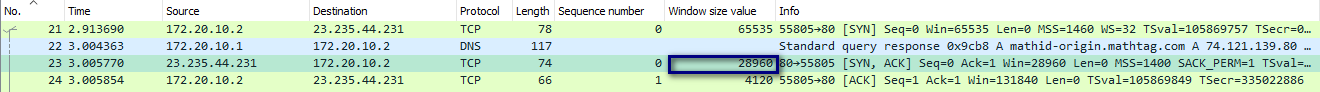
Ans. The first TCP [SYN, ACK] bit in the TCP segment gives the minimum amount of available buffer size.

Window Size: 28960

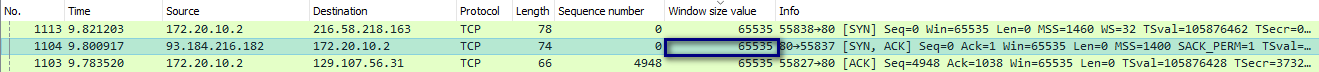
Packet number: 23



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

Ans. The minimum amount of available buffer space advertised at the receiver for the entire trace is 28960 bytes (Packet number: 23).

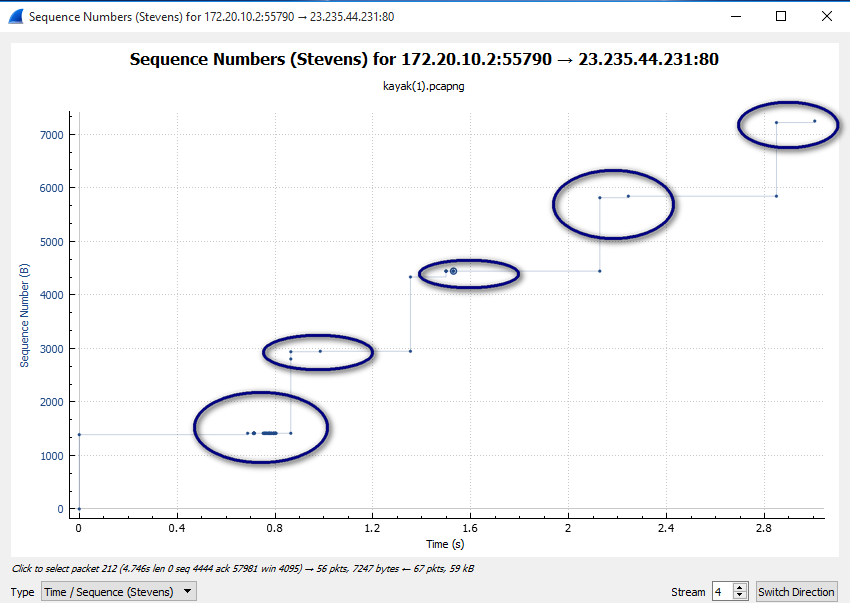
The maximum amount of available buffer space advertised at the receiver for the entire trace is 65535 bytes (Packet number: 1104).



This receiver window grows until it reaches the maximum receiver buffer size. By examining the trace, the sender is never throttled due to lack of receiver buffer space. Even when the advertised receiver window is having its lowest value i.e. 28960, the sender is constrained by congestion window. Window size will increase up to 65535 bits, allowing sufficient buffer size for the sender to send the data.

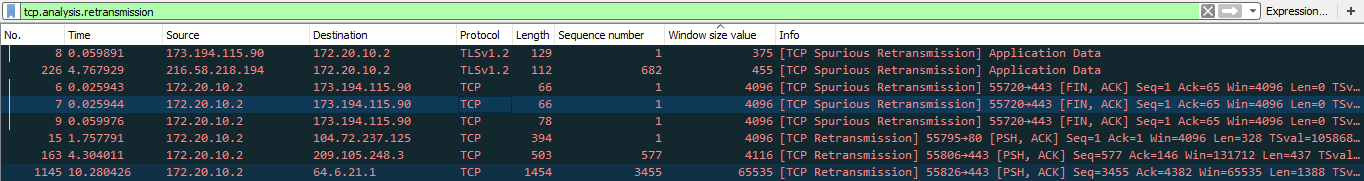
6. Are there any retransmitted segments in the trace file? What did you check for (in the trace) to answer this question?

Ans.



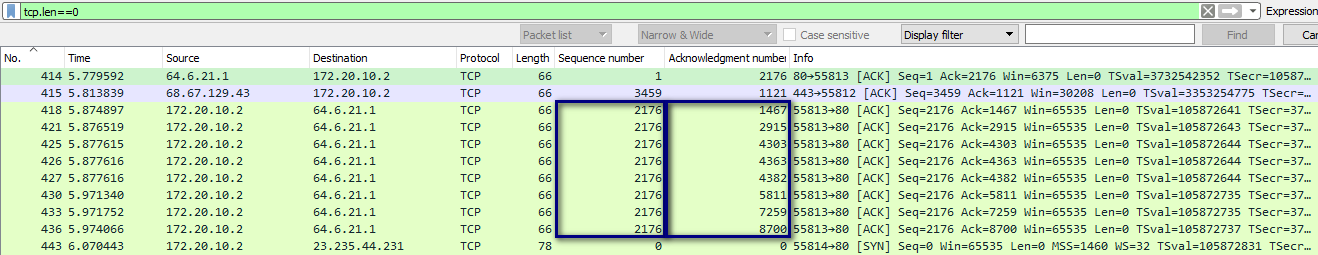
We have considered Stevens graph to check the retransmissions. In the above, graph it clearly represents the multiple re-transmissions of the same segment.

We have also used “tcp.analysis.retransmission” filter to find out the number of retransmissions.



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)?

Ans. Typically, Acknowledge data size is as follows;

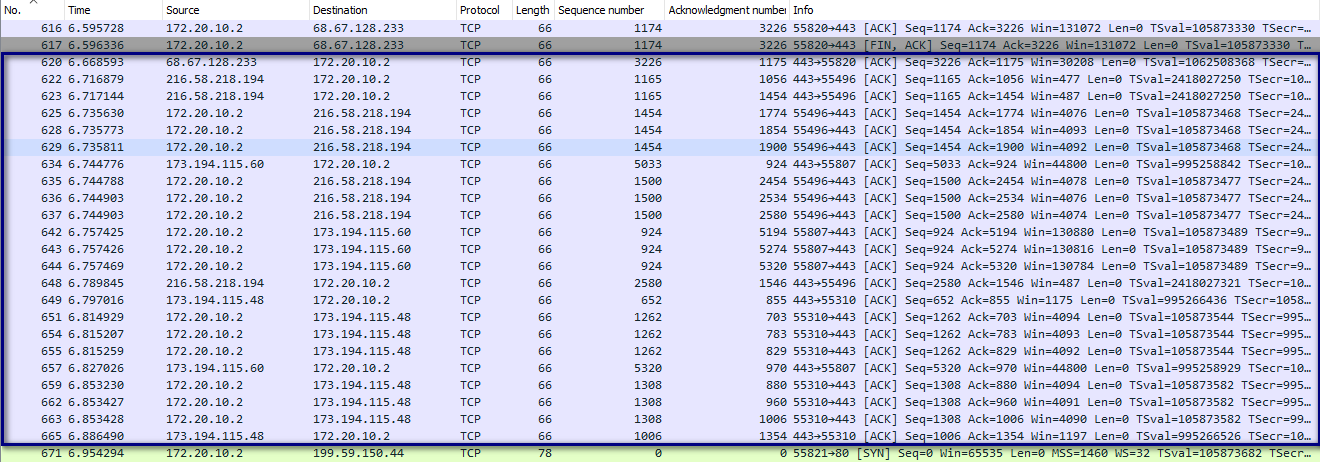


|  |  |  |  |
| --- | --- | --- | --- |
| Packet number | Sequence number | Acknowledgment number | Acknowledgement data size (Next ACK number-ACK) |
| 418 | 2176 | 1467 | 1467 |
| 421 | 2176 | 2915 | 1488 |
| 425 | 2176 | 4303 | 1388 |
| 426 | 2176 | 4363 | 60 |
| 427 | 2176 | 4382 | 19 |
| 430 | 2176 | 5811 | 1429 |
| 433 | 2176 | 7259 | 1488 |
| 436 | 2176 | 8700 | 1441 |

Approximately, 1097 is the Acknowledgment data size.

While examining the packets, we find that multiple acknowledgements has the same sequence number.

The following sample of data ACKs every other received segment:



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!

Ans.

We consider TCP trace, first sequence number to be packet number 29, and the last packet to be packet number 464.

So, the Throughput = Size of the data/Time

Size of the data = Last ACK Sequence number – First Sequence number

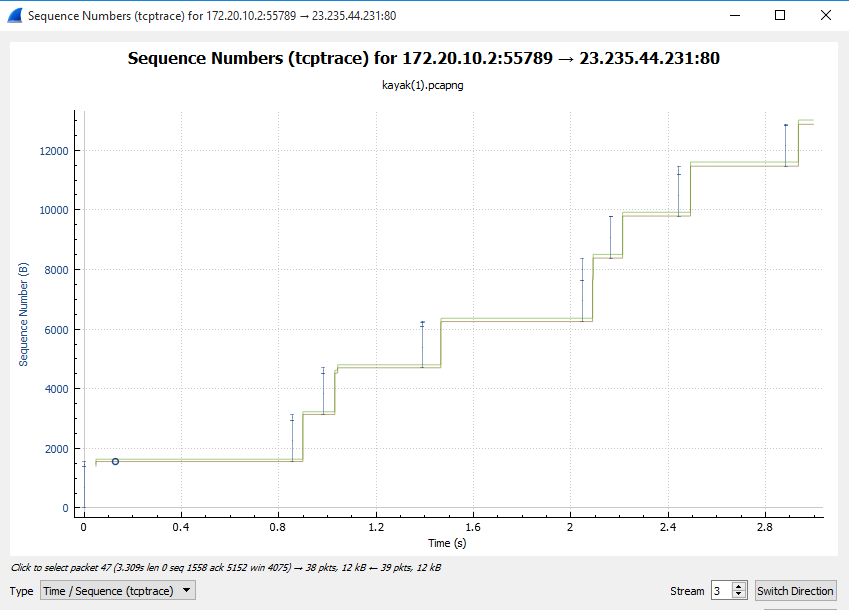
= 12870-1 = 12869

Time = 6.179502 – 3.180418

= 2.999

Throughput= 12869/2.999

= 4291 bytes/sec



9. Explain how you calculated this value.

Ans. We chose the first TCP packet with the Sequence and ACK number as 1.

By using the TCP trace report from **Statistics -> TCP Trace graphs -> TCP Trace** (as per the above image), we found that last ACK packet is 464 with the sequence number 12870.

In order to check the Throughput, we need the size of the data and the time taken.

**Size of the data =** Last ACK Sequence number – First Sequence number = 12870-1 = 12869

**Total Time Taken =** 6.179502 – 3.180418 = 2.999

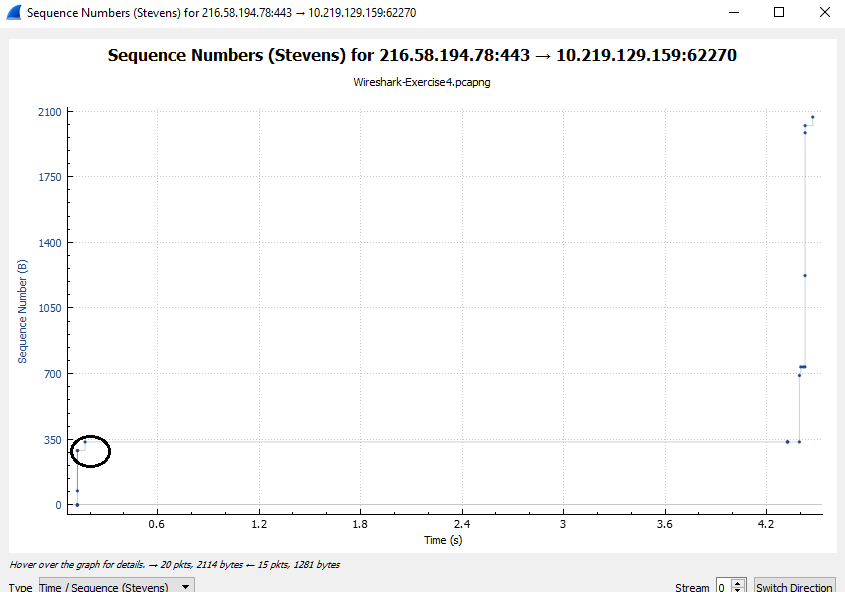
**Throughput=** 12869/2.999 = 4291 bytes/sec

**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 le to your computer from youtube.com.

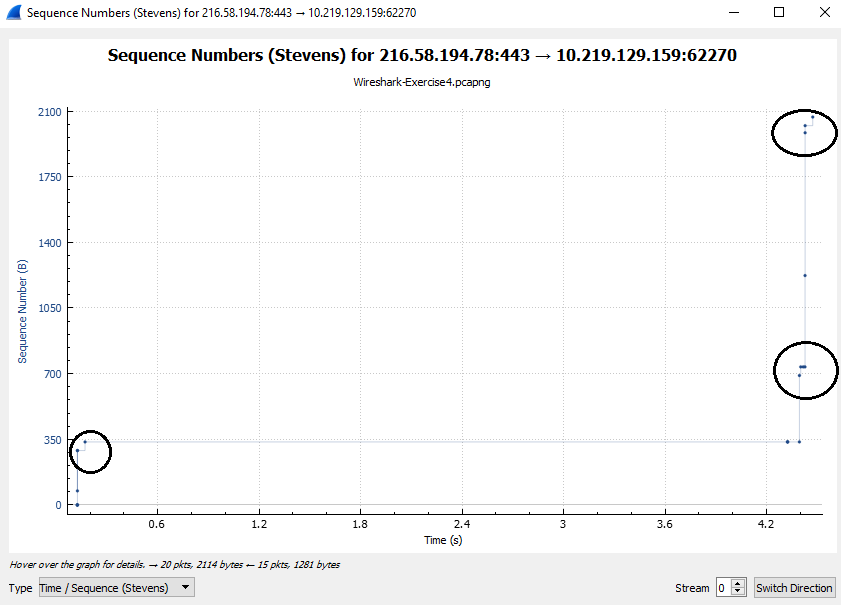
1. Can you identify where TCP's slow-start phase begins and ends.

Ans. TCP slow start phase begin from 0.1 secs



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

Ans. Congestion takes over at 0.1,4.45 secs and 4.5 secs



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

Ans. As per the ideal behavior of TCP, the sender is travelling in the same medium. Whereas, the data is travelled in multiple media. There can be more loss of data and retransmissions. And, practically TCP segments travel faster and aggressive in sending data.