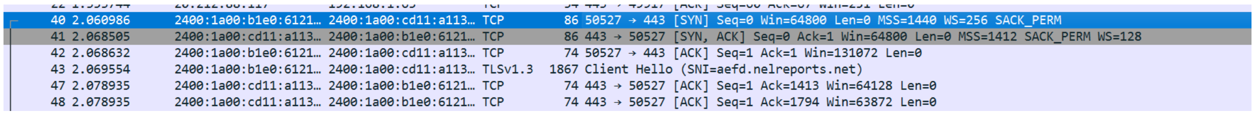
Connection initialization,Connection mainatainenance and Connection termination

TCP (Transmission Control Protocol) ensures reliable, ordered, and error-checked delivery of data between applications running on hosts communicating over an IP network. here is a clear breakdown of the three main phases of a TCP connection: initialization, maintenance, and termination.

1. Connection Initialization

Connection initialization is the process of establishing a connection between two devices or systems. It involves three steps to establish a connection between a client and a server:

- **SYN (Synchronize)**: The client initiates the connection by sending a TCP segment with the SYN flag set. This segment includes an initial sequence number (ISN) that the client will use to start counting bytes in the data it sends.
- **SYN-ACK (Synchronize-Acknowledge)**: The server responds with a segment that has both the SYN and ACK flags set. This segment acknowledges the client’s SYN segment (by including an acknowledgment number) and includes the server’s own initial sequence number.
- **ACK (Acknowledge)**: The client sends a final segment with the ACK flag set to acknowledge the server's SYN-ACK segment. At this point, the connection is established, and data can begin to be exchanged.

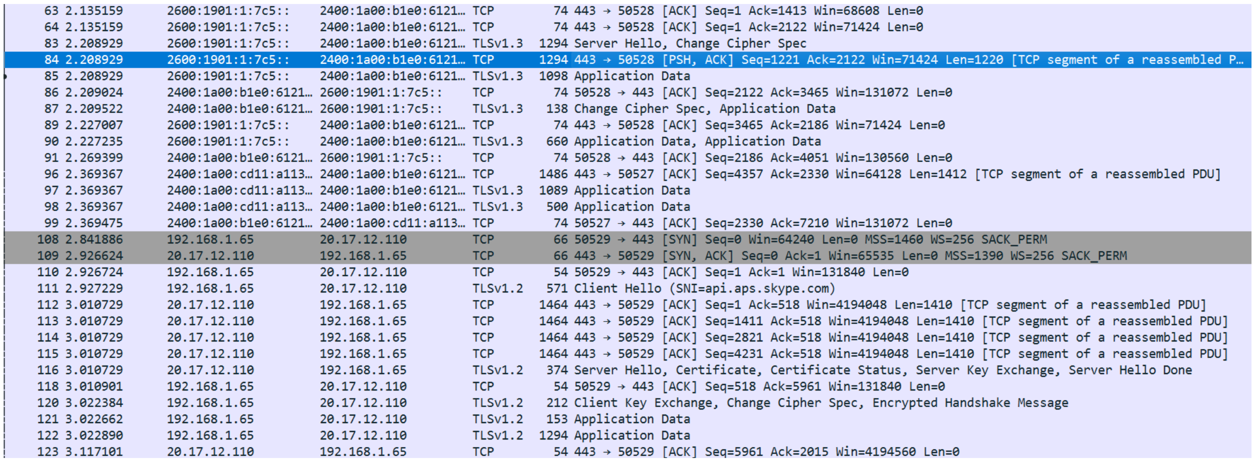


```
40 2.060986   2400:1a00:b1e0:6121… 2400:1a00:cd11:a113… TCP       86 50527 → 443 [SYN] Seq=0 Win=64800 Len=0 MSS=1440 WS=256 SACK_PERM
41 2.068505   2400:1a00:cd11:a113… 2400:1a00:b1e0:6121… TCP       86 443 → 50527 [SYN, ACK] Seq=0 Ack=1 Win=64800 Len=0 MSS=1412 SACK_PERM WS=128
42 2.068632   2400:1a00:b1e0:6121… 2400:1a00:cd11:a113… TCP       74 50527 → 443 [ACK] Seq=1 Ack=1 Win=131072 Len=0
43 2.069554   2400:1a00:b1e0:6121… 2400:1a00:cd11:a113… TLSv1.3 1867 Client Hello (SNI=aefd.nelreports.net)
47 2.078935   2400:1a00:cd11:a113… 2400:1a00:b1e0:6121… TCP       74 443 → 50527 [ACK] Seq=1 Ack=1413 Win=64128 Len=0
48 2.078935   2400:1a00:cd11:a113… 2400:1a00:b1e0:6121… TCP       74 443 → 50527 [ACK] Seq=1 Ack=1794 Win=63872 Len=0
```

2. Connection Maintenance

Once the connection is established, it needs to be maintained to ensure reliable communication. This involves:

- **Data Transmission**: Data is sent in segments, each with a sequence number. The receiver acknowledges receipt of these segments with ACK segments. This ensures that data is received correctly and in the right order.
- **Flow Control**: TCP uses a flow control mechanism (via the window size) to ensure that the sender does not overwhelm the receiver with too much data too quickly.
- **Error Checking**: Each segment includes a checksum to verify the integrity of the data. If a segment is lost or corrupted, it is retransmitted.
- **Congestion Control**: TCP adjusts the rate of data transmission based on network congestion to avoid overloading the network. This is done using algorithms like slow start, congestion avoidance, and fast recovery.



```
 63 2.135159   2600:1901:1:7c5::    2400:1a00:b1e0:6121… TCP       74 443 → 50528 [ACK] Seq=1 Ack=1413 Win=68608 Len=0
 64 2.135159   2600:1901:1:7c5::    2400:1a00:b1e0:6121… TCP       74 443 → 50528 [ACK] Seq=1 Ack=2122 Win=71424 Len=0
 83 2.208929   2600:1901:1:7c5::    2400:1a00:b1e0:6121… TLSv1.3 1294 Server Hello, Change Cipher Spec
 84 2.208929   2600:1901:1:7c5::    2400:1a00:b1e0:6121… TCP     1294 443 → 50528 [PSH, ACK] Seq=1221 Ack=2122 Win=71424 Len=1220 [TCP segment of a reassembled P…
 85 2.208929   2600:1901:1:7c5::    2400:1a00:b1e0:6121… TLSv1.3 1098 Application Data
 86 2.209024   2400:1a00:b1e0:6121… 2600:1901:1:7c5::    TCP       74 50528 → 443 [ACK] Seq=2122 Ack=3465 Win=131072 Len=0
 87 2.209522   2400:1a00:b1e0:6121… 2600:1901:1:7c5::    TLSv1.3  138 Change Cipher Spec, Application Data
 89 2.227007   2600:1901:1:7c5::    2400:1a00:b1e0:6121… TCP       74 443 → 50528 [ACK] Seq=3465 Ack=2186 Win=71424 Len=0
 90 2.227235   2600:1901:1:7c5::    2400:1a00:b1e0:6121… TLSv1.3  660 Application Data, Application Data
 91 2.269399   2400:1a00:b1e0:6121… 2600:1901:1:7c5::    TCP       74 50528 → 443 [ACK] Seq=2186 Ack=4051 Win=130560 Len=0
 96 2.369367   2400:1a00:cd11:a113… 2400:1a00:b1e0:6121… TCP     1486 443 → 50527 [ACK] Seq=4357 Ack=2330 Win=64128 Len=1412 [TCP segment of a reassembled PDU]
 97 2.369367   2400:1a00:cd11:a113… 2400:1a00:b1e0:6121… TLSv1.3 1089 Application Data
 98 2.369367   2400:1a00:cd11:a113… 2400:1a00:b1e0:6121… TLSv1.3  500 Application Data
 99 2.369475   2400:1a00:b1e0:6121… 2400:1a00:cd11:a113… TCP       74 50527 → 443 [ACK] Seq=2330 Ack=7210 Win=131072 Len=0
108 2.841886   192.168.1.65         20.17.12.110         TCP       66 50529 → 443 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 WS=256 SACK_PERM
109 2.926624   20.17.12.110         192.168.1.65         TCP       66 443 → 50529 [SYN, ACK] Seq=0 Ack=1 Win=65535 Len=0 MSS=1390 WS=256 SACK_PERM
110 2.926724   192.168.1.65         20.17.12.110         TCP       54 50529 → 443 [ACK] Seq=1 Ack=1 Win=131840 Len=0
111 2.927229   192.168.1.65         20.17.12.110         TLSv1.2  571 Client Hello (SNI=api.aps.skype.com)
112 3.010729   20.17.12.110         192.168.1.65         TCP     1464 443 → 50529 [ACK] Seq=1 Ack=518 Win=4194048 Len=1410 [TCP segment of a reassembled PDU]
113 3.010729   20.17.12.110         192.168.1.65         TCP     1464 443 → 50529 [ACK] Seq=1411 Ack=518 Win=4194048 Len=1410 [TCP segment of a reassembled PDU]
114 3.010729   20.17.12.110         192.168.1.65         TCP     1464 443 → 50529 [ACK] Seq=2821 Ack=518 Win=4194048 Len=1410 [TCP segment of a reassembled PDU]
115 3.010729   20.17.12.110         192.168.1.65         TCP     1464 443 → 50529 [ACK] Seq=4231 Ack=518 Win=4194048 Len=1410 [TCP segment of a reassembled PDU]
116 3.010729   20.17.12.110         192.168.1.65         TLSv1.2  374 Server Hello, Certificate, Certificate Status, Server Key Exchange, Server Hello Done
118 3.010901   192.168.1.65         20.17.12.110         TCP       54 50529 → 443 [ACK] Seq=518 Ack=5961 Win=131840 Len=0
120 3.022384   192.168.1.65         20.17.12.110         TLSv1.2  212 Client Key Exchange, Change Cipher Spec, Encrypted Handshake Message
121 3.022662   192.168.1.65         20.17.12.110         TLSv1.2  153 Application Data
122 3.022890   192.168.1.65         20.17.12.110         TLSv1.2 1294 Application Data
123 3.117101   20.17.12.110         192.168.1.65         TCP       54 443 → 50529 [ACK] Seq=5961 Ack=2015 Win=4194560 Len=0
```

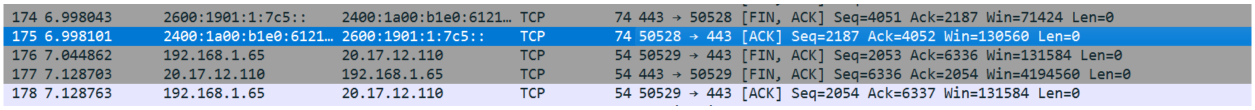
3. Connection Termination

Connection termination is the process of ending an established connection. This can be done gracefully or abruptly. When the communication is done, the connection is terminated gracefully to ensure all data is transmitted and acknowledged.

This process involves:

- **FIN (Finish)**: One side (either client or server) sends a segment with the FIN flag set to indicate it has finished sending data. This starts the connection termination process.

- **ACK**: The receiving side acknowledges the FIN segment with an ACK segment.
- **FIN from Receiver**: The receiver, once it has finished sending all its data, sends its own FIN segment to indicate it’s also done.
- **Final ACK**: The original sender acknowledges the receiver’s FIN segment with a final ACK segment. At this point, the connection is fully closed.
-



```
174 6.998043    2600:1901:1:7c5::    2400:1a00:b1e0:6121… TCP    74 443 → 50528 [FIN, ACK] Seq=4051 Ack=2187 Win=71424 Len=0
175 6.998101    2400:1a00:b1e0:6121… 2600:1901:1:7c5::    TCP    74 50528 → 443 [ACK] Seq=2187 Ack=4052 Win=130560 Len=0
176 7.044862    192.168.1.65         20.17.12.110         TCP    54 50529 → 443 [FIN, ACK] Seq=2053 Ack=6336 Win=131584 Len=0
177 7.128703    20.17.12.110         192.168.1.65         TCP    54 443 → 50529 [FIN, ACK] Seq=6336 Ack=2054 Win=4194560 Len=0
178 7.128763    192.168.1.65         20.17.12.110         TCP    54 50529 → 443 [ACK] Seq=2054 Ack=6337 Win=131584 Len=0
```

Each side of the connection goes through a **four-way handshake** to close the connection, ensuring that all data has been transmitted and acknowledged before the connection is terminated.