TCP Connection Establishment Process: The "Three-Way Handshake"

(Page 3 of 4)

Normal Connection Establishment: The "Three Way Handshake"

To establish a connection, each device must send a *SYN* and receive an *ACK* for it from the other device. Thus, conceptually, we need to have four control messages pass between the devices. However, it's inefficient to send a *SYN* and an *ACK* in separate messages when one could communicate both simultaneously. Thus, in the normal sequence of events in connection establishment, one of the *SYNs* and one of the *ACKs* is sent together by setting both of the relevant bits (a message sometimes called a *SYN+ACK*). This makes a total of three messages, and for this reason the connection procedure is called a *three-way handshake*.

Key Concept: The normal process of establishing a connection between a TCP client and server involves three steps: the client sends a *SYN* message; the server sends a message that combines an *ACK* for the client's *SYN* and contains the server's SYN; and then the client sends an *ACK* for the server's *SYN*. This is called the *TCP three-way handshake*.

Table 152 describes in detail how the three-way handshake works (including a summary of the preparation discussed in the previous topic). It is adapted from the table describing the TCP finite state machine, but shows what happens for both the server and the client over time. Each row shows the state the device begins in, what action it takes in that state and the state to which it transitions. The transmit and receive parts of each of the three steps of the handshake process are shown in the table, as well as in Figure 211.

Table 152: TCP "Three-Way Handshake" Connection Establishment Procedure

Client			Server		
Start State	**Action**	**Move To State**	**Start State**	**Action**	**Move To State**
CLOSED	The client cannot do anything until the server has performed a passive *OPEN* and is ready to accept a connection. (Well, it can try, but nothing will be accomplished until the server is ready.)	—	***CLOSED***	The server performs a passive *OPEN*, creating a transmission control block (TCB) for the connection and readying itself for the receipt of a connection request (*SYN*) from a client.	***LISTEN***
CLOSED	**Step #1 Transmit:** The client performs an active OPEN, creating a transmission control block (TCB) for the connection and sending a *SYN* message to the server.	***SYN-SENT***	***LISTEN***	The server waits for contact from a client.	—
SYN-SENT	The client waits to receive an *ACK* to the *SYN* it has sent, as well as the server's *SYN*.	—	***LISTEN***	**Step #1 Receive, Step #2 Transmit:** The server receives the *SYN* from the client. It sends a single *SYN+ACK* message back to the client that contains an *ACK* for the client's *SYN*, and the server's own *SYN*.	***SYN-RECEIVED***
SYN-SENT	**Step #2 Receive, Step #3 Transmit:** The client receives from the server the *SYN+ACK* containing the *ACK* to the client's *SYN*, and the *SYN* from the server. It sends the server an *ACK* for the	***ESTABLISHED***	***SYN-RECEIVED***	The server waits for an *ACK* to the *SYN* it sent previously.	—

	server's *SYN*. The client is now done with the connection establishment.				
ESTABLISHED	The client is waiting for the server to finish connection establishment so they can operate normally.		***SYN-RECEIVED***	**Step #3 Receive:** The server receives the *ACK* to its *SYN* and is now done with connection establishment.	***ESTABLISHED***
ESTABLISHED	The client is ready for normal data transfer operations.		***ESTABLISHED***	The server is ready for normal data transfer operations.	

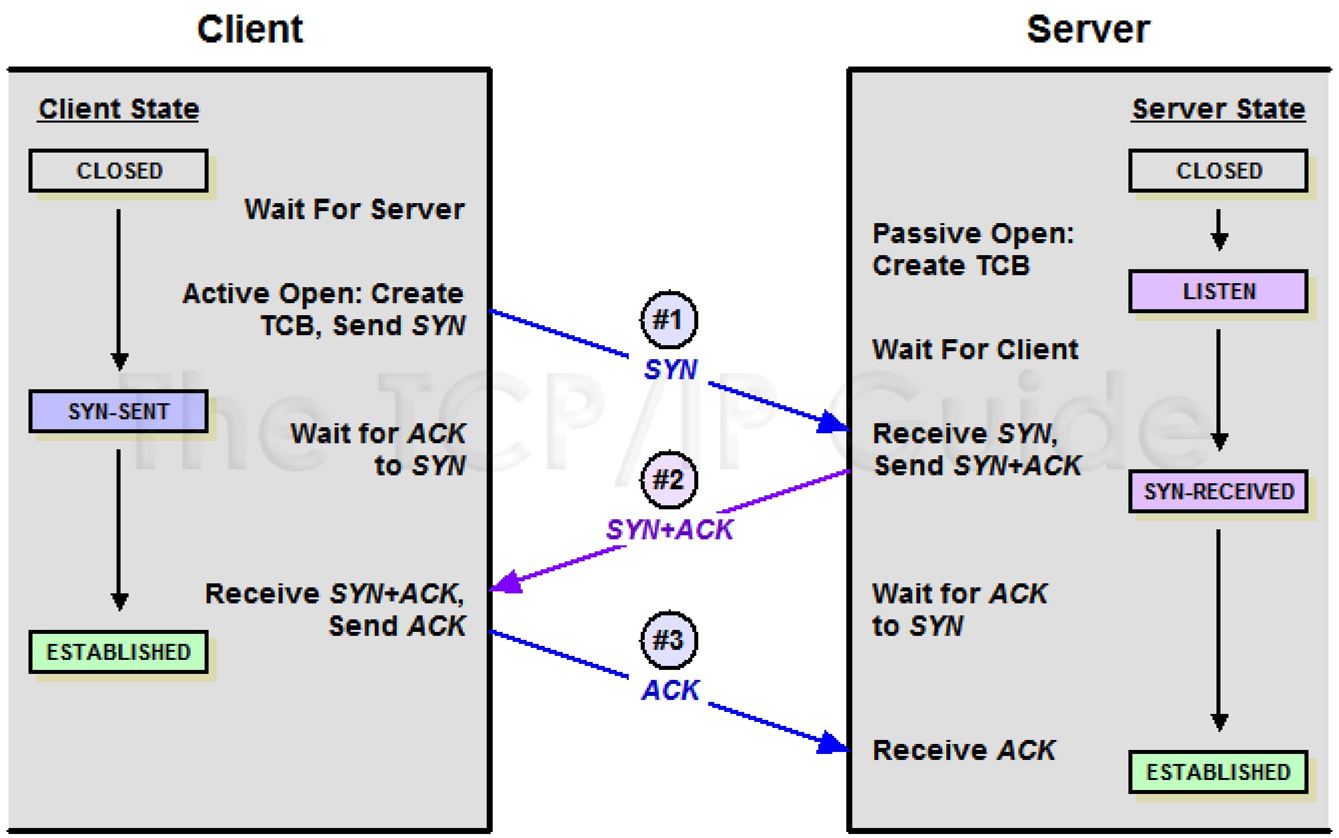

Figure 211: TCP "Three-Way Handshake" Connection Establishment Procedure

This diagram illustrates how a conventional connection is established between a client and server, showing the three messages sent during the process and how each device transitions from the *CLOSED* state through intermediate states until the session is *ESTABLISHED*.