**Network/Socket Programming**

Socket programming is a way of connecting two nodes on a network to communicate with each other. Server forms the listener socket while client reaches out to the server.

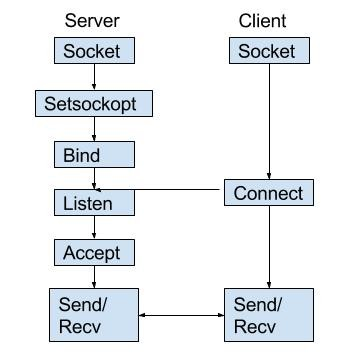
****

Figure 1: State diagram for server and client model

\* If there are different options, for parameters, we use the one that has this next to it

“ヽ( ͝° ͜ʖ͡°)ﾉ”

1. **UNIX:[[1]](#endnote-2)**

a. Server:

* Socket creation:

- int sockfd = socket(domain, type, protocol)

**sockfd**: socket descriptor, an integer (like a file-handle)   
 **domain**: integer, communication domain

- AF\_INET (IPv4 protocol) ← ヽ( ͝° ͜ʖ͡°)ﾉ

- AF\_INET6 (IPv6 protocol)

**type**: communication type  
 - SOCK\_STREAM: TCP (reliable, connection oriented) ← ヽ( ͝° ͜ʖ͡°)ﾉ  
 - SOCK\_DGRAM: UDP (unreliable, connectionless, fast)  
 **protocol**: Protocol value for Internet Protocol(IP), which is 0. This is the same number which appears on protocol field in the IP header of a packet.

* (Optional) Manipulating options for the socket:

- int setsockopt(int sockfd, int level, int optname, const void \*optval, socklen\_t optlen);

* Bind: binding the socket to a number (address or port number)

- int bind(int sockfd, const struct sockaddr \*addr, socklen\_t addrlen);

* Listen: putting the server socket in a passive mode, where it waits for the client to approach the server to make a connection

- int listen(int sockfd, int backlog);

* Accept: Extracting the first connection request on the queue of pending connections for the listening socket, sockfd, creates a new connected socket, and returns a new file descriptor referring to that socket. Connection is established and ready for data transfering.

- int new\_socket= accept(int sockfd, struct sockaddr \*addr, socklen\_t \*addrlen);

b. Client:

* Socket connection: same as server
* Connect: calling connects the socket referred to by the file descriptor sockfd to the address specified by addr. Server’s address and port is specified in addr.

- int connect(int sockfd, const struct sockaddr \*addr, socklen\_t addrlen);

*\*Based on UNIXServer.cpp and UNIXClient.cpp codes*

1. **Windows:**

1. https://www.geeksforgeeks.org/socket-programming-cc/ [↑](#endnote-ref-2)