Part II:

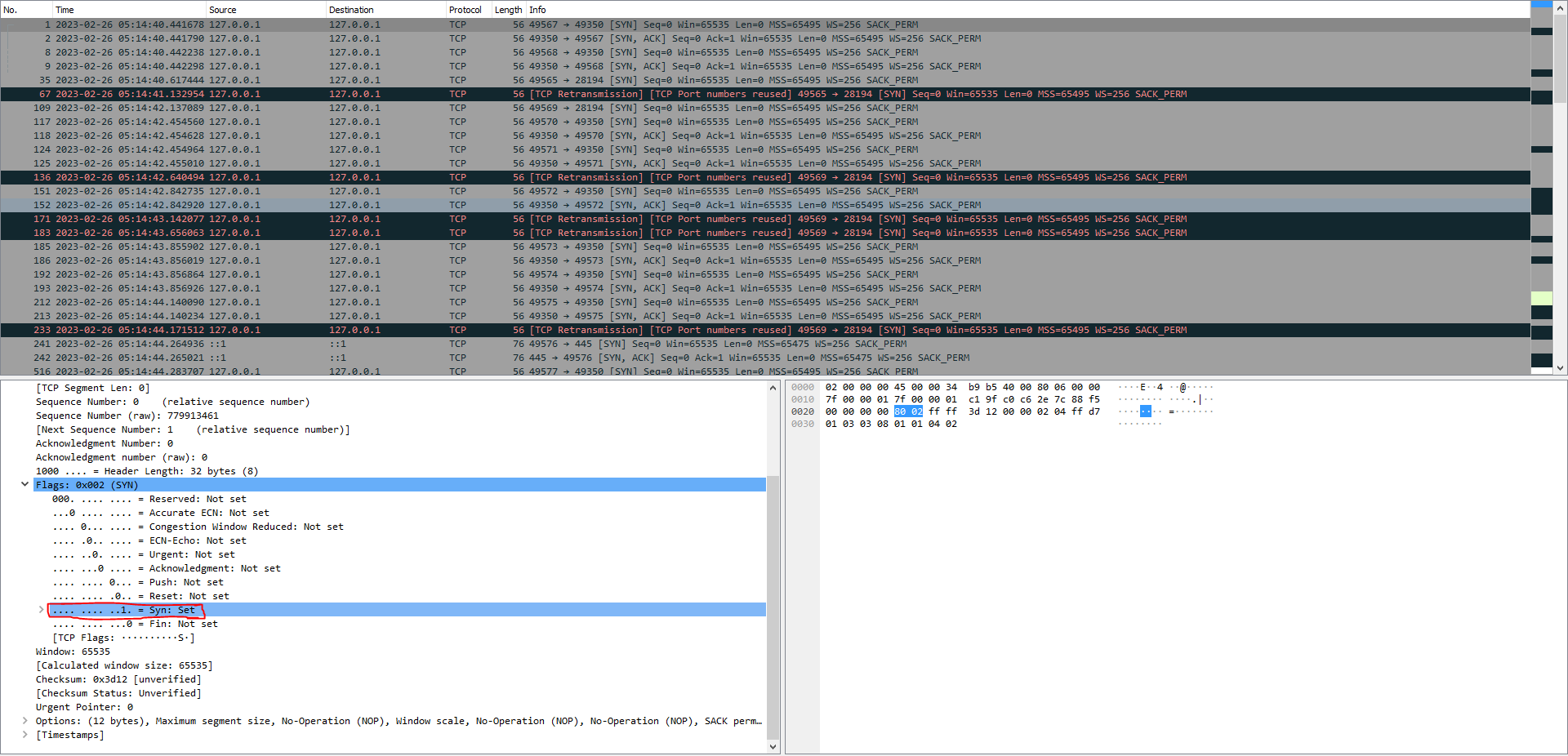
1.The URL didn't change for the data to be uploaded. HTTP supports several methods for data or form uploads, including POST, PUT, and PATCH. However, the most commonly used method for file uploads is POST. The HTTP/1.1 specification defines the POST method as the default method for submitting forms and uploading files.

2a. tcp.flags.syn==1 or tcp.flags.ack==0

2b. The browser (client)

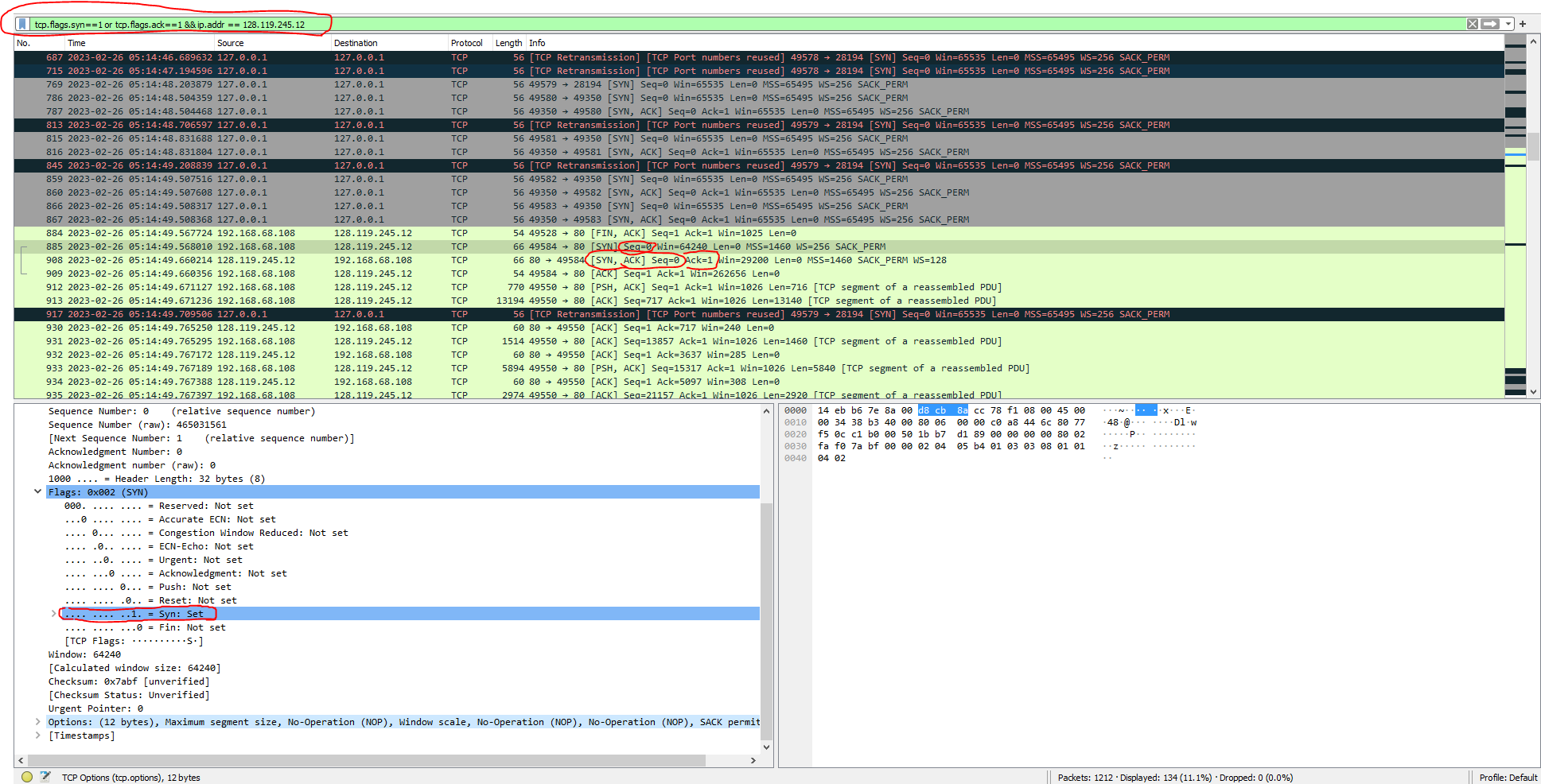
2c. The TCP protocol assigns an ISN to each new byte, beginning with 0.

Filter it on Wireshark with “tcp.flags.syn==1”, then select any of the displayed packets to view their details in the Wireshark packet details pane. The TCP header section of the packet details will show the SYN flag set to 1, which identifies the segment as a SYN segment.

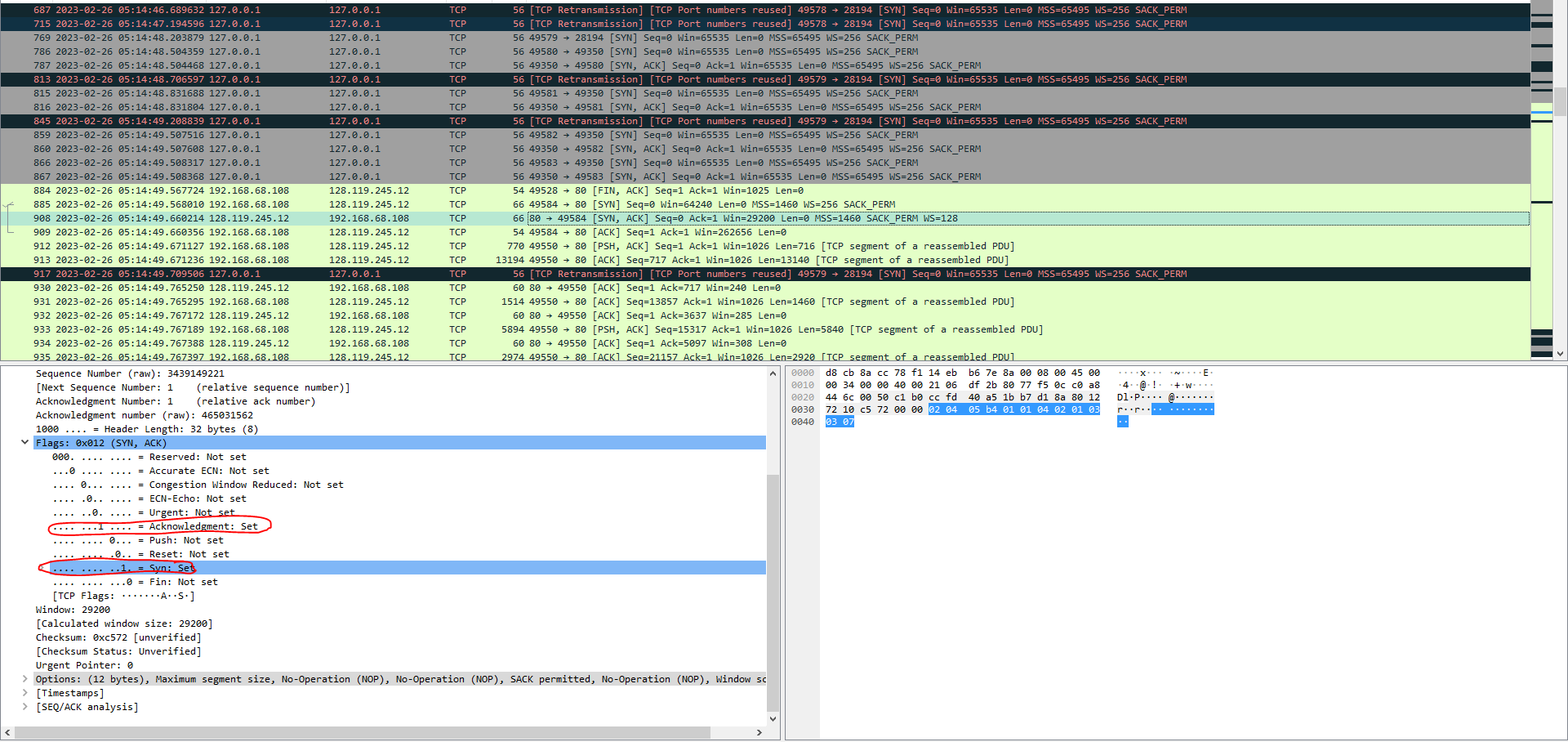


2d. 0 Yes it is the same but they are not related. Instead it is considered to be the first packet sent by the server, that’s why it is 0.

ACK = 1. First sequence number sent by the client was Seq = 0, now it wants the client to transmit the next packet with seq = 1.



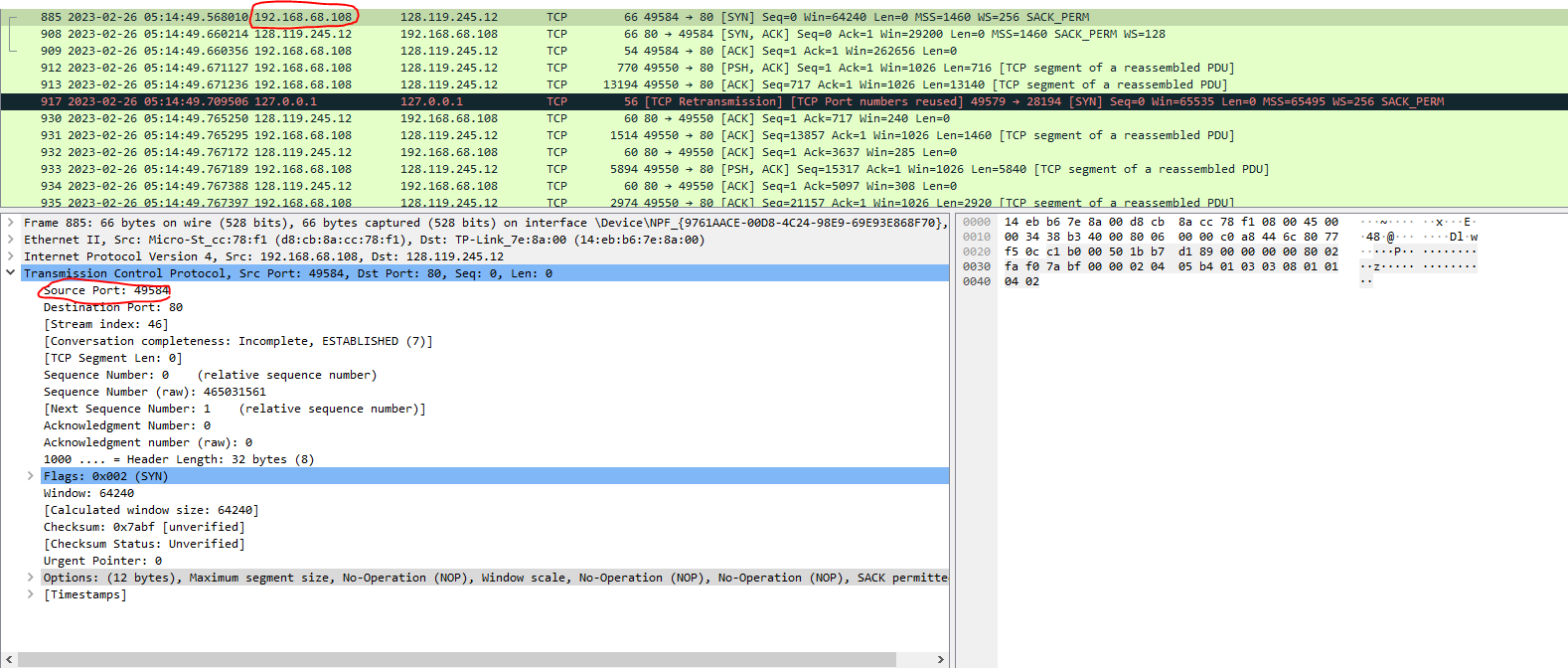
In the "Filter" field, type in the filter expression "tcp.flags.syn==1 && tcp.flags.ack==1”. You can then select any of the displayed packets to view their details in the Wireshark packet details pane. Only the packets that have both the SYN and ACK flags set to 1, which indicates that they are SYN-ACK segments



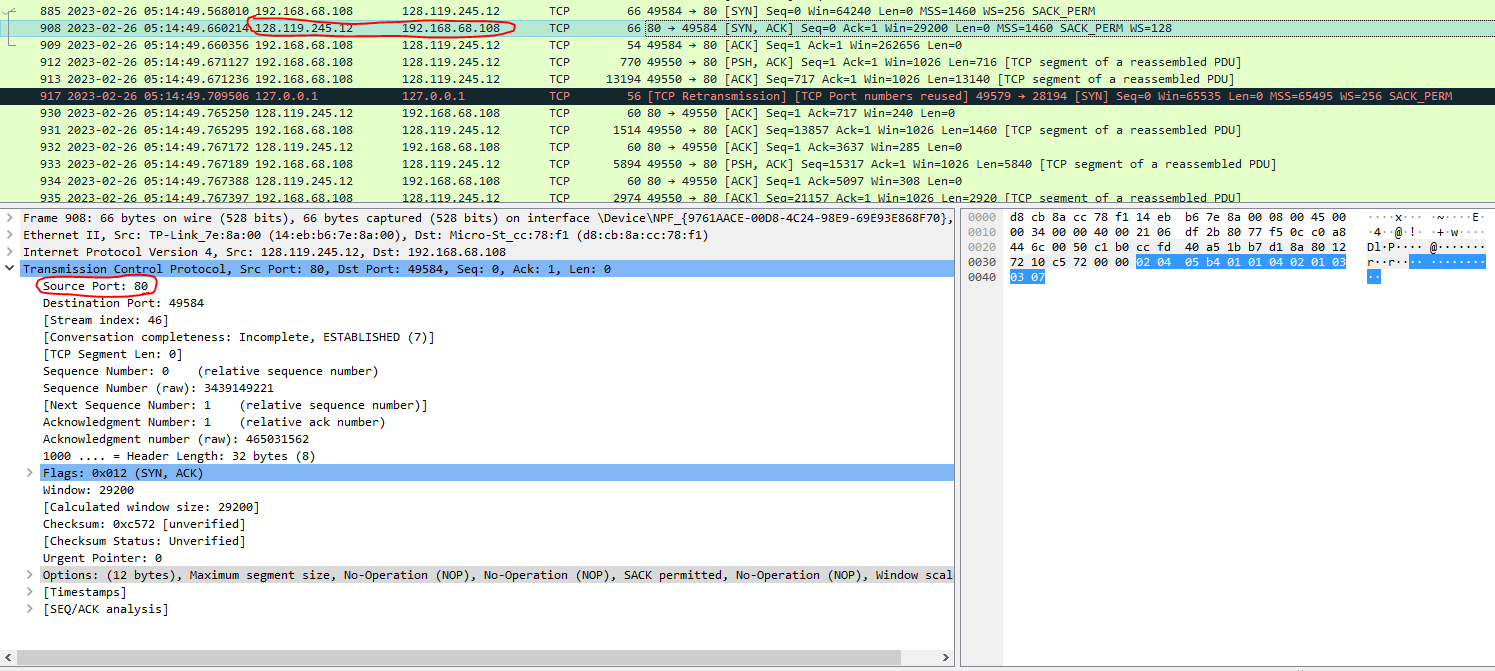
2e. The time elapsed from when the SYN is transmitted until the SYN ACK is received is approximately 92 milliseconds. This time represents the network round-trip time (RTT) between the client and server. The RTT is the time it takes for a packet to travel from the sender to the receiver and back. In this case, the client sends a SYN packet to the server, and the server responds with a SYN ACK packet. The time it takes for the SYN packet to reach the server and for the SYN ACK packet to return to the client is the RTT.



2f. 192.168368.108. Port: 49584, it is not a well known port, therefore an ephemeral port.



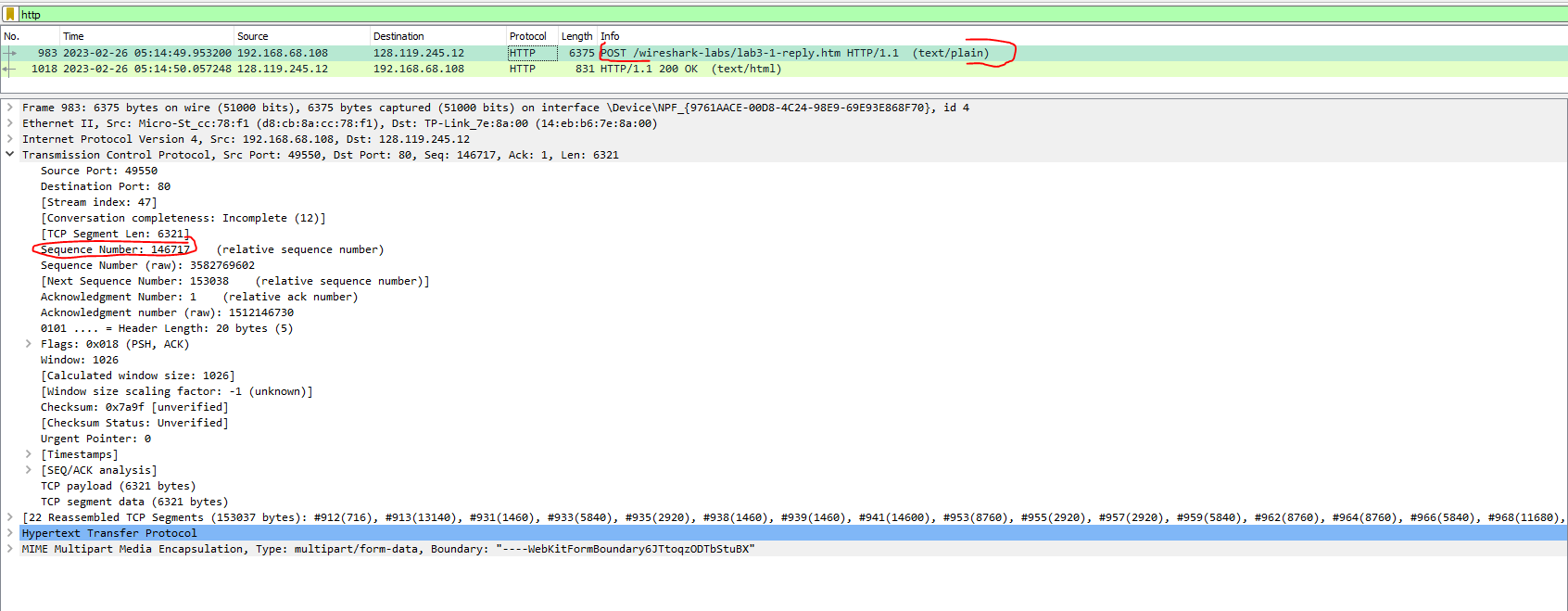
2g. 128.119.245.12. Port: 80. It is a well known port.



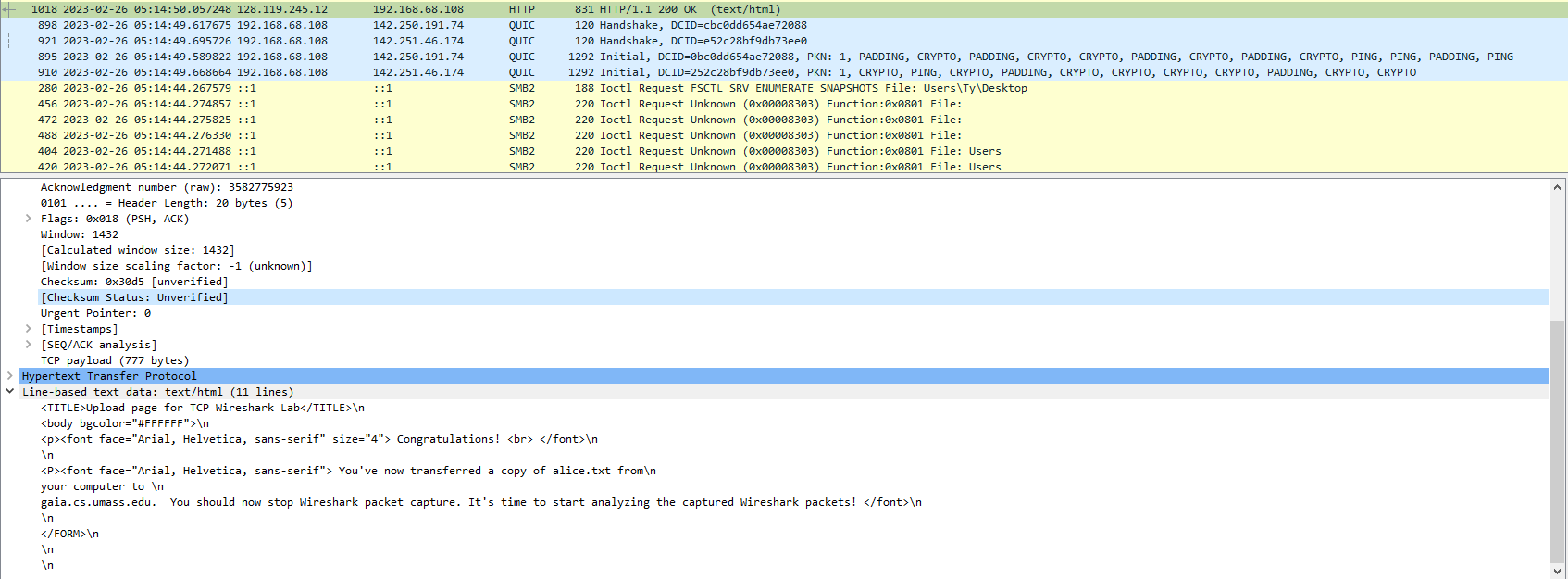
3a.. The HTTP method used to upload the file is POST.

3a-i. To find this, I used the display filter "http”

3a.ii. 146717



3b.



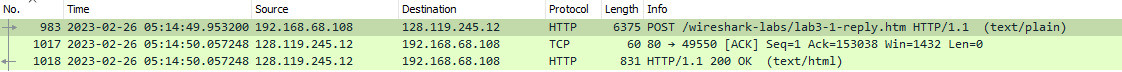
3c. 44.596 milliseconds for RTT (Data Packet)

92 ms + 44.496 ms = 136.49600 ms



4.a-b-c

| # | SN | Time transmitted | Time ACK received | #bytes transferred | Sample RTT | Estimated RTT |
| --- | --- | --- | --- | --- | --- | --- |
| 1 | 146717 | 2023-02-26 05:14:49.953200 | 2023-02-26 05:14:50.057248 | 6321 | 104.048 | 104.048 ms |
| 2 | 105837 | 2023-02-26 05:14:49.944128 | 2023-02-26 05:14:50.03504 | 8760 | 90.912ms | 102.406ms |
| 3 | 62037 | 2023-02-26 05:14:49.863466 | 2023-02-26 05:14:49.953181 | 8760 | 89.715ms | 100.819625 ms |





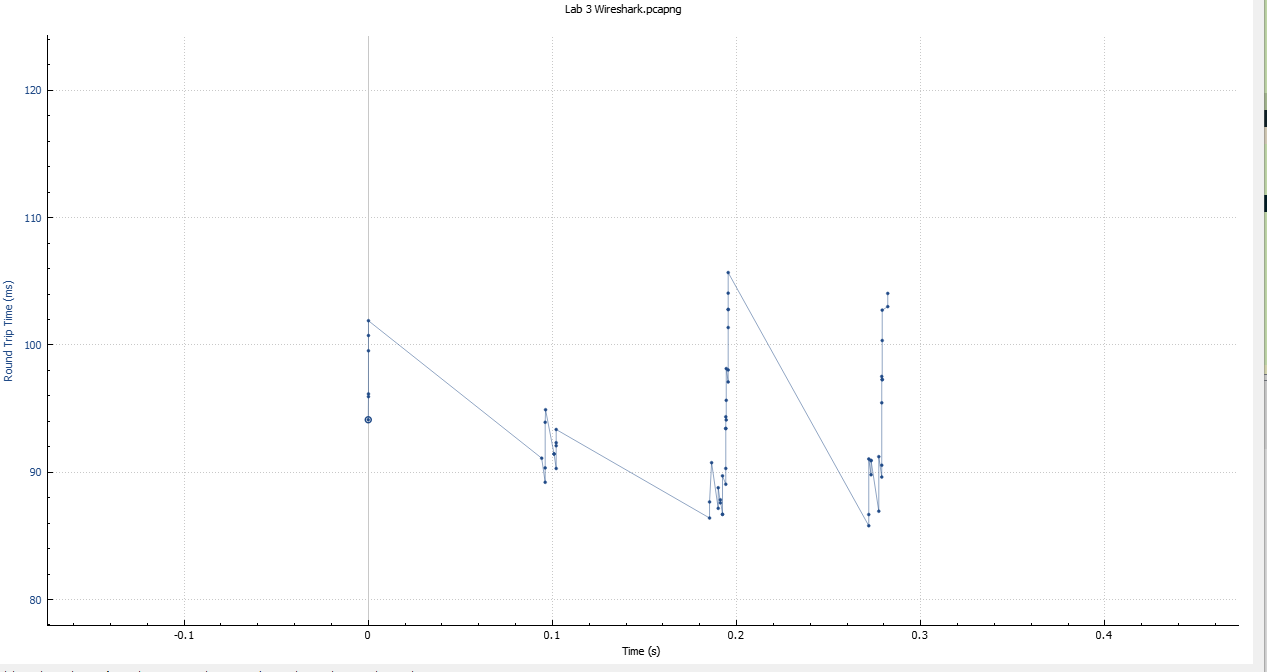




5. Estimated RTT1 = Sample RTT1 = 104.048 ms

Estimated RTT2 = (1 - 0.125) \* 104.048 ms + 0.125 \* 90.912 ms = 102.406ms

Estimated RTT3 = (1 - α) \* Estimated RTT2 + α \* Sample RTT3 = (1 - 0.125) \* 102.406ms + 0.125 \* 89.715 ms = 100.819625 ms

6. It compares quite similar to the calculations in the table, hovering around 100ms   


7. In TCP, the receiver typically acknowledges the receipt of data up to a sequence number (known as the "acknowledgment number") in the received TCP segment. The acknowledgment number is the next sequence number that the receiver is expecting to receive from the sender.  
As for determining which ACK scheme the receiver is following, there is typically only one standard ACK scheme used in TCP, which is the cumulative ACK scheme.

8a. 717

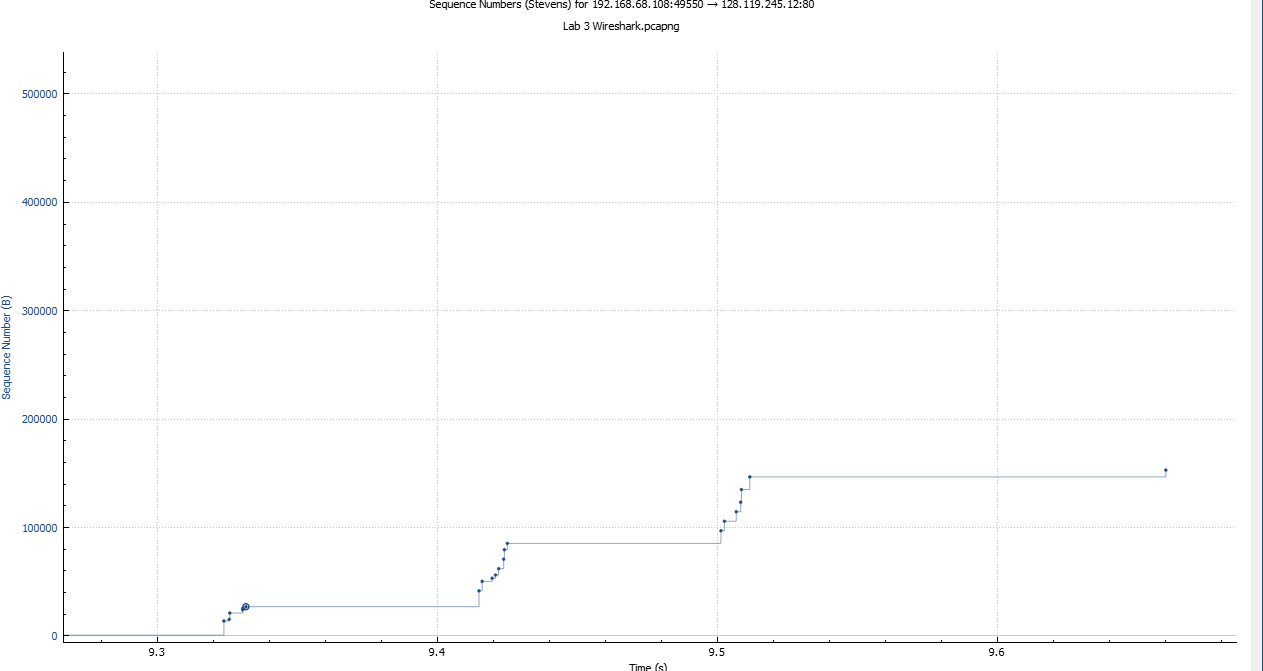
8b. 13160 bytes

8c. The x-axis of the graph represents time, and the y-axis represents the sequence number.

Each dot on the graph represents a TCP packet that has been captured by Wireshark. The position of the dot on the y-axis indicates the sequence number of the packet, while the position of the dot on the x-axis indicates the time at which the packet was captured.

8d. Slow start begins when a new TCP connection is established or when the sender has not sent data for a certain period of time (known as the idle timeout).

8e. No retransmitted segments.



9a. In TCP, cwnd (congestion window) is a variable that represents the maximum amount of data that a sender can transmit before receiving an acknowledgment (ACK) from the receiver. The cwnd is used to control the rate at which data is transmitted over the network, in order to avoid congesting the network.

9b. Maxes out at approximately 1432 bytes. Doesn’t seem to get any larger than that. I looked at the calculated window size on TCP @ Packet Details Pane

10.

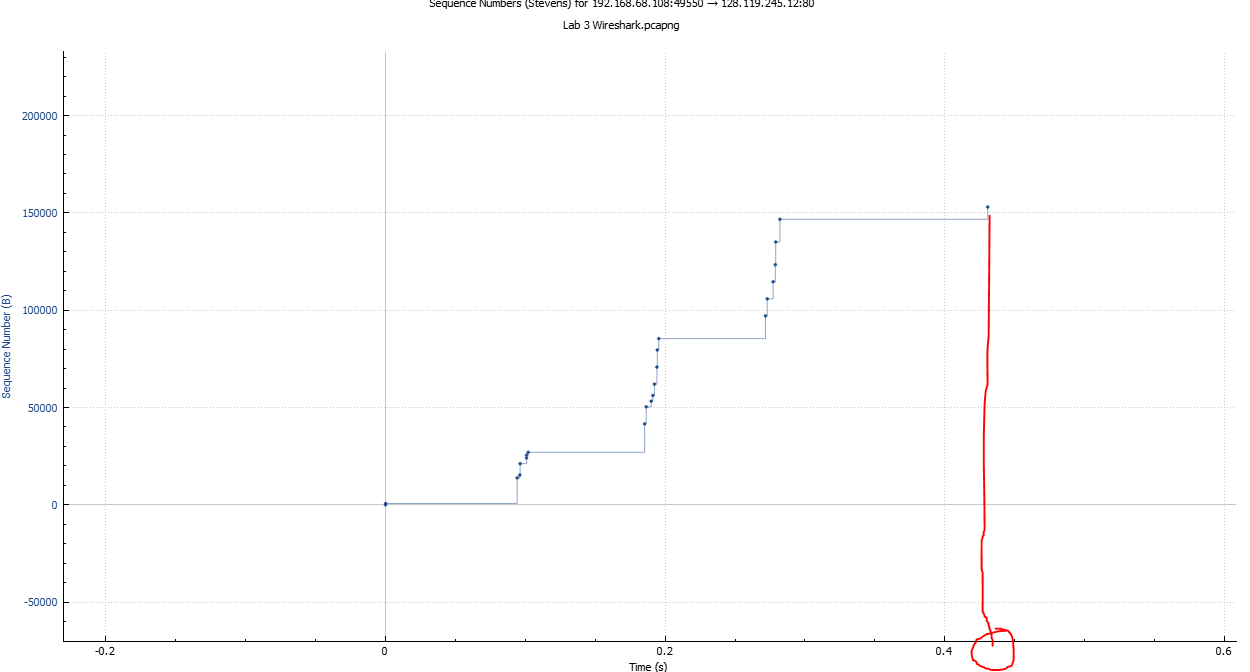
From the Sequence Numbers (Stevens) graph, we can see that:

Last ACK = 153038



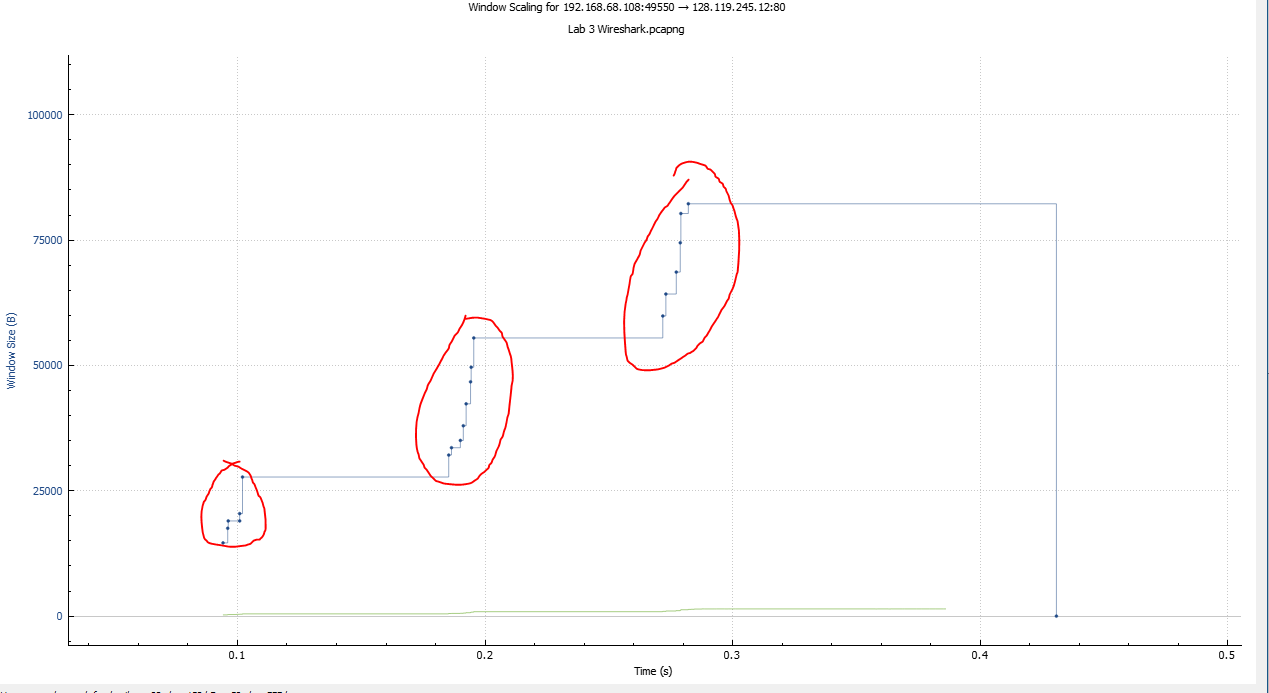
That means the total packet size transmitted is 153038 bytes

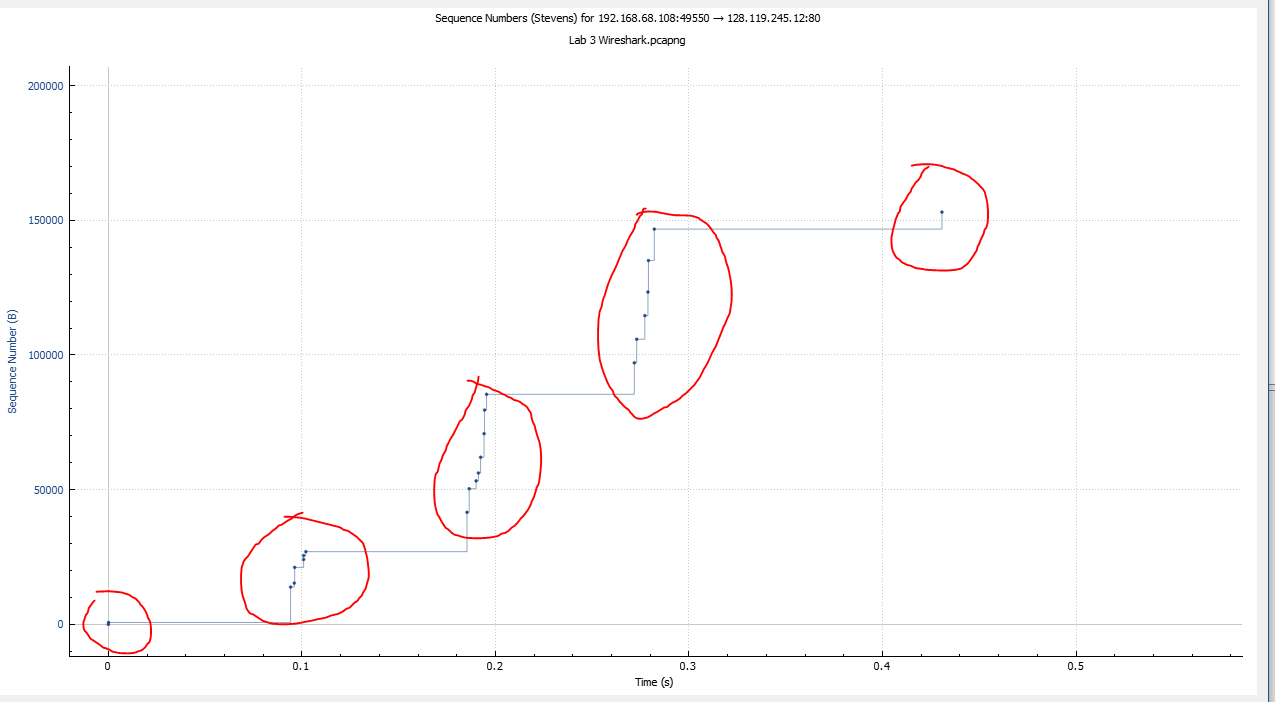
Duration of Transmission: 0.43 sec



So 153038 bytes/0.43 sec = **355902.326 bps or 355.902326 kBps**

11a.



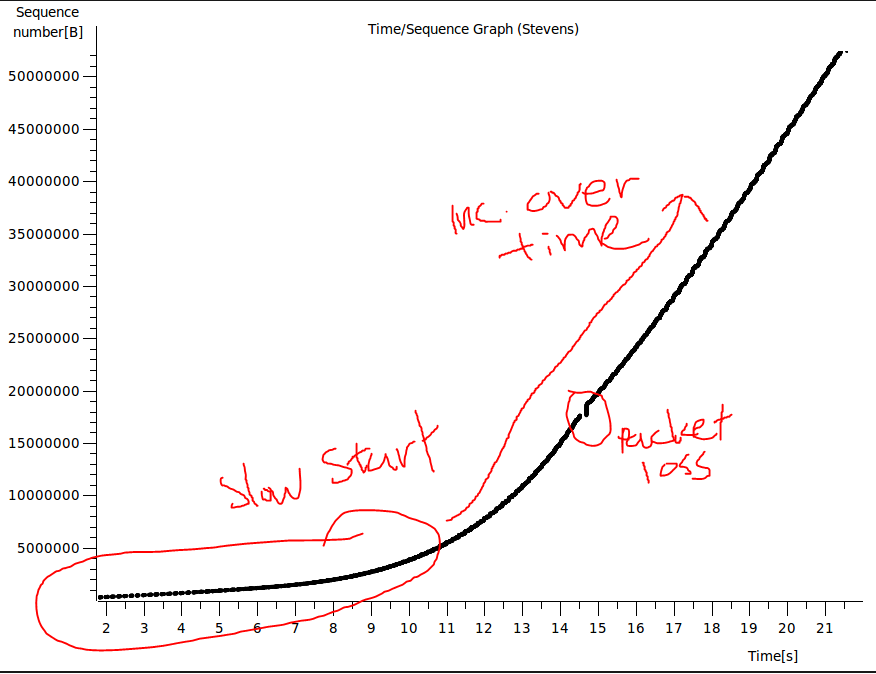


11b. Point to Point, Pinelined, Flow and Congestion Controlled.

11c. Client to Server

12a. Packet loss can still occur during the download even if there is no intentional introduction of loss.This can be due to a variety of factors, such as network congestion, hardware or software errors, and transmission issues.

12b.



12c.

i.Throughput is a measure of the amount of data that is transferred over a network or system in a given amount of time. It is usually expressed in bits per second (bps) or bytes per second (Bps). In this example, the throughput can be measured from the wget command's output, which reports the download speed in megabytes per second (MB/s). The output shows that the download speed was 2.36 MB/s, which means that the throughput was 2.36 megabytes per second.

ii.

First Packet:

5 2023-02-26 22:50:04.899017000 10.0.2.15 80.249.99.148 TCP 76 47636 > http [SYN] Seq=0 Win=29200 Len=0 MSS=1460 SACK\_PERM=1 TSval=109127 TSecr=0 WS=128

Last Packet:

18020 2023-02-26 22:50:26.644265000 10.0.2.15 80.249.99.148 TCP 56 47636 > http [ACK] Seq=140 Ack=52429087 Win=52968 Len=0

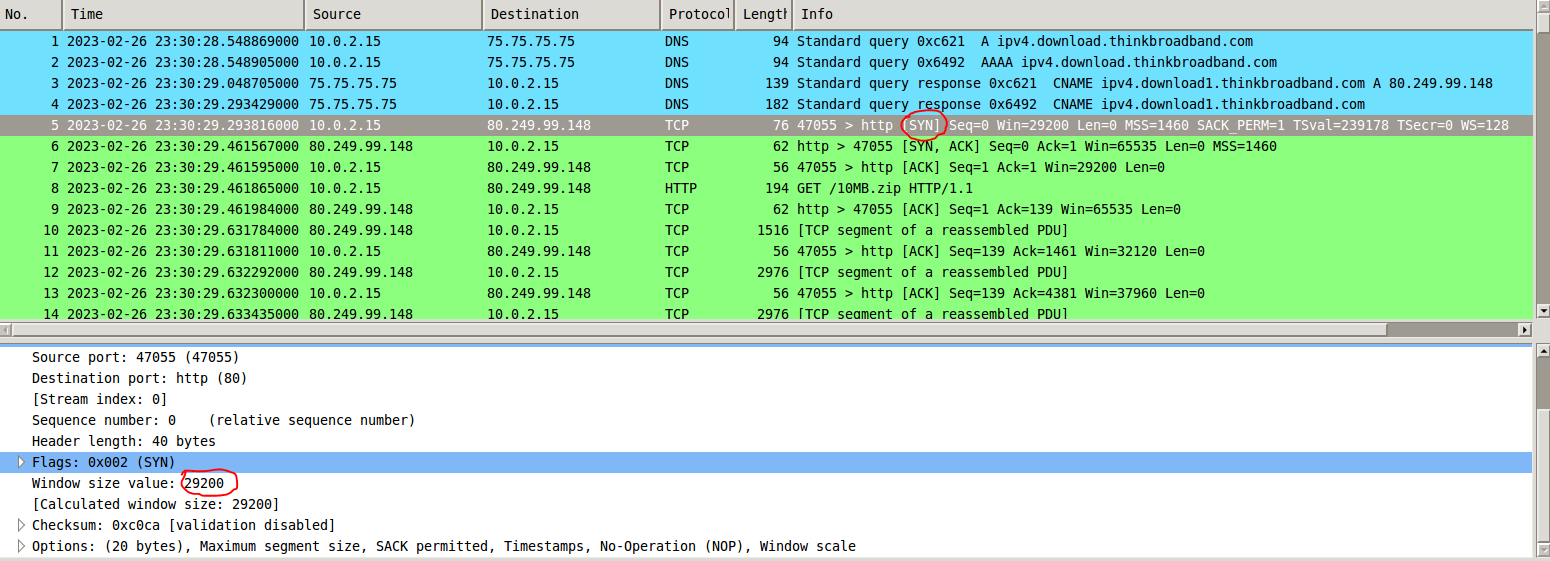
From the ACK, 52429087 bytes.  
21.745248 seconds elapsed.

**19 288 476.1 bits per second**

13a. In the TCP packet header, the window size field indicates the amount of data, in bytes, that the receiver is willing to accept at a given time from the sender without acknowledgement. It represents the receiver's buffer space for storing received data before processing it.

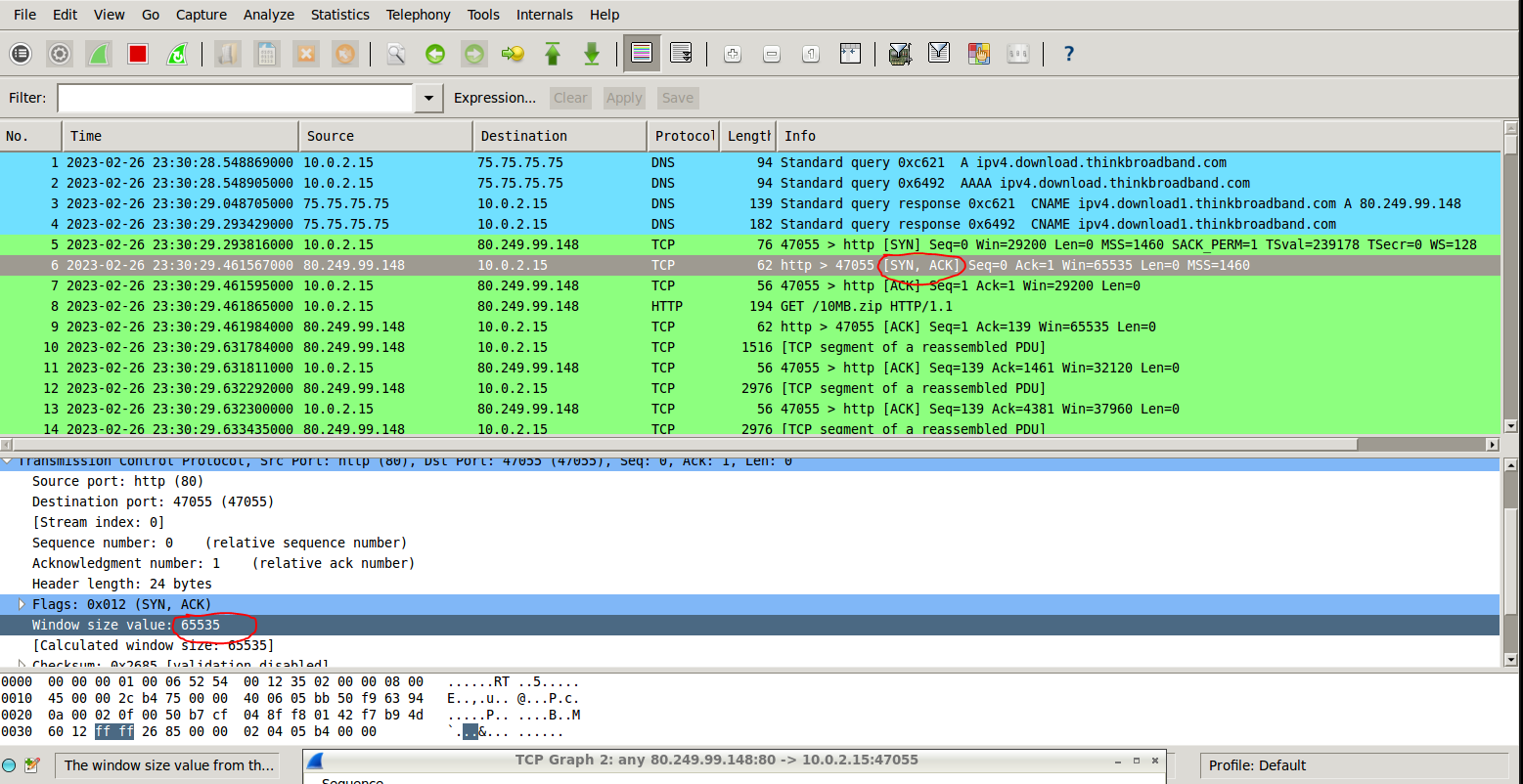
13b. SYN

29200 bytes



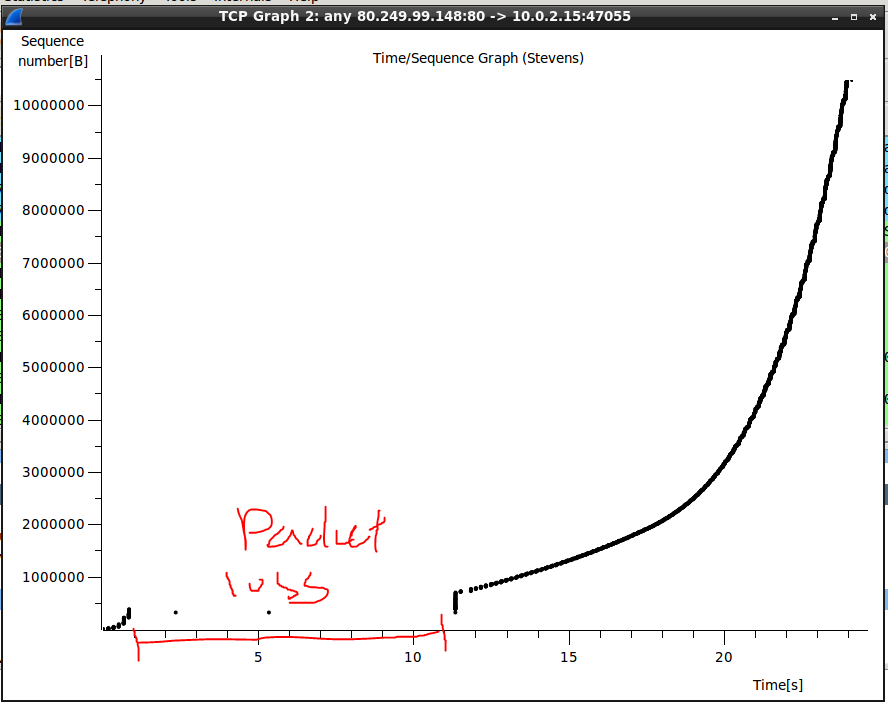
13c. SYN-ACK

65535 bytes



13d. No. We have different internet speed and we trigger the packet loss command at different times.

13e. Begin at 1 second, and ends at around 11 seconds.



f. It looks like it tries to accelerate at the start and is trying to catch up. Nearly a vertical up line.

g.



24.298276 seconds elapsed.

Last ACK #: 10486045, therefore 10486045 bytes total.  
431555.102922 bps

16.

a. A fundamental difference between UDP and TCP is the reliability of data transmission.

b. In the presence of packet loss, the average end-to-end delay to download a file using UDP is likely to increase.UDP is a connectionless protocol and does not provide any mechanisms for error recovery or packet retransmission. Therefore, if a packet is lost during transmission, the sender does not receive any feedback or acknowledgement from the receiver. As a result, the sender continues to transmit the subsequent packets, leading to a higher number of lost packets and increasing the overall delay.

c. If a packet is lost during file transfer using UDP, the sender will not receive any acknowledgement or feedback from the receiver.

d. Yes, from the UDP sender's perspective, the time to deliver the file can change when there is packet loss.

e.In UDP, the receiver does not provide any feedback or acknowledgement to the sender about packet loss.