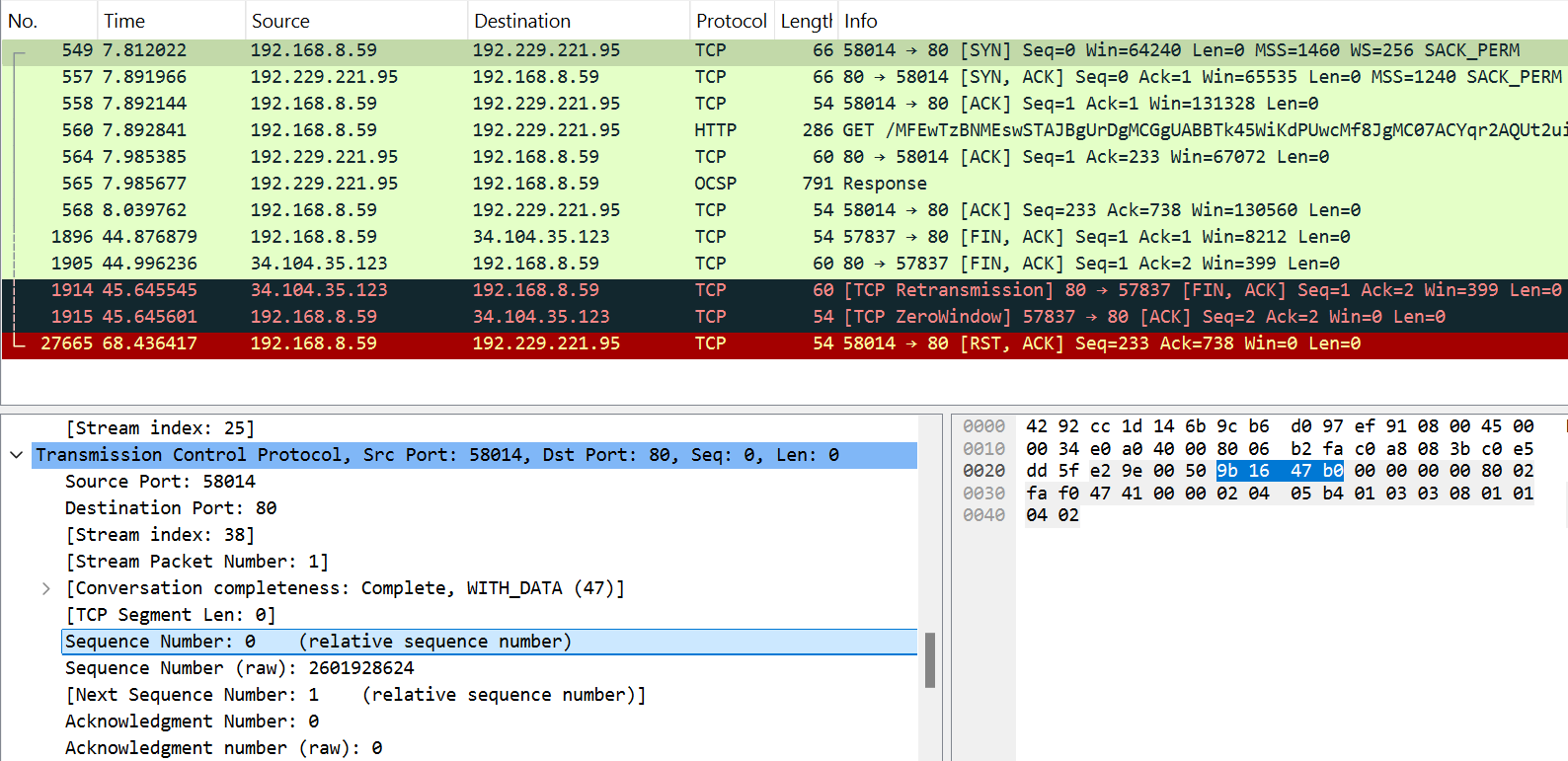
**Part2\_Task2\_Step2**

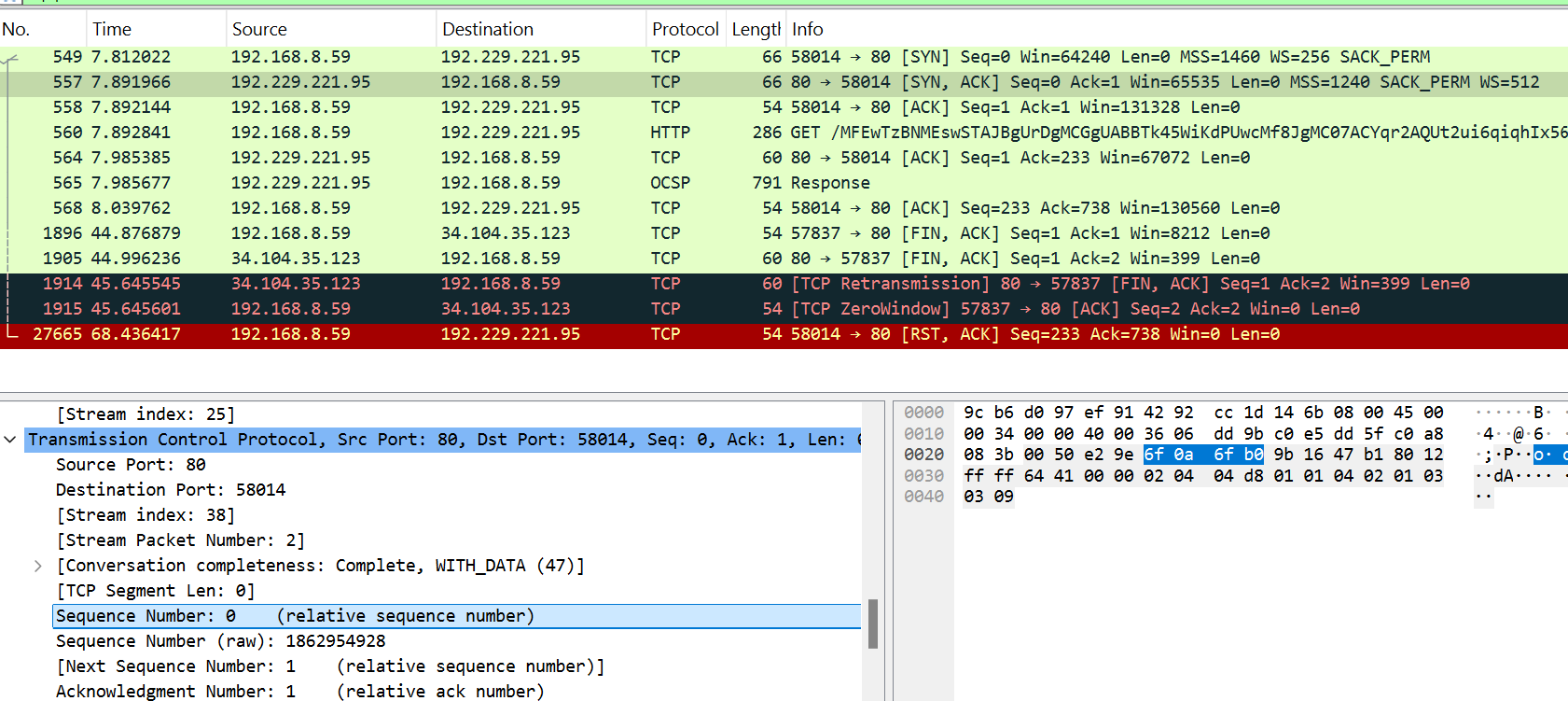
**Step 2: Sequence and Acknowledgment Numbers**

As we can see:

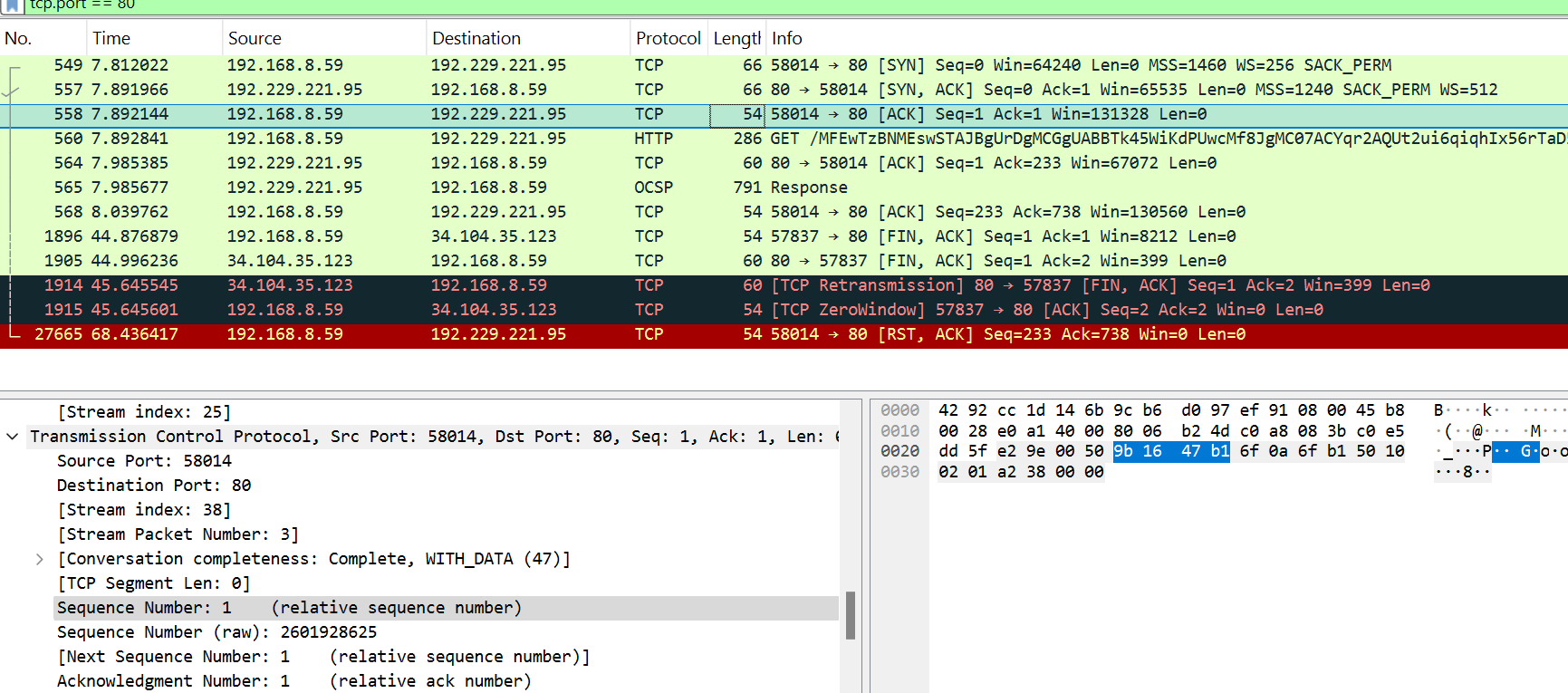
In the first packet, the sequence number is **`0`** and the acknowledgment number is **`0`.**



In the second packet, the sequence number is **`0`** and the acknowledgment number is **`1`.**



In the third packet, the sequence number is **`1`** and the acknowledgment number is **`1`.**



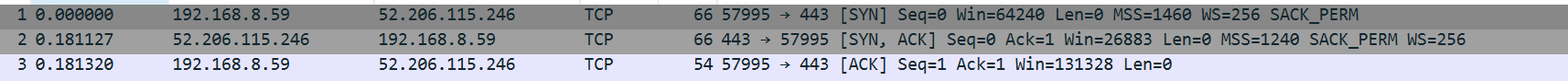
**Part2\_Task2\_Step3**

**TCP Three-Way Handshake**

1. **SYN (Synchronization):**
   * The client (192.168.8.59) initiates the connection by sending a SYN packet to the server (52.206.115.246). This packet requests synchronization and starts the process of establishing a connection.
2. **SYN-ACK (Synchronization-Acknowledgment):**
   * The server responds to the client with a SYN-ACK packet. This packet acknowledges the client's SYN request and sends its own SYN request to the client.
3. **ACK (Acknowledgment):**
   * The client sends an ACK packet back to the server, acknowledging the server's SYN-ACK packet. This completes the three-way handshake, and a reliable TCP connection is established.

**Summary**

This three-way handshake is the initial exchange of data packets required to establish a TCP connection. It ensures that both the client and server are ready to communicate, and it sets up the parameters for the data transfer that will follow.



**TCP VS UDP**

**Part4:**

|  |  |  |
| --- | --- | --- |
| **Reasons** | **TCP or UDP** |  |
| TCP establishes a reliable connection using a three-way handshake (SYN, SYN-ACK, ACK) and ensures data delivery with acknowledgments and retransmissions. | **TCP** | Reliability and Connection Establishment |
| TCP guarantees data integrity and ordering through sequence numbers and acknowledgments, ensuring correct and complete data delivery. | **TCP** | Data Integrity and Ordering |

|  |  |  |
| --- | --- | --- |
| **UDP** | **TCP** |  |
| is ideal for applications where speed and low latency are more important than reliability. This includes real-time communications, streaming, online gaming, and DNS lookups. | is suitable for applications where reliable data transmission is crucial and where data integrity and ordering must be guaranteed. These include web browsing, email, file transfers, and secure remote access | **Use Cases** |
| offers lower latency and faster data transfer due to its connectionless nature and minimal overhead, but it lacks guarantees for data delivery and ordering, which may lead to packet loss or duplication. | provides high reliability with guaranteed delivery and ordering of packets, but it comes with increased overhead and potential latency due to its connection-oriented nature. | **Performance** |