IT-305:-

Computer Networks

**Lab Report-03**

**Name: Aarsh Bhavsar**

**ID: 202101474**

**Group: G6**

**Analyzing congestion policy, RTT of TCP using NetSim and wireshark.**

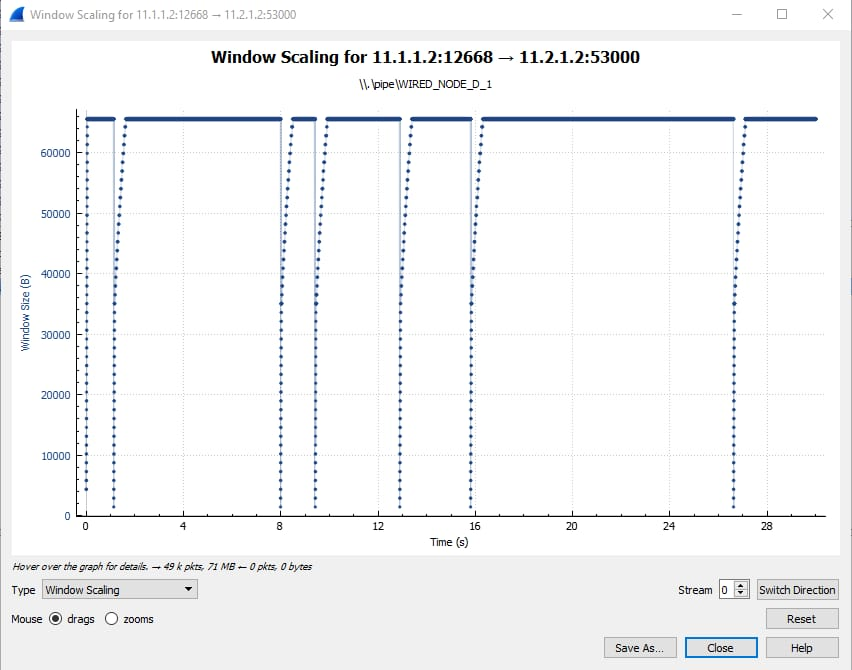
**1.2 Exercise:**

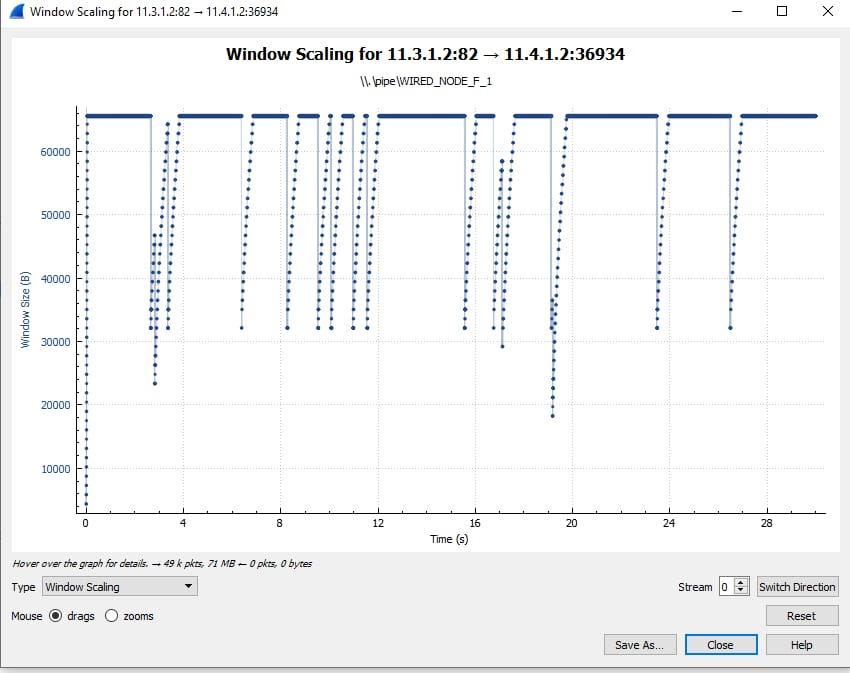
1. **For both the variant, analyze graph of congestion window, answer the following by marking in the graph.**

**(a) Identify the event of TCP slow start.**

**(b) Identify the event of packet loss and time out.**

**(c) Identify the intervals of time when TCP congestion avoidance is operating.**





A) The graph obtained from my observations for Tahoe shows the slow start early on. And, it is observed in Reno around 2s.

B) The packet needs to be delivered again in cases of packet loss or timeout; these events cause abrupt drops in the graph.

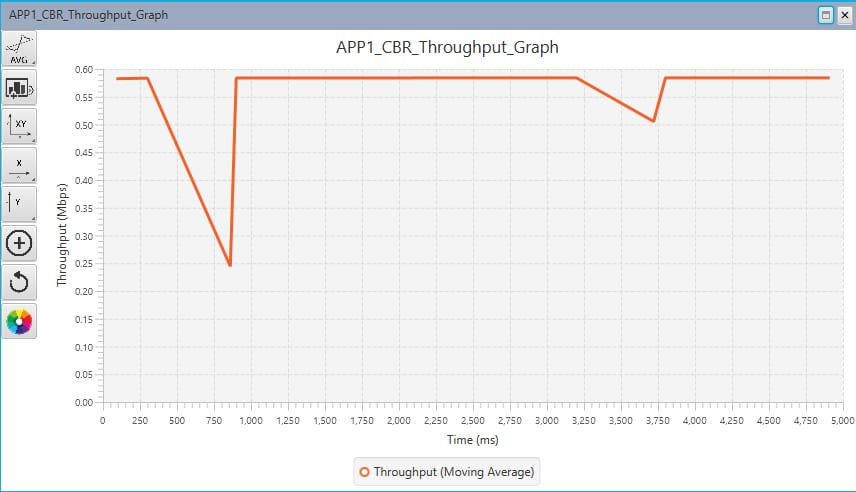
C) The abrupt rises in the graph that indicate the TCP segment is using the maximum sequence number feasible are known as congestion avoidance procedures.

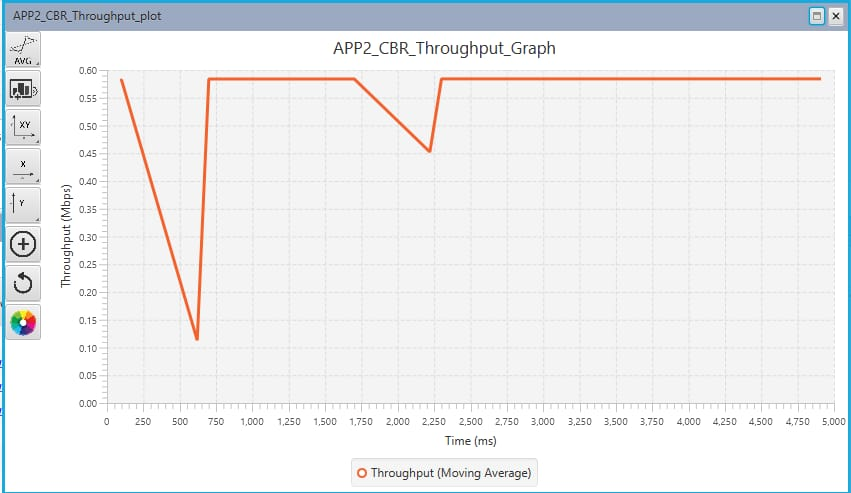
1. **What is the difference in congestion control policy of Tahoe and Reno, with respect to congestion avoidance and two events of congestion avoidance phase. Explain briefly in your log book.**

Tahoe and Reno are both TCP congestion control algorithms. Reno's fast recovery approach maintains better transmission during congestion, while Tahoe's approach is more cautious, reducing window size aggressively after losses.

**2.1 Exercise:**

**1. Take 3 wired nodes and one router and configure 2 identical CBR applications with default app specifications between them, as shown in Figure 4. Keep link properties as default. Run the simulation for 5 seconds. Check throughput for both applications and write down your observations.**

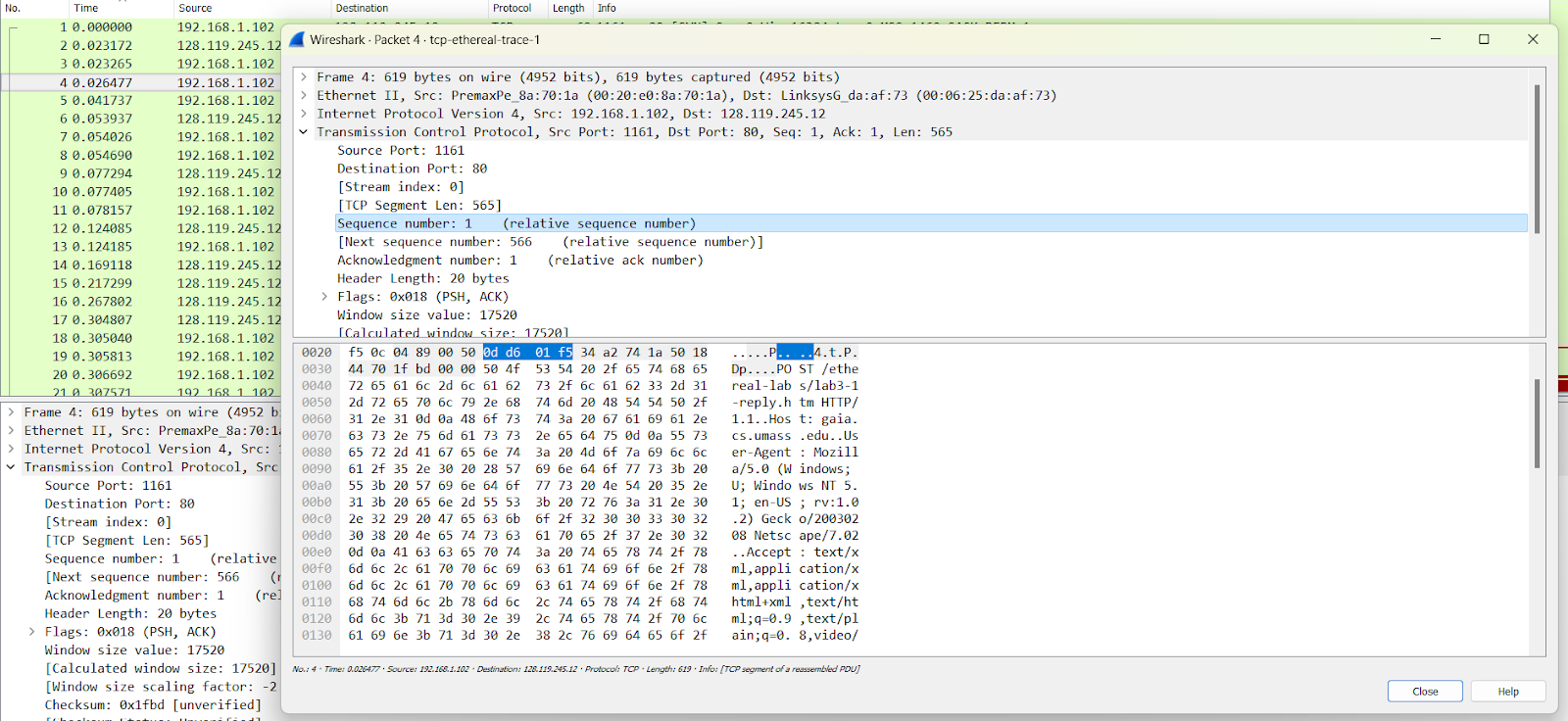


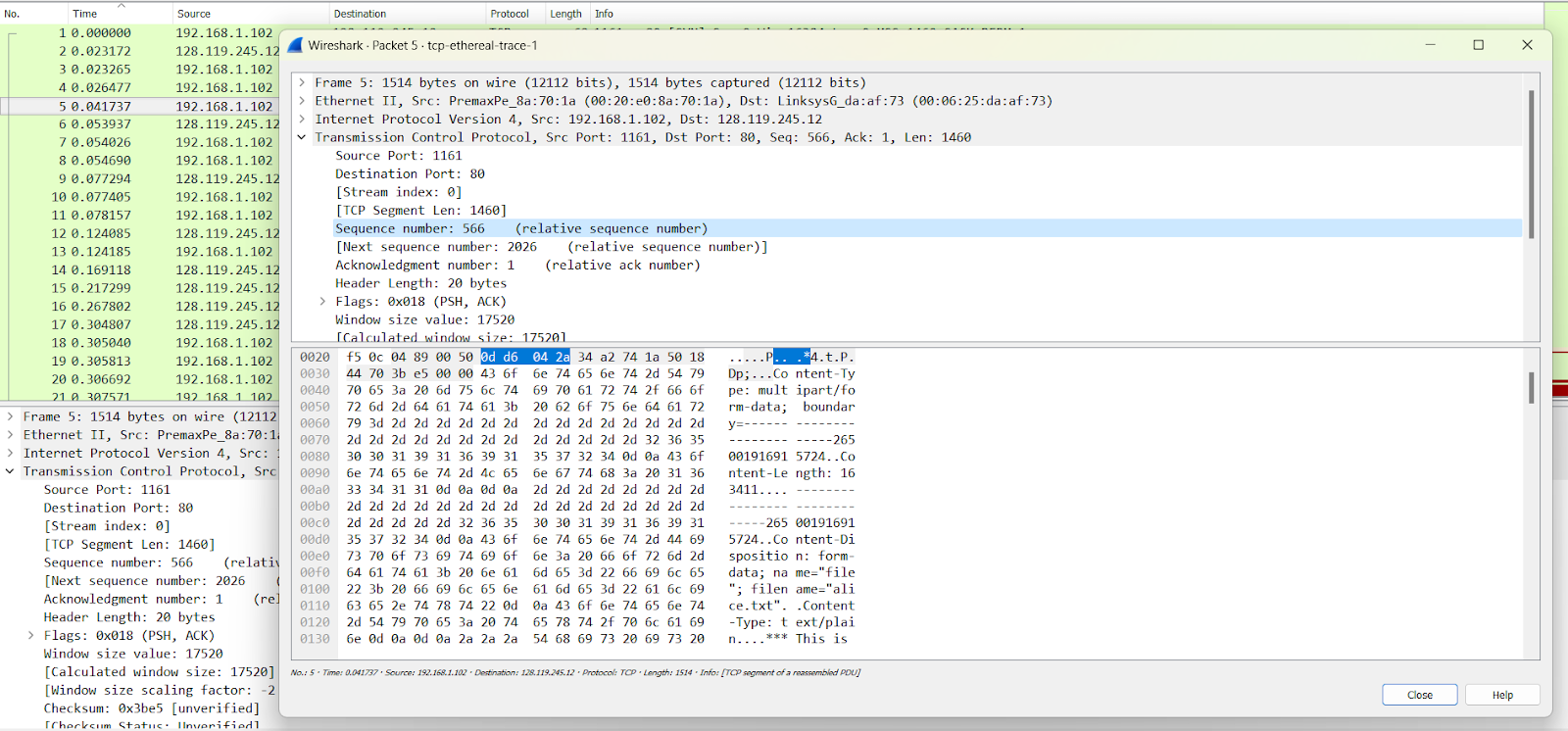
****

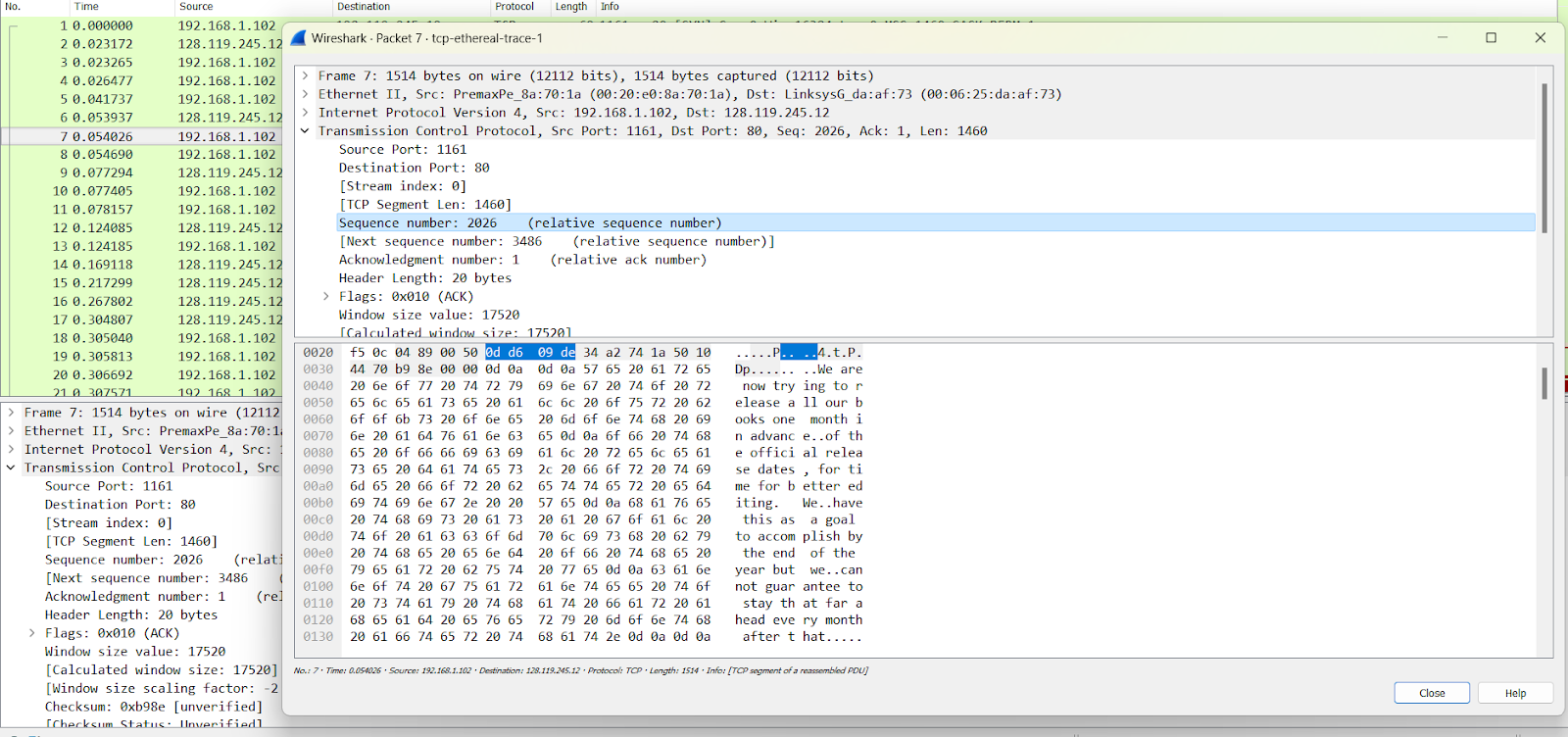
**3.2 Exercise:**

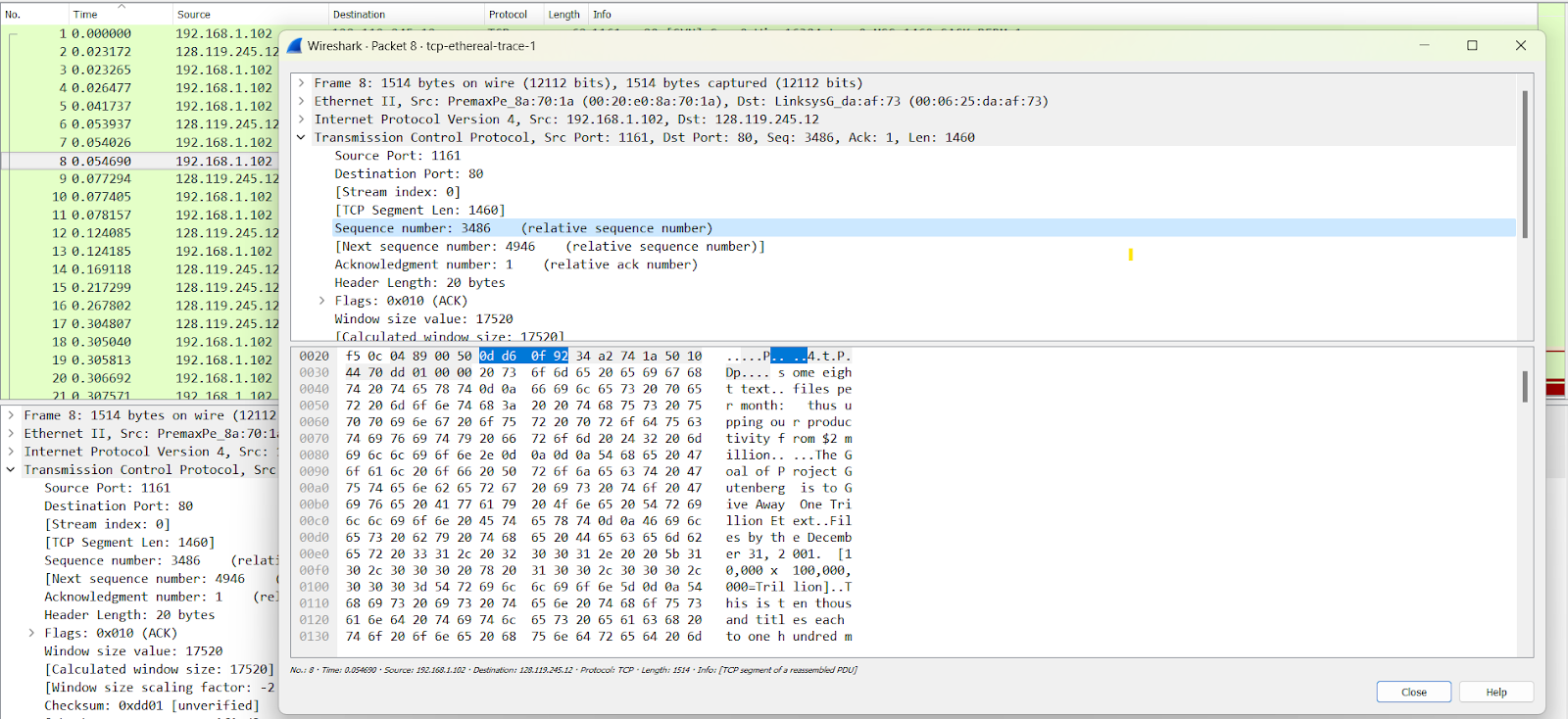
1. Estimated RTT = (0.875)\*EstimatedRTT + (0.125)\*RTT

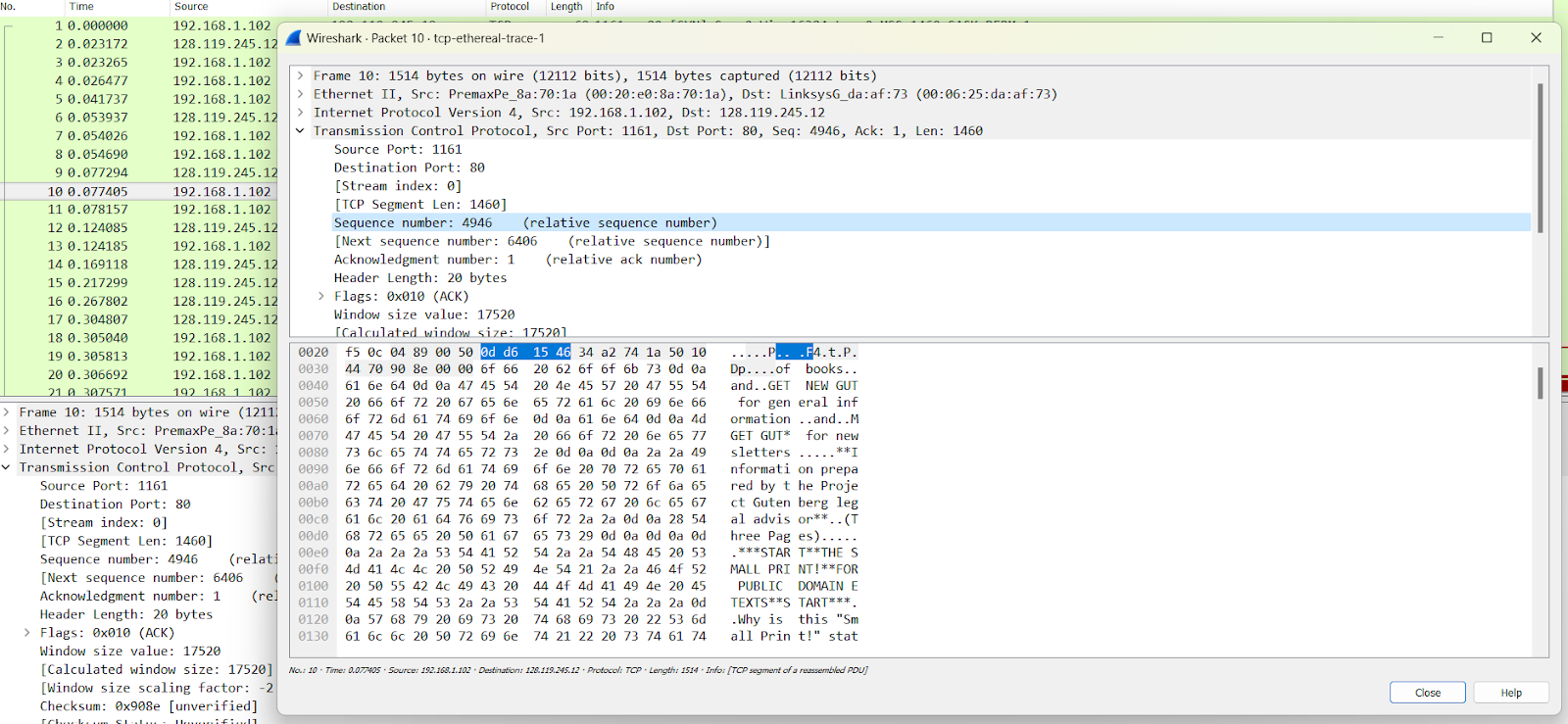
|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Seq.No | Sent Time | ACK received | RTT(ms) | Estimated RTT(ms) | Length(bytes) |
| 1 | 0.026477 | 0.053937 | 27.5 | 27.5 | 565 |
| 566 | 0.041737 | 0.077294 | 35.3 | 28.475 | 1460 |
| 2026 | 0.054026 | 0.124085 | 70.01 | 33.67 | 1460 |
| 3486 | 0.054690 | 0.169118 | 114.5 | 43.77 | 1460 |
| 4946 | 0.077405 | 0.217299 | 139.9 | 55.789 | 1460 |
| 6406 | 0.078157 | 0.267802 | 189.7 | 72.53 | 1460 |

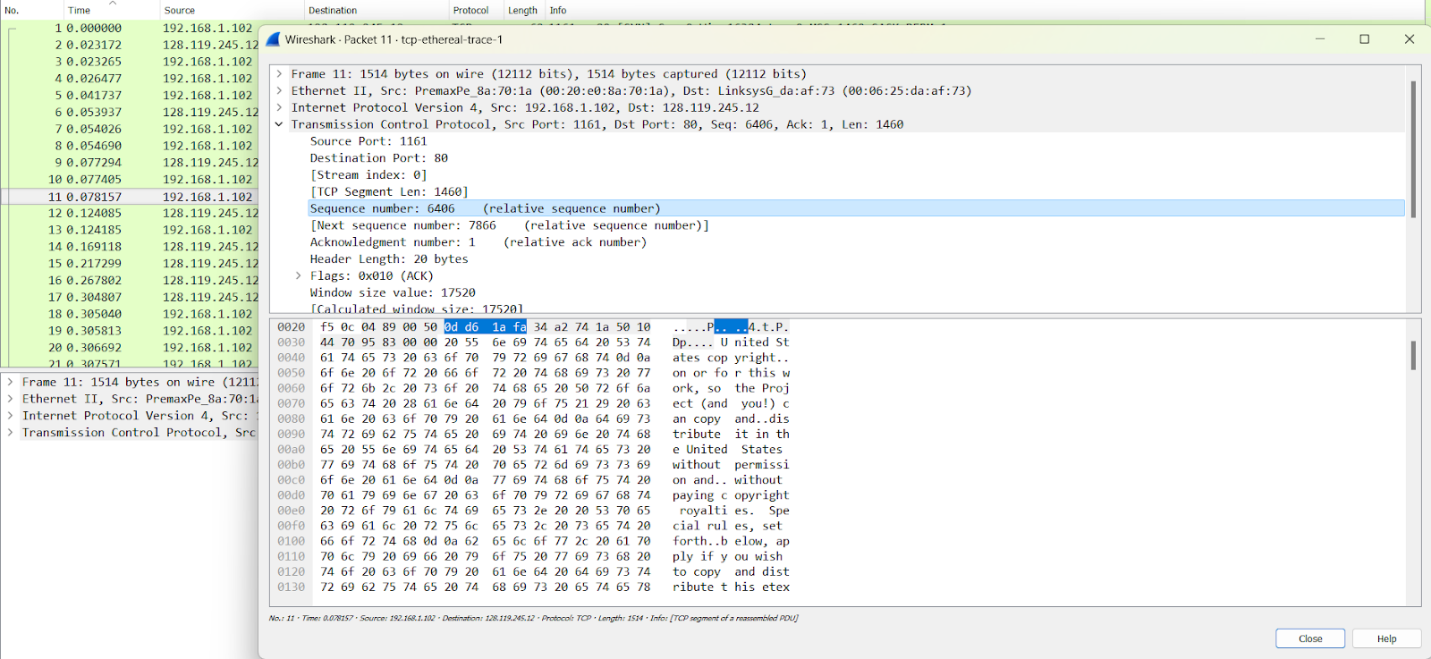




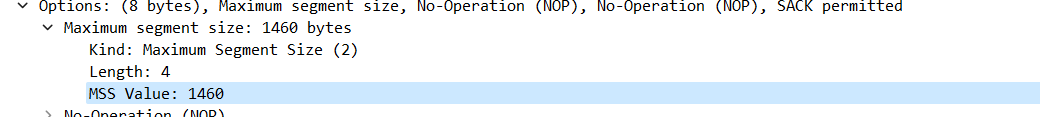


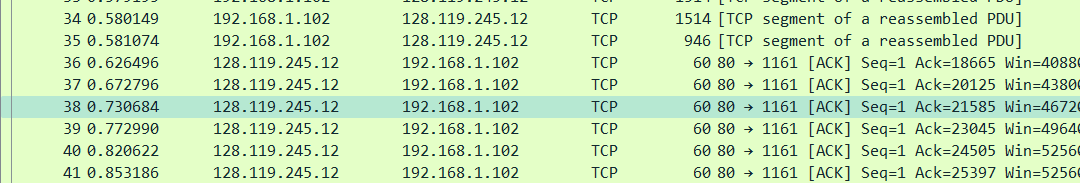






1. The minimum buffer space is 1460 bytes. That’s why each segment is sent in a maximum packet size of 1460 bytes. Also, specified in the SYN\_ACK segment sent by the server.



1. No, because there wasn’t any timeout or duplicate packet found in the trace. If there were any, It would be shown in WireShark.
2. Yes there are such cases as shown in the figure below.
3. Throughput : 1.4\*106 bits/s.

