TCP Sockets

Name:Sumit Kumar
Roll No:-22CS30056
CS39006 Networks Laboratory
Assignment 3

Part-2: Wireshark Analysis

1.What are the source and destination IP addresses and ports? Share the screenshots to justify your answer.

While sending the file from client to server:

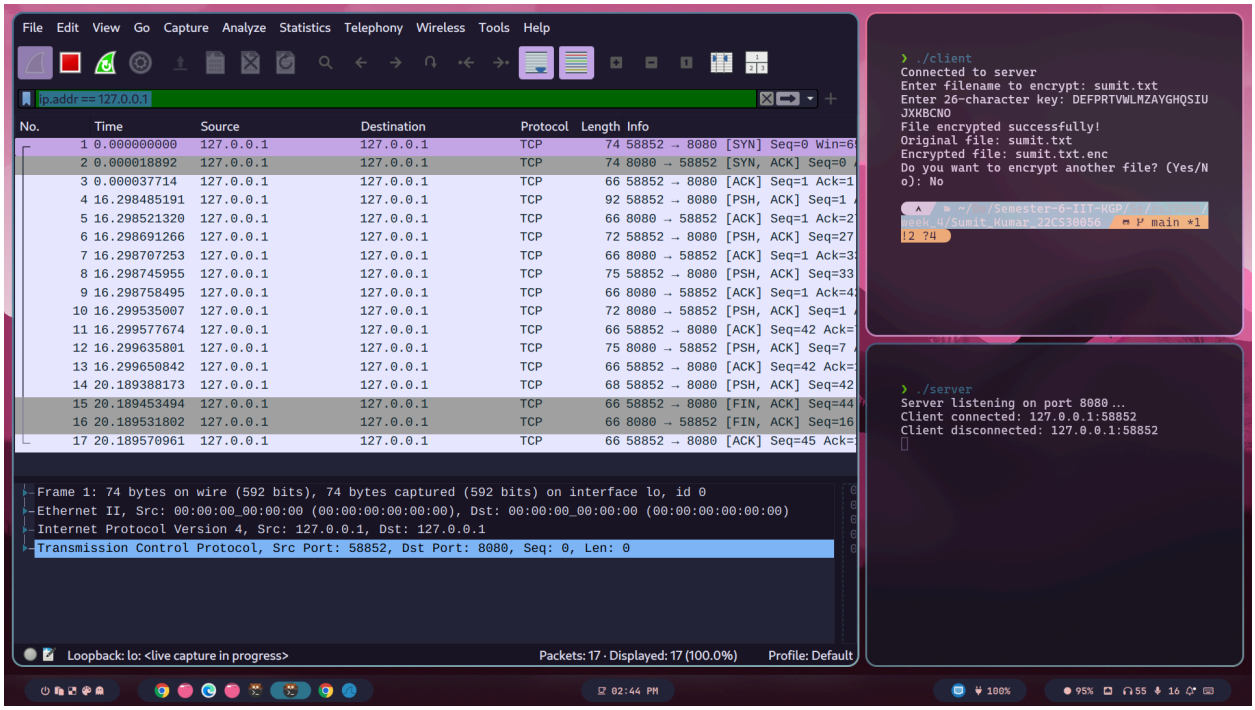
Source IP: 127.0.0.1
Source Port:58852

Destination Ip:127.0.0.1
Destination Port:8080

While sending the encryted file from server to client:

Source IP: 127.0.0.1
Source Port:8080

Destination Ip:127.0.0.1
Destination Port:58852



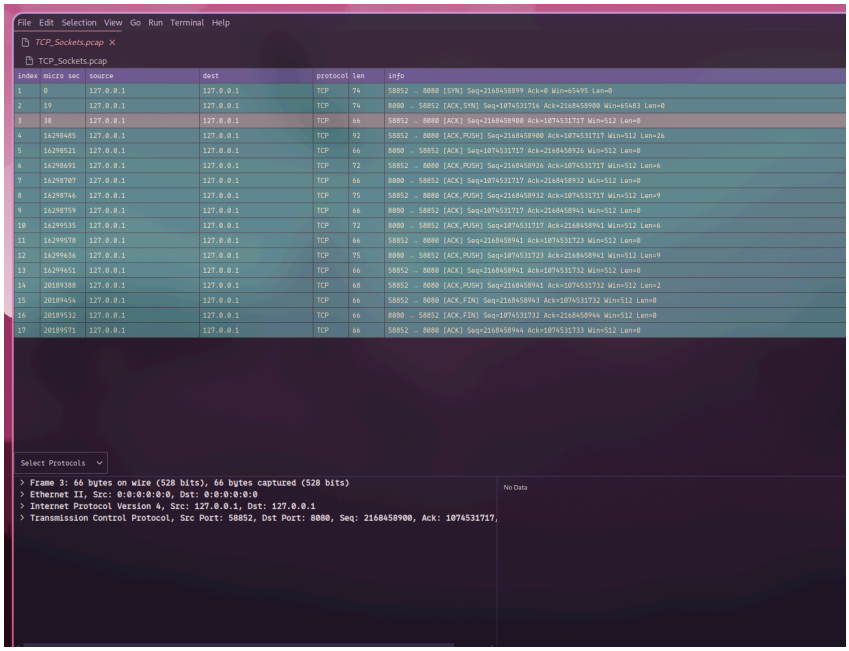
2. Inspect the Three-way handshaking procedure and capture all packets exchanged for it. Attach the necessary screenshots to demonstrate it.

From the image, we can inspect the TCP three-way handshake process, which consists of the following steps:

1. **SYN (Synchronization)**
    - The client (127.0.0.1) sends a **SYN** packet to the server (127.0.0.1) on port **8080**, indicating an attempt to establish a connection.
    - This is visible in the **first highlighted packet** in the Wireshark capture.
2. **SYN-ACK (Synchronization-Acknowledgment)**
    - The server responds with a **SYN-ACK** packet, acknowledging the client's request and indicating that it is ready to establish the connection.
    - This corresponds to the **second highlighted packet** in the capture.
3. **ACK (Acknowledgment)**
    - The client then sends an **ACK** packet to confirm the connection establishment.
    - This is the **third packet** in the exchange.

The Wireshark capture clearly displays this handshake process between **source port 58852** (client) and **destination port 8080** (server). The packets show sequence numbers, acknowledgments, and TCP flags.

The request indexes 1,2 and 3 denote this



3.Inspect the connection closure procedure and capture all packets exchanged for it. Attach the necessary screenshots to demonstrate it.

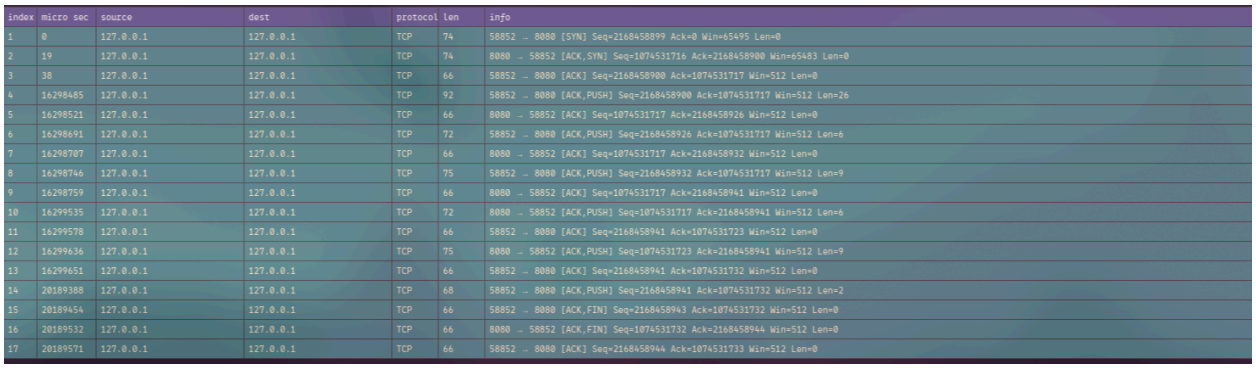
TCP Connection Closure (Four-Way Handshake)

TCP uses a **four-step** termination process to gracefully close a connection:

1. **FIN (Finish) from Client**
    - The client (127.0.0.1) sends a **FIN** packet to the server (127.0.0.1:8080) to indicate that it has finished sending data.
    - The server acknowledges this request.
2. **ACK (Acknowledgment) from Server**
    - The server sends an **ACK** packet to confirm it received the FIN request.
3. **FIN (Finish) from Server**

- The server sends its own **FIN** packet to indicate it is also done sending data.

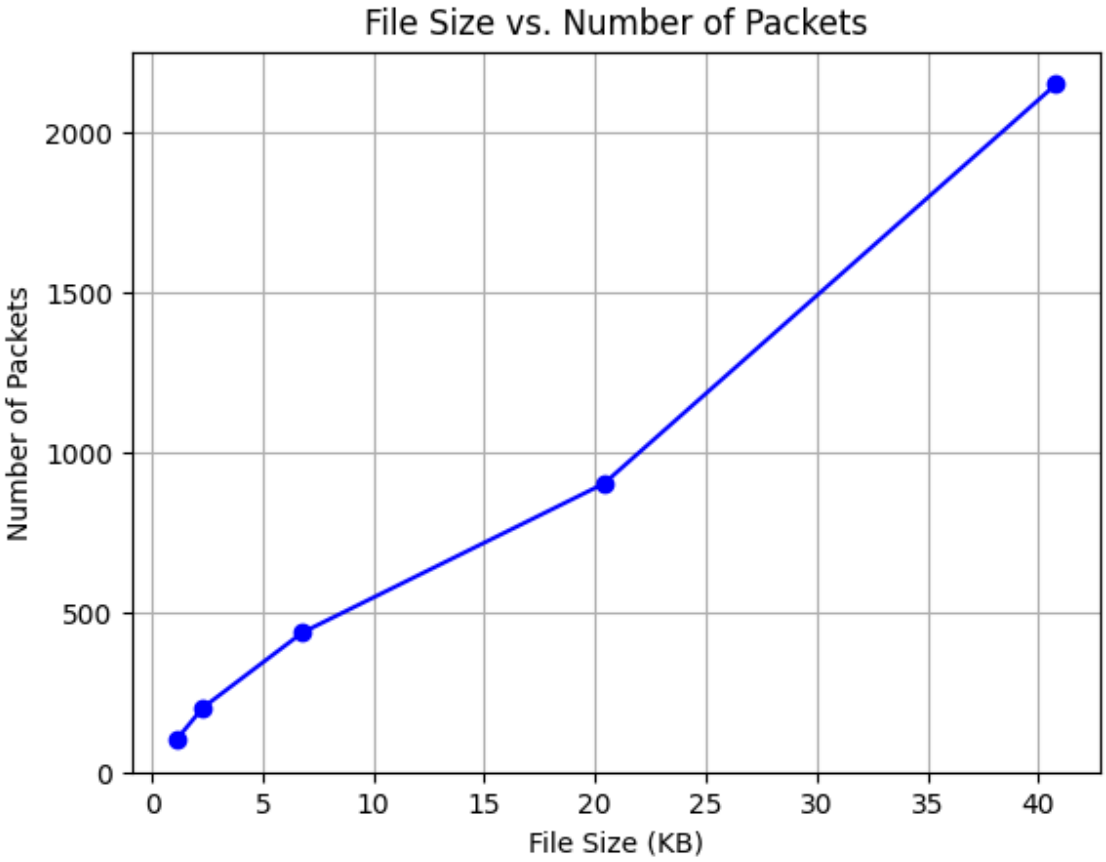
4. **ACK (Final Acknowledgment) from Client**
   - The client responds with an **ACK**, confirming the server's FIN, and the connection is fully closed.



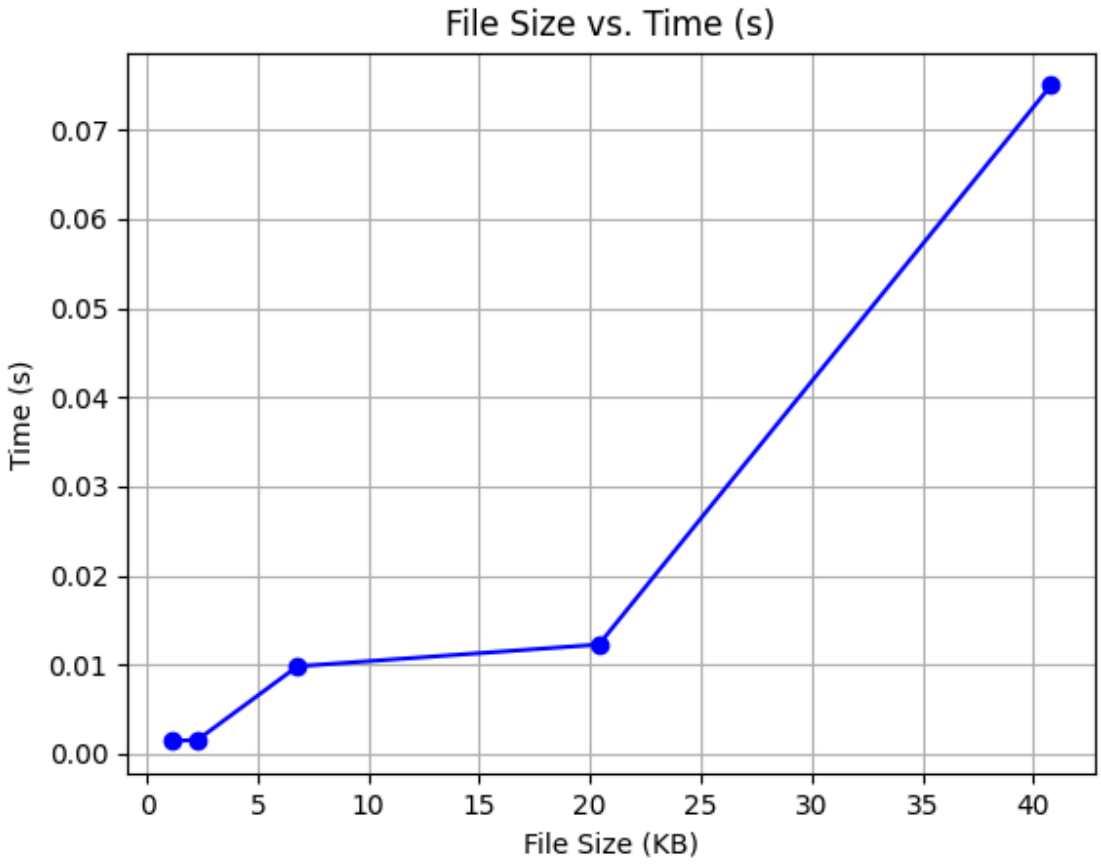
index	micro sec	source	dest	protocol	len	info
1	0	127.0.0.1	127.0.0.1	TCP	74	58852 → 8080 [SYN] Seq=2168458899 Ack=0 Win=65495 Len=0
2	19	127.0.0.1	127.0.0.1	TCP	74	8080 → 58852 [ACK,SYN] Seq=1074531716 Ack=2168458900 Win=65483 Len=0
3	38	127.0.0.1	127.0.0.1	TCP	66	58852 → 8080 [ACK] Seq=2168458900 Ack=1074531717 Win=512 Len=0
4	16298485	127.0.0.1	127.0.0.1	TCP	92	58852 → 8080 [ACK,PUSH] Seq=2168458900 Ack=1074531717 Win=512 Len=26
5	16298521	127.0.0.1	127.0.0.1	TCP	66	8080 → 58852 [ACK] Seq=1074531717 Ack=2168458926 Win=512 Len=0
6	16298691	127.0.0.1	127.0.0.1	TCP	72	58852 → 8080 [ACK,PUSH] Seq=2168458926 Ack=1074531717 Win=512 Len=6
7	16298707	127.0.0.1	127.0.0.1	TCP	66	8080 → 58852 [ACK] Seq=1074531717 Ack=2168458932 Win=512 Len=0
8	16298746	127.0.0.1	127.0.0.1	TCP	75	58852 → 8080 [ACK,PUSH] Seq=2168458932 Ack=1074531717 Win=512 Len=9
9	16298759	127.0.0.1	127.0.0.1	TCP	66	8080 → 58852 [ACK] Seq=1074531717 Ack=2168458941 Win=512 Len=0
10	16299535	127.0.0.1	127.0.0.1	TCP	72	8080 → 58852 [ACK,PUSH] Seq=1074531717 Ack=2168458941 Win=512 Len=6
11	16299578	127.0.0.1	127.0.0.1	TCP	66	58852 → 8080 [ACK] Seq=2168458941 Ack=1074531723 Win=512 Len=0
12	16299636	127.0.0.1	127.0.0.1	TCP	75	8080 → 58852 [ACK,PUSH] Seq=1074531723 Ack=2168458941 Win=512 Len=9
13	16299651	127.0.0.1	127.0.0.1	TCP	66	58852 → 8080 [ACK] Seq=2168458941 Ack=1074531732 Win=512 Len=0
14	20189388	127.0.0.1	127.0.0.1	TCP	68	58852 → 8080 [ACK,PUSH] Seq=2168458941 Ack=1074531732 Win=512 Len=2
15	20189454	127.0.0.1	127.0.0.1	TCP	66	58852 → 8080 [ACK,FIN] Seq=2168458943 Ack=1074531732 Win=512 Len=0
16	20189532	127.0.0.1	127.0.0.1	TCP	66	8080 → 58852 [ACK,FIN] Seq=1074531732 Ack=2168458944 Win=512 Len=0
17	20189571	127.0.0.1	127.0.0.1	TCP	66	58852 → 8080 [ACK] Seq=2168458944 Ack=1074531733 Win=512 Len=0

The **FIN, ACK, FIN, and ACK** sequence present in the request index 15 and 16.

4. Inspect the traffics and count the number of packets exchanged for the transfer of a file(related to data only) between client and server. Plot a graph ‘file size vs the number of packets’ clearly based on your observation


File Size vs. Number of Packets
Number of Packets
File Size (KB)
0
500
1000
1500
2000
0
5
10
15
20
25
30
35
40

5. Measure the total time taken for the file transfer , its encryption and send back it from server to the client. Plot a graph 'file size vs time' clearly based on your observation and also attach the necessary screenshots.



6.Calculate the average size packet exchanged during the data communication? Take reference of the plotted graph in the above question.

1. Looking at some key points from Image 1 (File Size vs. Number of Packets):
   - At 40 KB ≈ 2100 packets
   - At 20 KB ≈ 900 packets
   - At 7 KB ≈ 400 packets
   - At 2 KB ≈ 150 packets

2. The formula for average packet size would be:
   Average Packet Size = File Size / Number of Packets

3. Let's calculate using the 40 KB point for most accuracy:

- File Size = 40 KB = 40,960 bytes
- Number of Packets ≈ 2100

4. Calculation:
   40,960 bytes / 2100 packets = 19.5 bytes per packet

Therefore, the average packet size during this data communication is approximately 19.5 bytes per packet.