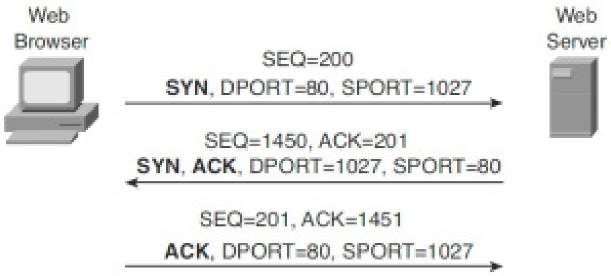
Basic TCP operation

niktips.wordpress.com/2012/06/06/basic-tcp-operation/

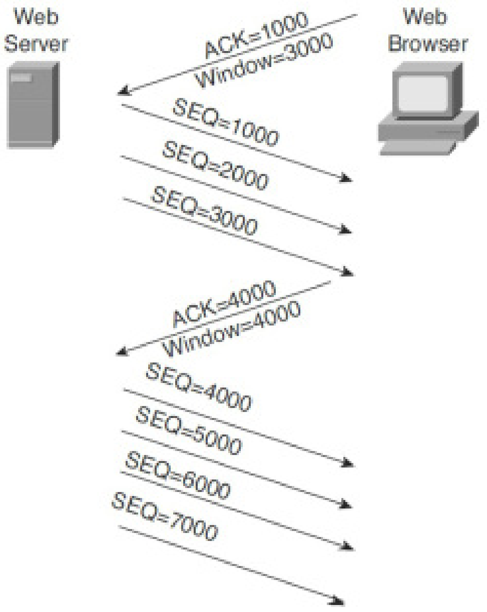
The main purpose of TCP is error recovery and flow control. TCP is a connection-oriented protocol. It means that before sending any data it establishes connection and terminates it upon completion.


Web Browser
Web Server
SEQ=200
SYN, DPORT=80, SPORT=1027
SEQ=1450, ACK=201
SYN, ACK, DPORT=1027, SPORT=80
SEQ=201, ACK=1451
ACK, DPORT=80, SPORT=1027

During connection establishment server and client agree upon sequence and acknowledgment numbers. Implicitly client also notifies server of its source port. Sequence is a characteristic of TCP data segment. Sequence starts with a random number and then each time a new packet is sent, sequence is incremented by the number of bytes sent in the previous TCP segment. Acknowledgment segment is almost the same but from the receiver side. It does not contain data and is equal to sender's sequence number incremented by the number of bytes received (you will see example below). ACK segment acknowledges that host has received sent data.

Client-server handshake is performed in three steps:

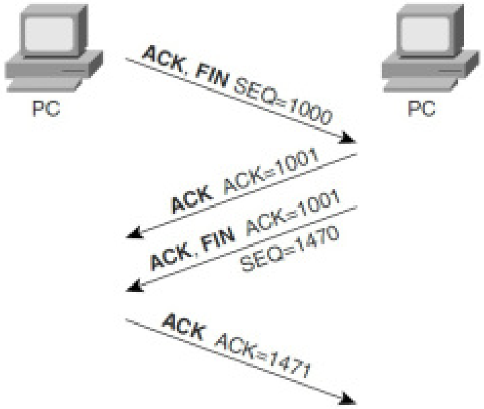
1. Client sends packet to the server with the SYN flag set, indicating that it's willing to establish a connection. Client sets its sequence to a random number and sends the segment to the server.
2. Server acknowledges that it agrees to establish connection, sets its sequence to a random number, acknowledgment to the client sequence + 1 and send them to the client.
3. In the third message client sets its acknowledgment to the server's sequence + 1 and send back to the server.



Now when both client and server know each other's sequence and acknowledgment numbers, they can start sending data. Here it's important to point out that TCP uses "windows" to send data. Window essentially is a number of bytes host can send before it receives acknowledgment from the recipient. Lets say window equals 3000 and server sends three segments 1000 bytes each. Initially we pick random SEQ number which equals to 1000 and increment it by 1000, which is the segment size, with each next segment. When client has received all three packets it answers with the ACK equal to the last SEQ number + size of the last packet. And so on. If no errors occur receiver usually increases its window.

Finally, when PC1 wants to close the connection it sends a FIN segment. PC2 on the opposite side notifies the application that the connection is closing. But since it takes some time for the application to complete its operation, PC2 sends an ACK to the PC1, to notify it of an agreement to finish the connection. Otherwise after a timeout PC1 will continue to retransmit the FIN segment thinking that it has been lost. When application is terminated PC2 sends its FIN segment, PC1 replies with ACK and connection is closed.

PS: All rights to the pictures go to Wendell Odom


PC
PC
ACK, FIN SEQ=1000
ACK ACK=1001
ACK, FIN ACK=1001 SEQ=1470
ACK ACK=1471