TCP Client-Server Implementation and Study
Experiment No: AV-341-2025-Lab-8

Saurabh Kumar
SC22B146
April 7, 2025

Date and Time of experiment: April 15, 2025, 15:00 IST

Objectives

- Capture packet data for multiple TCP sessions using Wireshark.
- Identify
  - The Connection-set up packets and describe the information transferred as part of the connection setup.
  - The connection tear down related packets and their description.
  - Flags you observe and their relevance.
  - Estimate bandwidth vs time graph during the session.

Tools Used

- PC: 12th Gen Intel(R) Core(TM) i5-1240P 1.70 GHz, Windows 11, 64-bit, (Reduced to) 4 GB RAM
- Software used: Wireshark

Procedure

1. Open Wireshark and search for packets on Wi-Fi.
2. TCP Connection Setup (3-Way Handshake):
   - Identify a TCP connection (use flag: `tcp.flags`), where a client initiates a connection by sending a SYN packet to the server, indicating its initial sequence number.
   - The server responds with a SYN-ACK packet, acknowledging the client's sequence number and sending its own.
   - The client sends an ACK packet back to the server, confirming the server's sequence number.
   - The TCP connection is established at this point.
3. TCP Session Teardown (4-Way Termination):

- The client sends a FIN packet to indicate it has no more data to send
- The server replies with an ACK, acknowledging the client's FIN.
- The server then sends its own FIN packet to indicate it is ready to close the connection.
- The client responds with an ACK, completing the connection termination

4. To visualize various graphs, follow the steps:
   - TCP Graphs: Statistics tab →TCP Stream Graphs →Time-Sequence Graph (tcptrace)
   - Throughput Graph (IO Graph): Statistics tab →I/O Graphs
   - RTT: Statistics tab →TCP Stream Graphs →Round Trip Time

Observations

- TCP Connection Setup: During the TCP connection setup phase, a three-way handshake is performed to establish a reliable communication channel between the client and the server.

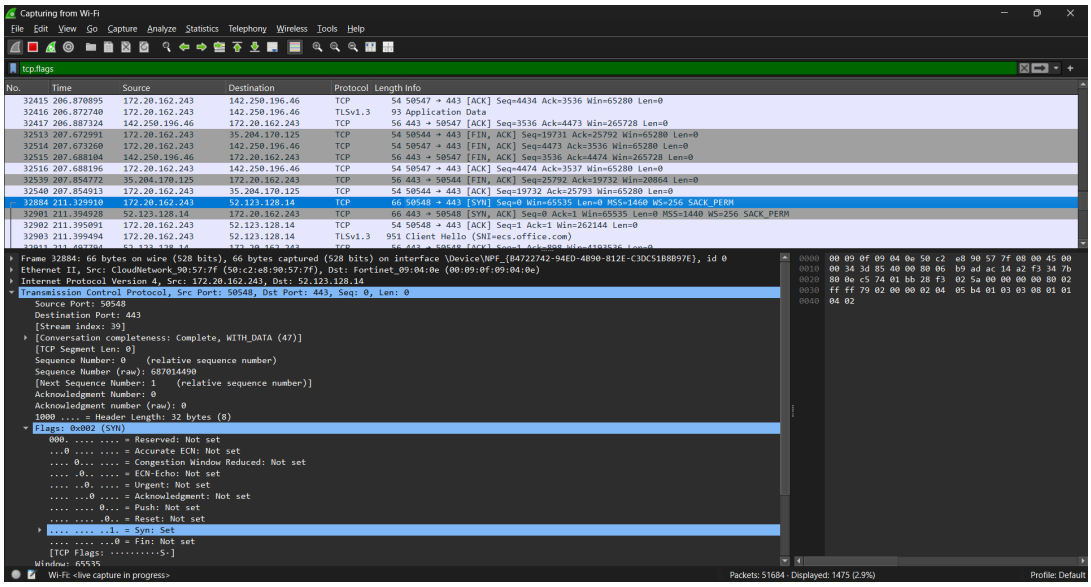

Figure 1: TCP connection packet showing sequence no., flags, etc. (SYN)

  - **SYN**: Sent by the client to initiate a connection. It includes the initial sequence number.
  - **SYN-ACK**: Sent by the server in response to the SYN, acknowledging the client's sequence number and sending its own.
  - **ACK**: Sent by the client to acknowledge the server's SYN-ACK, completing the handshake.

- Packet Transmission

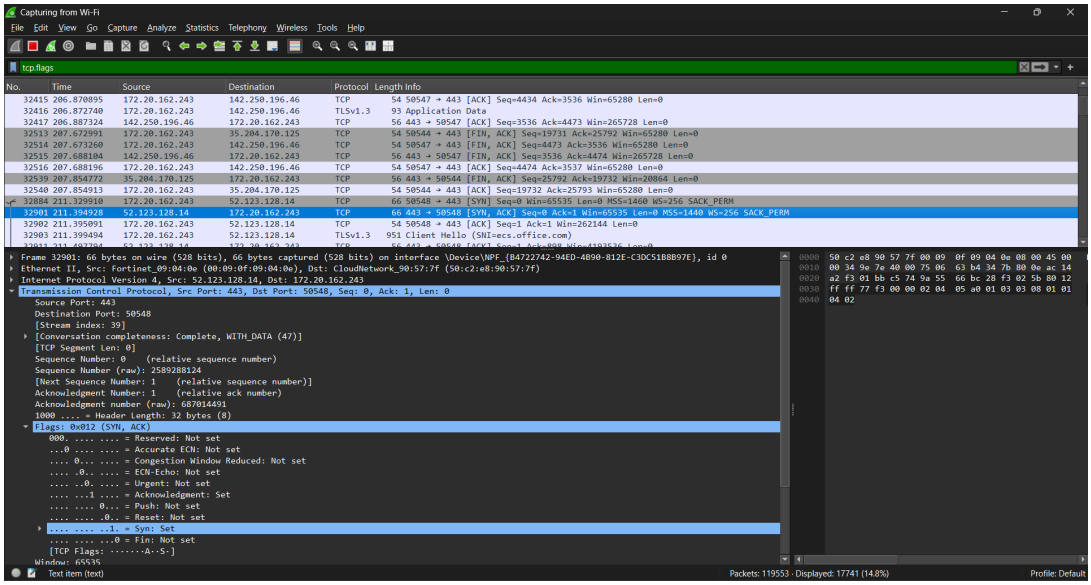

Figure 2: TCP connection packet showing sequence no., flags, etc. (SYN-ACK)

- Connection Teardown Packets
  - **FIN**: Sent by one end to indicate it has finished sending data.
  - **ACK**: Acknowledges the receipt of the FIN.
  - **FIN**: Sent by the other end to close its side of the connection.
  - **ACK**: Final acknowledgment to confirm closure.
- Bandwidth vs Time Graph: To visualize the bandwidth usage.
  - TCP Graphs (Time-Sequence Graph (tcptrace)): This shows data flow over time.

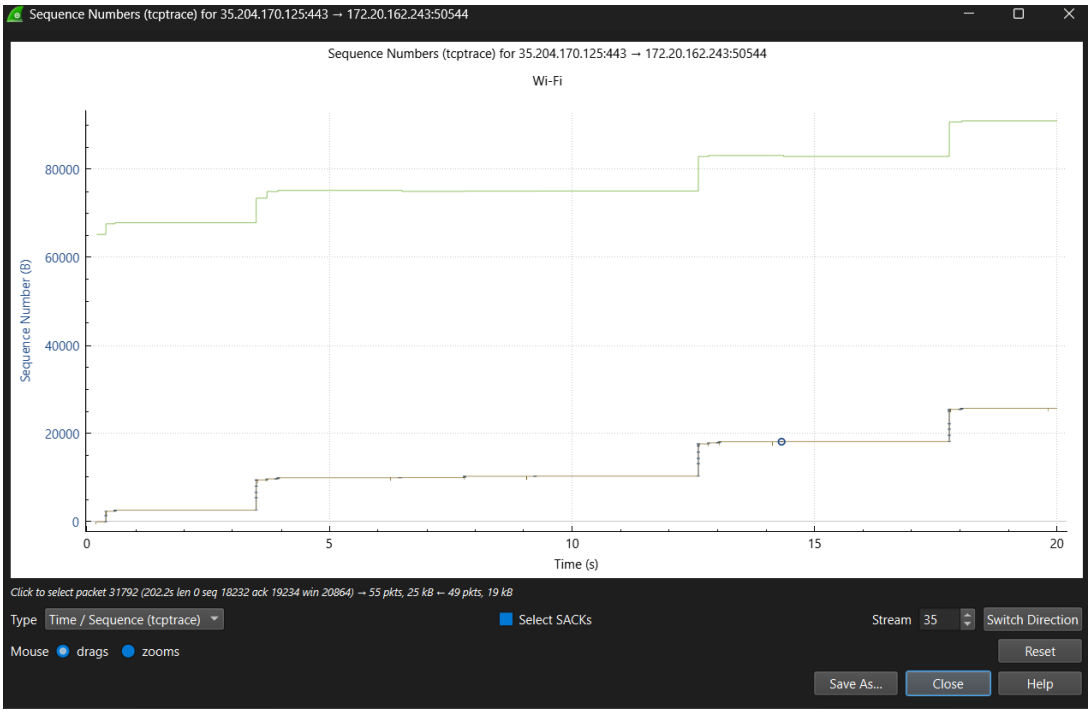

Figure 3: tcptrace graph showing data flow over time

– Throughput Graph (IO Graph): This graph will show bandwidth (bytes/sec) vs time.

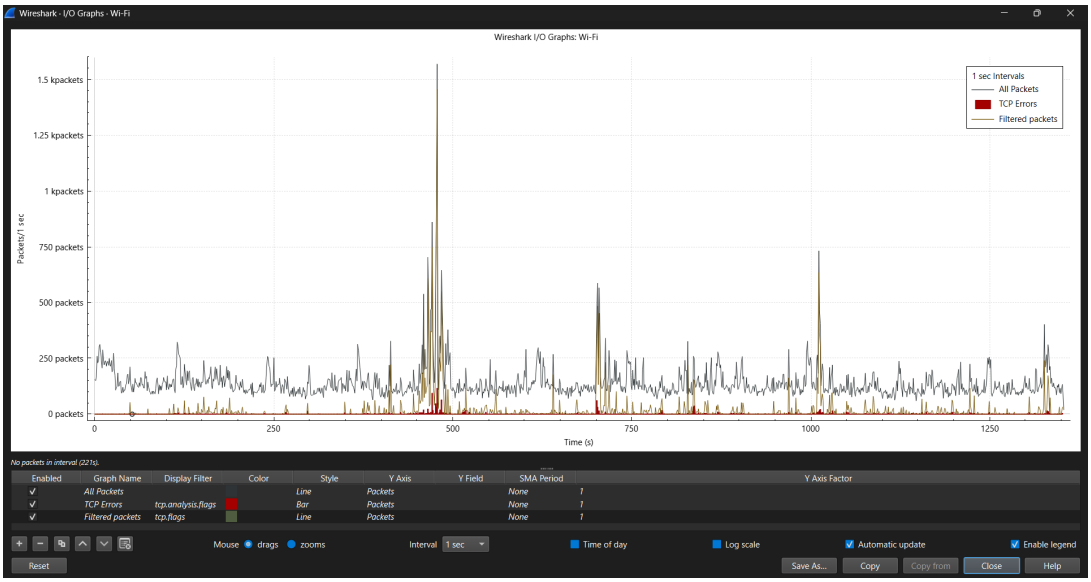

Figure 4: IO graph showing bandwidth vs time

– RTT: This shows Round Trip Time.

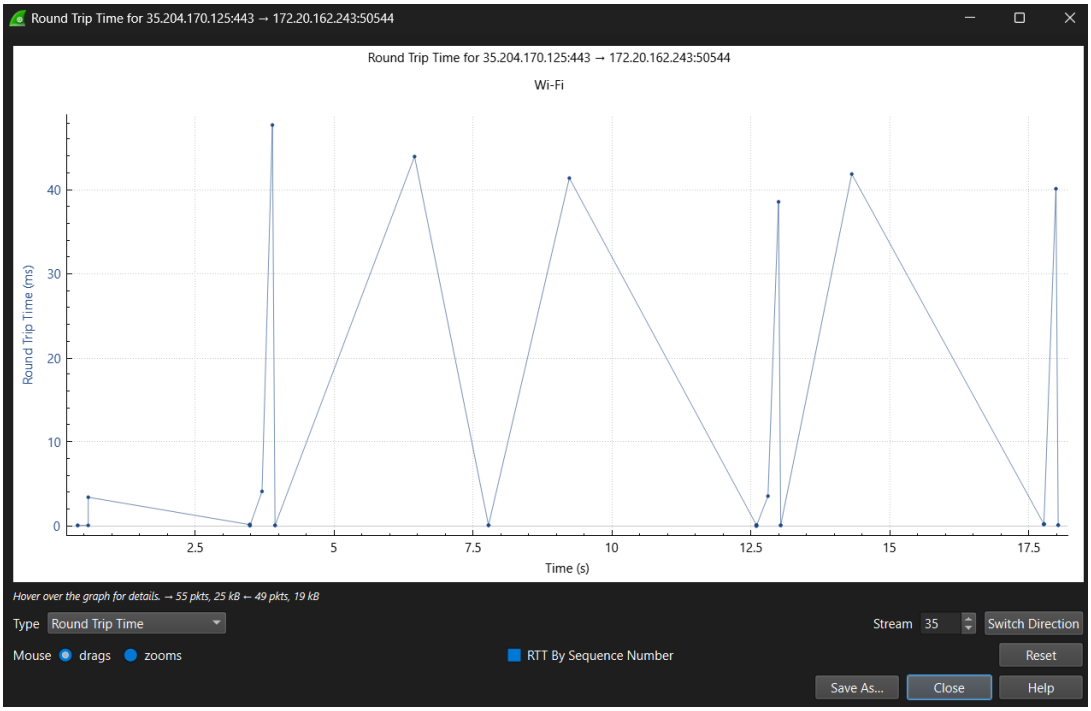

Figure 5: RTT graph

Conclusions

- This experiment provided practical insight into how TCP connections are established and terminated using a three-way handshake and a four-way termination. Using Wireshark, we captured and analyzed live TCP packet data, examined critical TCP flags, and visualized connection behavior through sequence graphs and throughput charts. These observations reinforced the theoretical concepts of reliable data transfer in TCP.