

This sequence diagram was generated with EventStudio System Designer (<http://www.EventHelix.com/EventStudio>).

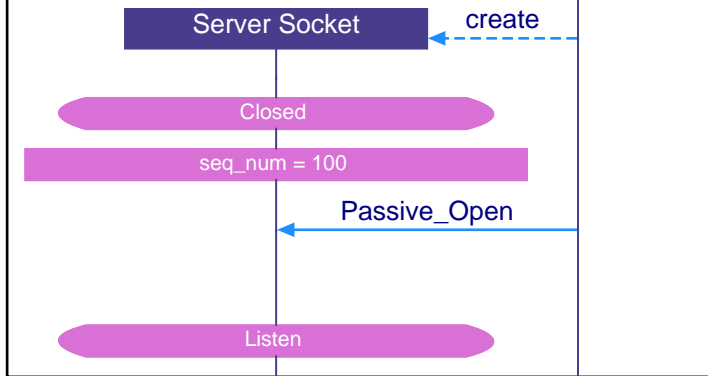
TCP (Transmission Control Protocol) provides a reliable end to end service that delivers packets over the Internet. Packets are delivered in sequence without loss or duplication.

This sequence diagram explores following: (1) The three-way handshake to establish a TCP (2) Data transfer using the byte oriented sequence numbers (3) Release of a TCP connection.

The TCP socket creation and deletion on the server and client is also covered.

Socket initialization

Server socket initialization



Server Application creates a Socket

The Socket is created in Closed state

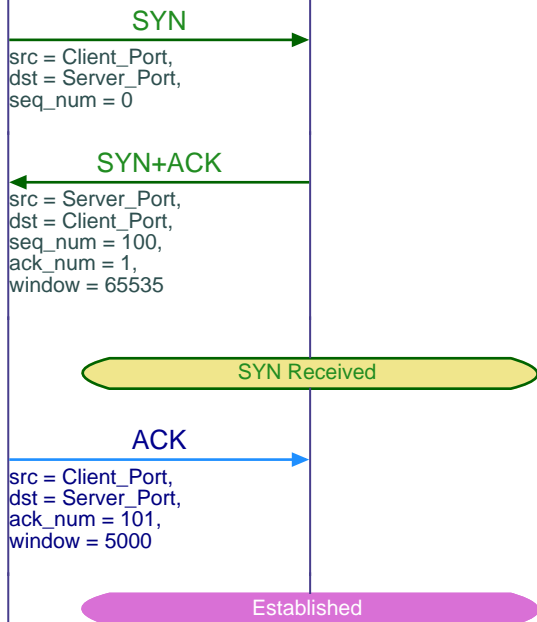
Server sets the initial sequence number to 100

Server application has initiated a passive open. In this mode, the socket does not attempt to establish a TCP connection. The socket listens for TCP connection request from clients

Socket transitions to the Listen state

Server awaits client socket connections.

Client initiated three way handshake to establish a TCP connection



SYN TCP segment is received by the server

Server sets the SYN and the ACK bits in the TCP header. Server sends its initial sequence number as 100. Server also sets its window to 65535 bytes. i.e. Server has buffer space for 65535 bytes of data. Also note that the ack sequence number is set to 1. This signifies that the server expects a next byte sequence number of 1

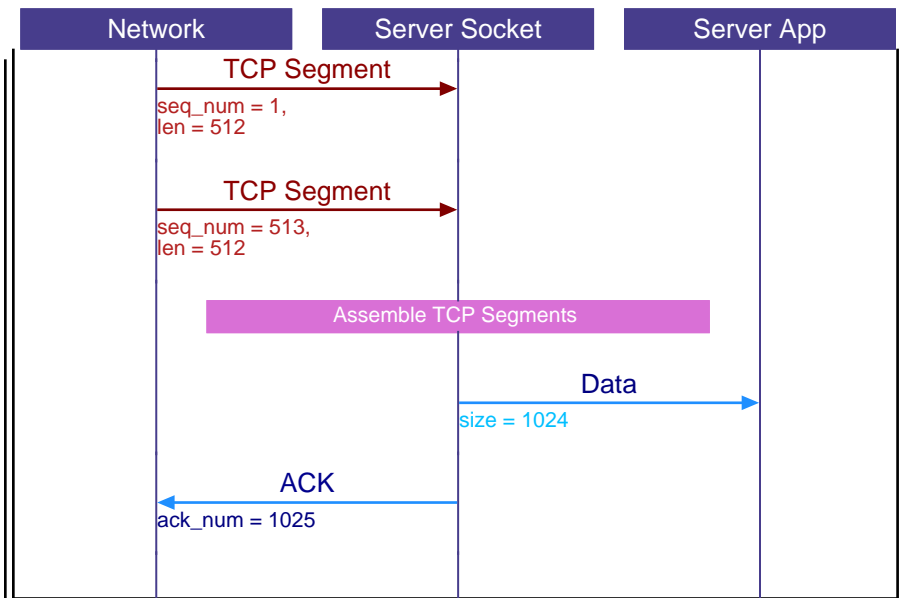
Now the server transitions to the SYN Received state

Server receives the TCP ACK segment

Now the server too moves to the Established state

Data transfer phase: Here a short data transfer takes place, thus TCP slow start has little impact

Client to server data transfer

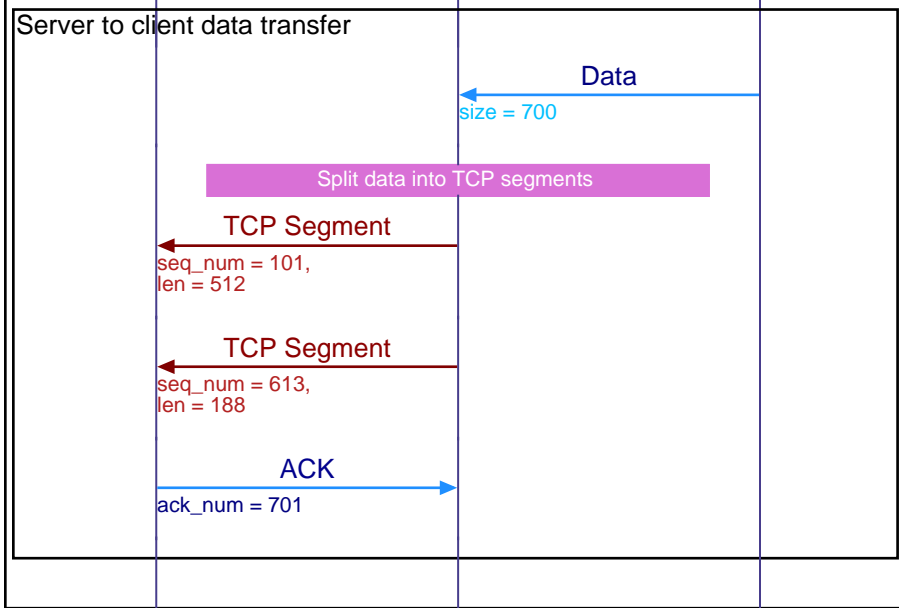


Server receives both the segments

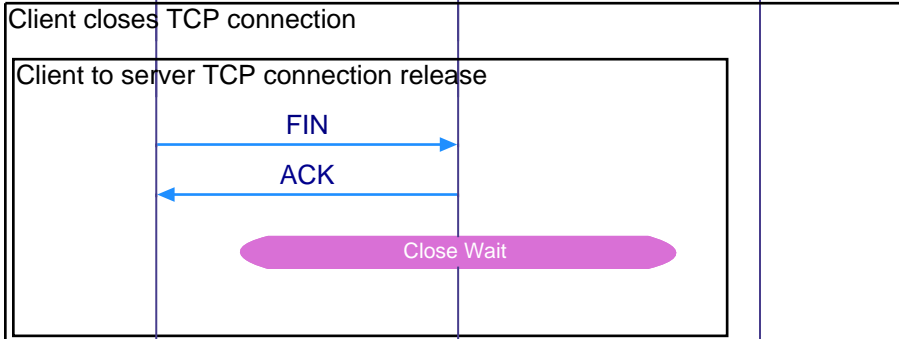
Server receives two consecutive segments, thus it assembles the segments

Assembled Data is passed to the Server Application

Server acknowledges the data segments with the next expected sequence number as 1025 (TCP typically sends an acknowledgement every two received segments)



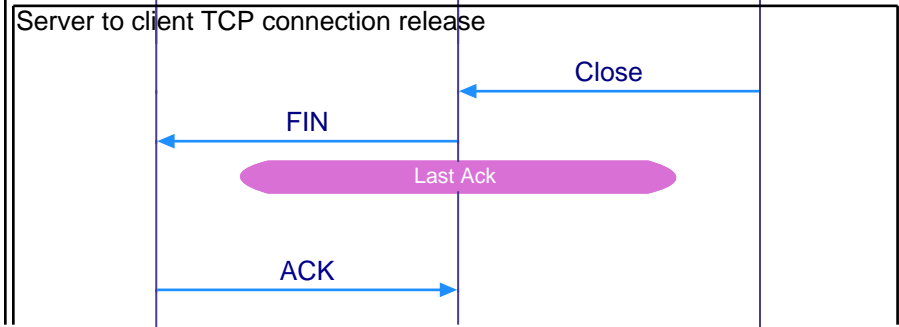
Now server responds back with data for the client



Server receives the FIN

Server responds back with ACK to acknowledge the FIN

Server changes state to Close Wait. In this state the server waits for the server application to close the connection

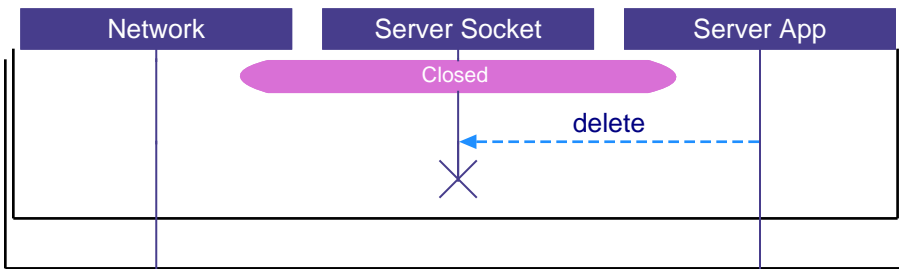


Server application closes the TCP connection

FIN is sent out to the client to close the connection

Server changes state to Last Ack. In this state the last acknowledgement from the client will be received

Server receives the ACK



Server moves the connection to closed state

This sequence diagram was generated with EventStudio System Designer (<http://www.EventHelix.com/EventStudio>).