Computer Networks Lab 5

Shane Cincotta

April 17, 2020

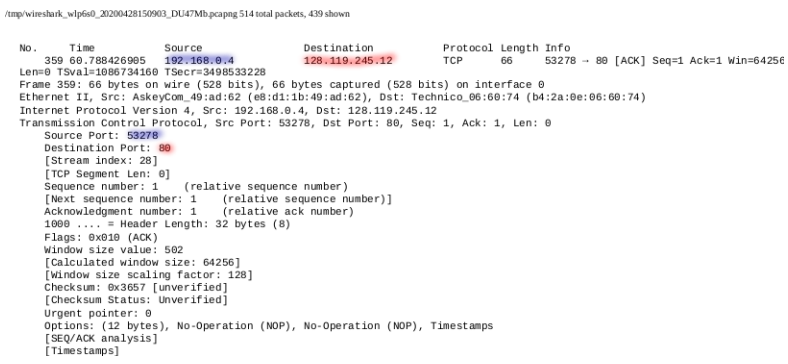

Figure 1: IP address and TCP port number used by the client computer

1

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?

According to figure 1, the IP address is 192.168.0.4 and the port number is 53278.

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?

According to figure 1, the destination IP is 128.119.245.12 and the destination port is 80.

3

What is the IP address and TCP port number used by your client computer (source) to transfer the file to gaia.cs.umass.edu?

According to figure 1, the IP address is 192.168.0.4 and the port number is 53278.

4

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?

According to figure 2, the sequence number of the TCP SYN segment that is used to initiate the TCP connection is 0. This segment is identified by the SYN tag being set to 1.

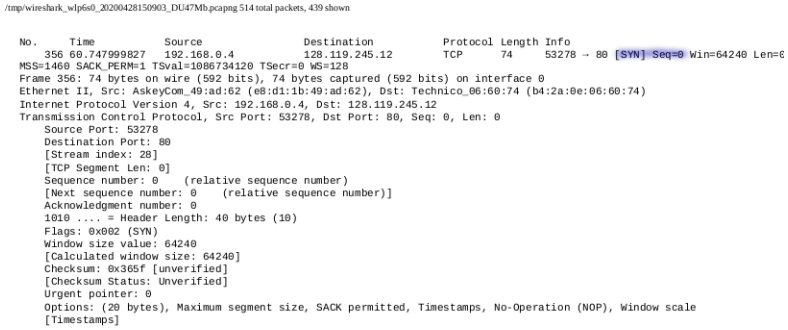

Figure 2: TCP SYN segment that is used to initiate the TCP connection

5

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?

According to figure 3, the sequence number of the SYNACK segment sent by gaia.cs.umass.edu to the client computer is 0. The value of the ACK is 1. This number is determined by adding 1 to the initial sequence number of the SYN segment from the client to the computer. The SYN flag and ACK flag are set to 1 and they indicate that this segment is an SYNACK segment.

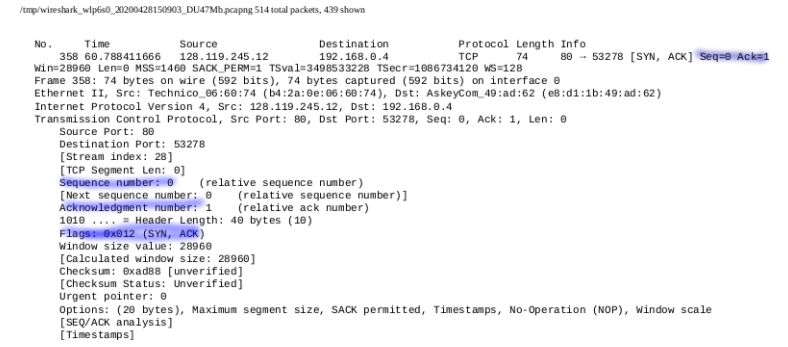

Figure 3: SYNACK segment sent by gaia.cs.umass.edu

6

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

According to figure 4, the sequence number of the TCP segment containing the HTTP POST command is 149145.

Figure 4: Sequence number of the TCP segment containing the HTTP POST command

7

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?

According to figure 5, the sequence numbers of the first 6 segments are 1, 579, 716, 2164, 3612, 5060. Figure 6 shows the send time, received (ACK) time and RTT.

The required formula is $EstimatedRTT_s = 0.875 * EstimatedRTT + 0.125 * SampleRTT$.

$EstimatedRTT_1 = 0.0956$

$EstimatedRTT_2 = 0.0956$

$EstimatedRTT_3 = 0.0958$

$EstimatedRTT_4 = 0.0961$

$EstimatedRTT_5 = 0.0982$

$EstimatedRTT_6 = 0.1001$

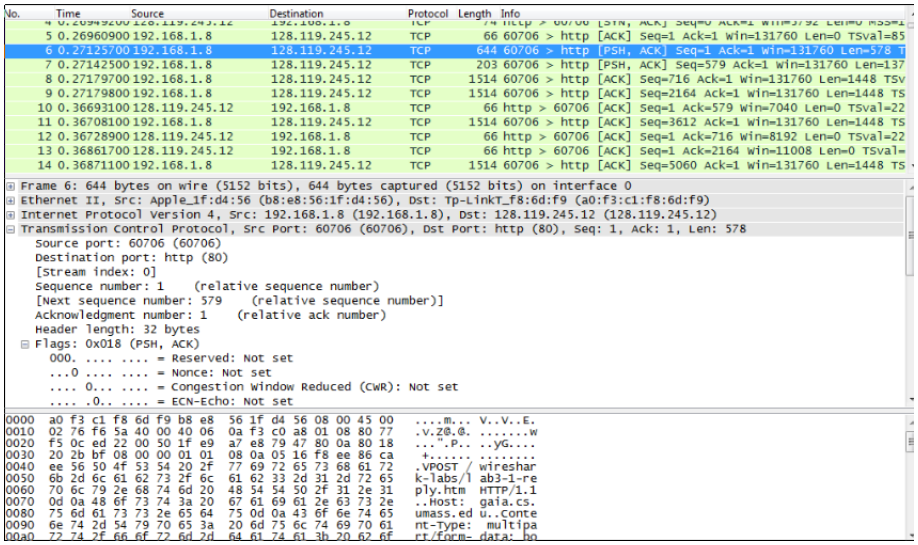

Figure 5: First 6 segments

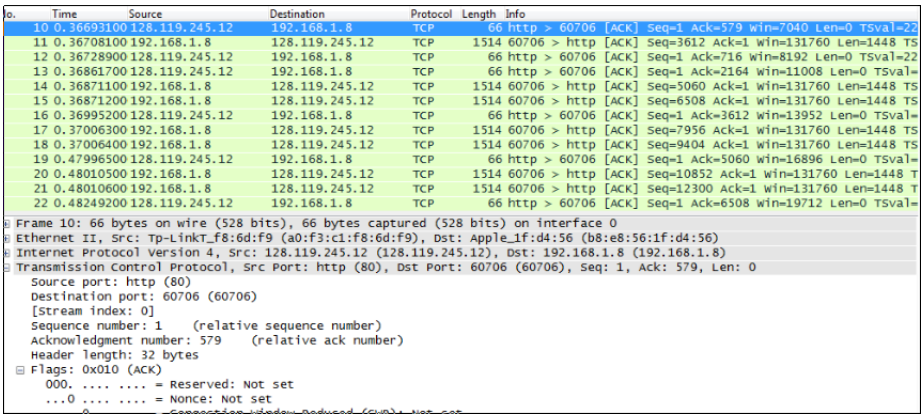

Figure 6: First 6 segments ACKs

	Sent time	ACK received time	RTT
Segment 1	0.271257000	0.366931000	0.095674
Segment 2	0.271425000	0.367289000	0.095864
Segment 3	0.271797000	0.368617000	0.09682
Segment 4	0.271798000	0.369952000	0.098154
Segment 5	0.367081000	0.479965000	0.112884
Segment 6	0.368711000	0.482492000	0.113781

Figure 7: Sending and receiving time of AKCs

8

What is the length of each of the first six TCP segments?

According to figure 8, the legnths of the first 6 segments are 74, 74, 66, 1514, 1514, 1514.

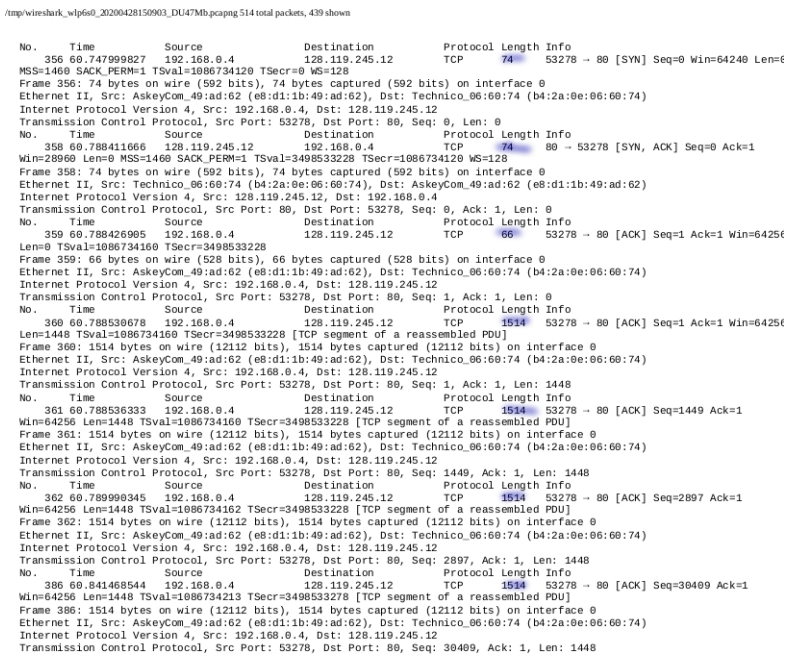

Figure 8: Length of each of the first six TCP segments

9

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?

According to figure 9, the minimum amount of available buffer space advertised at the received for the entire trace is 28960. Throughout the trace, the window grows until it reaches a max buffer size. Thus the sender is never throttled due to a lack of received buffer space.

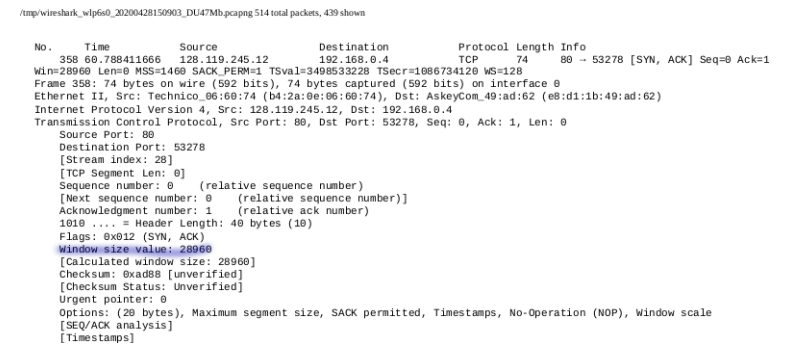

Figure 9: Minimum amount of available buffer space

10

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

According to figure 10, there are no retransmitted segments in the trace file. This can be checked by observing the sequence numbers of the TCP segments. All the sequence numbers increase monotonically with respect to time.

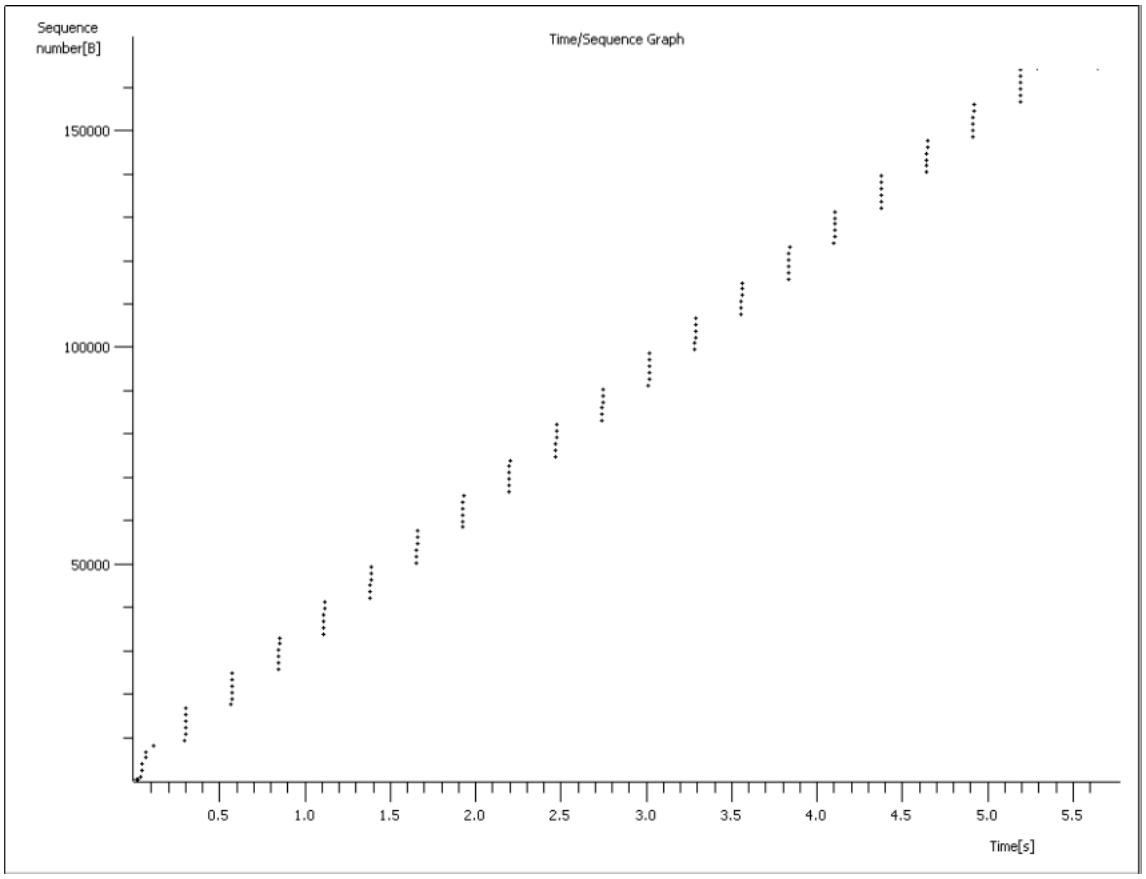

Figure 10: Time sequence graph

11

How much data does the receiver typically acknowledge in an ACK? Can you identify cases where the receiver is ACKing every other received segment (see Table 3.2 on page 250 in the text).

The amount of data being transfered can be found by observing the difference of the sequence numbers. It typically acknowledges 1460. According to the table, there are instances where the receiver is ACKing every other segment.

	acknowledged sequence number	acknowledged data
ACK 1	566	566
ACK 2	2026	1460
ACK 3	3486	1460
ACK 4	4946	1460
ACK 5	6406	1460
ACK 6	7866	1460
ACK 7	9013	1147
ACK 8	10473	1460
ACK 9	11933	1460
ACK 10	13393	1460
ACK 11	14853	1460
ACK 12	16313	1460

Figure 11: Table of ACK data

12

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

The througput can be calculated by dividing the total data by the total time taken. The total data is determined by finding the difference of the first and last sequence number. According to figure 12, the total time taken is $60.987296150 - 60.788426905 = 0.19886924$ and the difference of sequence numbers is $149362 - 1 = 149361$. Thus the throughput is $\frac{149361}{0.19886924} = 751051.2938049$.

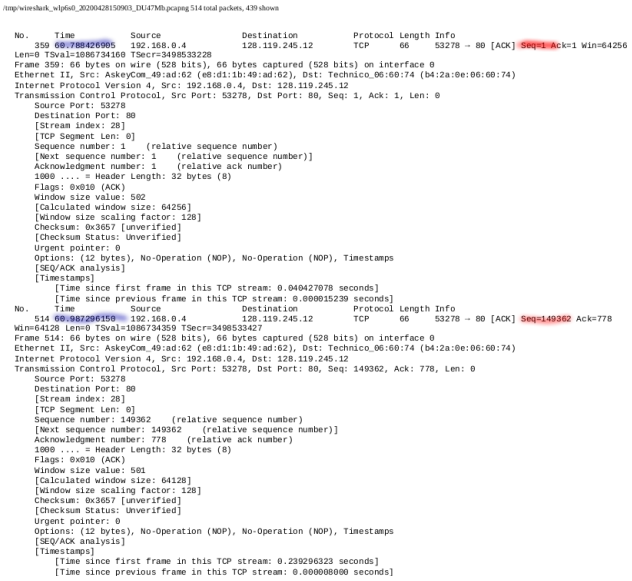

Figure 12: Firt and last packet data

13

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 slowstart phase begins and ends, and where congestion avoidance takes over? Comment on ways in which the measured data differs from the idealized behavior of TCP that we've studied in the text.

According to figure 13, TCP appears to begin at 0.27 seconds and ends at 10.35 seconds. Congestion avoidance begins at about 0.7 seconds. We know this is occuring because there is a reduced quantity of packets being disbatched. We expect the sequence numbers to increase lineraly. Here we can see that packets are sent in batches of 6, also, there is clearly non-linear behavior, especially at the start of the graph.

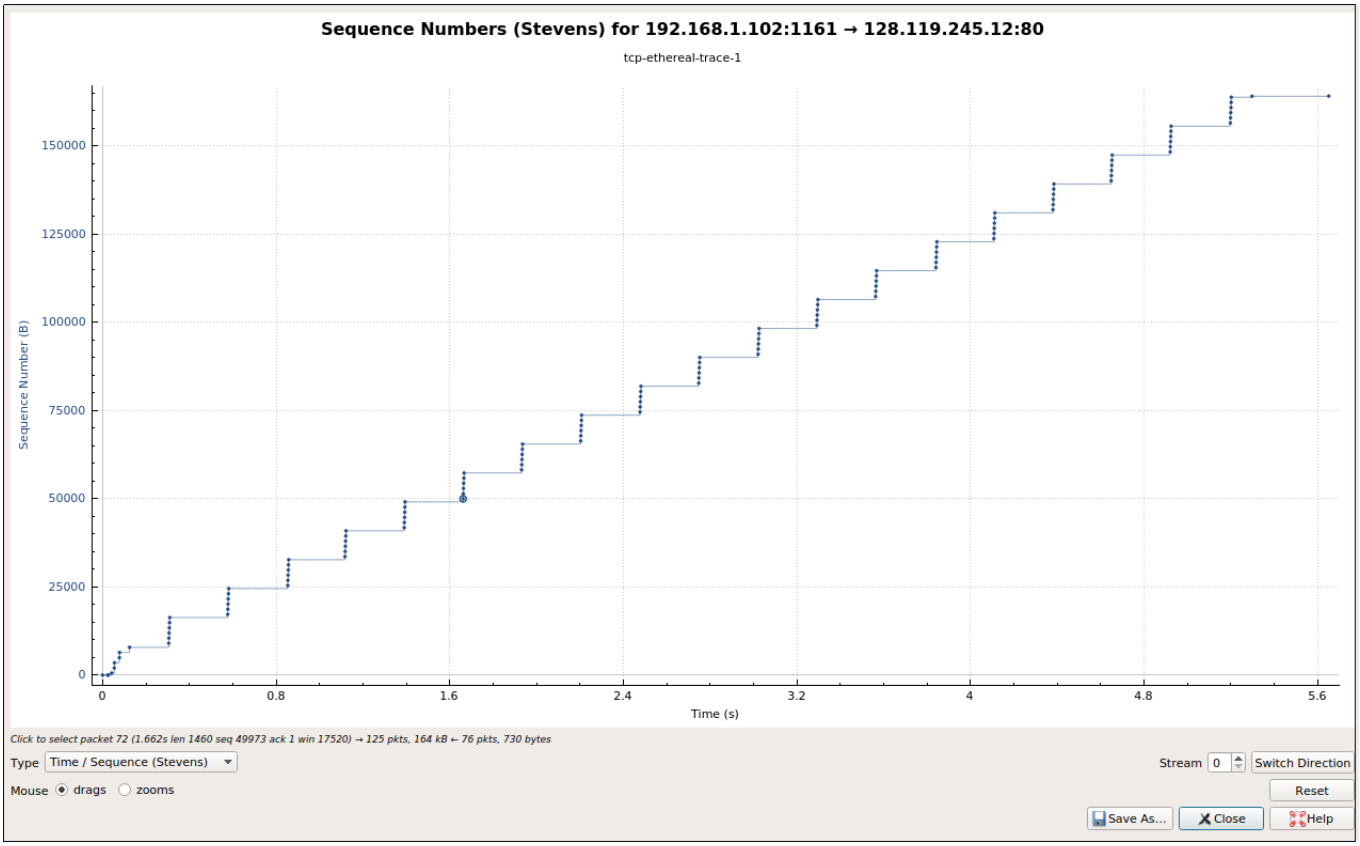

Figure 13: Time sequence graph

14

Answer each of two questions above for the trace that you have gathered when you transferred a file from your computer to gaia.cs.umass.edu.

According to figure 14, it appears to begin at 0.04 seconds and ends at 0.2 seconds. Congestion avoidance begins at 0.8 seconds when it begins changing the window size. This graph is more linear than the previous example with some gaps inbetween the batches. The ideal case is a perfectly linear graph.

According to figure 15, it appears to begin at 0.04 seconds and ends at 0.24 seconds. Congestion avoidance begins at about 0.9 seconds when the window size begins to increase. The figure does not resemble the ideal graph at all. It has large spikes of sequence numbers and the flattens with the next batch.

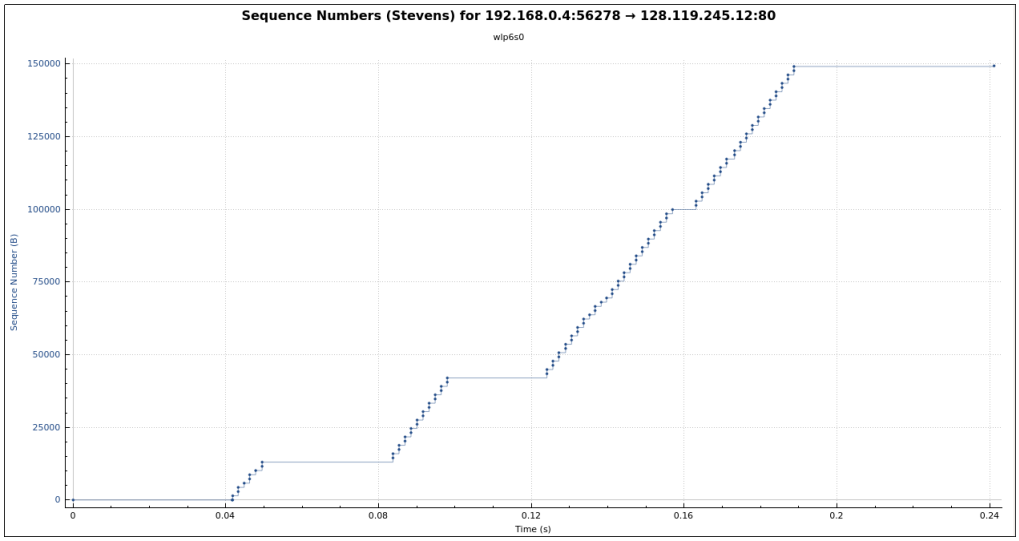

Figure 14: Time sequence graph

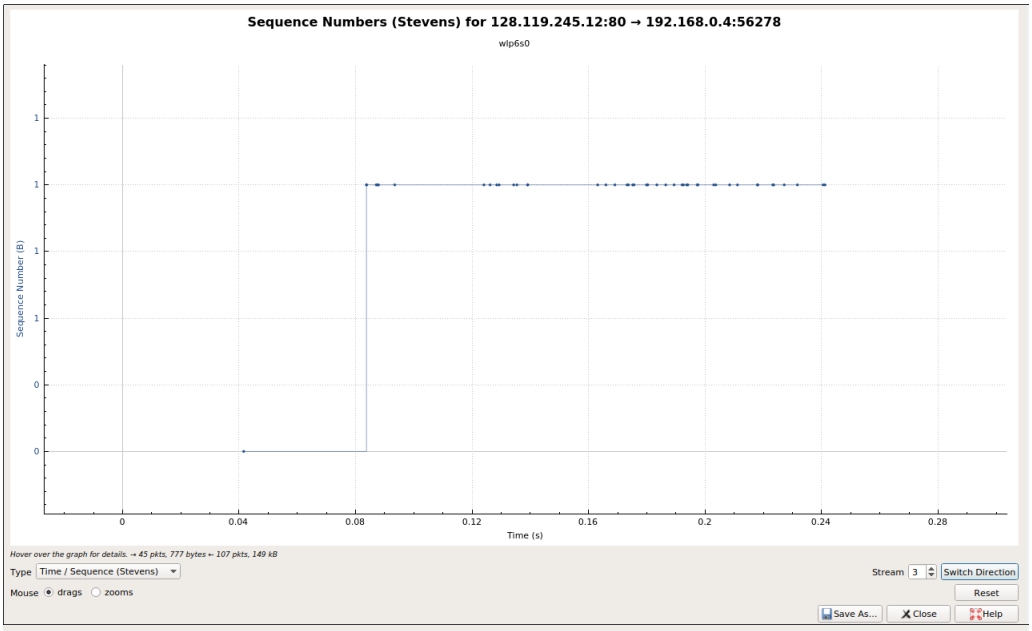

Figure 15: Time sequence graph