Assignment 7 - Write a program using UDP Sockets for your chosen applications (using java / c)

1. Explain UDP & Socket APIs. Explain different parameters to create a socket.

User Datagram Protocol (UDP) is a transport layer protocol. Unlike TCP, it is an unreliable & connectionless protocol. So there is no need to establish a connection prior to the data transfer.

UDP socket routines enable simple IP communication using the user datagram protocol (UDP). Messages/ datagram are sent to other hosts on an IP network without the need to set up special transmission channels or data paths beforehand. The UDP socket needs to be opened for communication.

UDP socket APIs is a collection of socket calls that enables you to perform the following primary communication function between application programs -

1) Set up and establish connections to other users on the network
2) Send & receive data to & from other users
3) Close down connections.

~~The socket fu~~

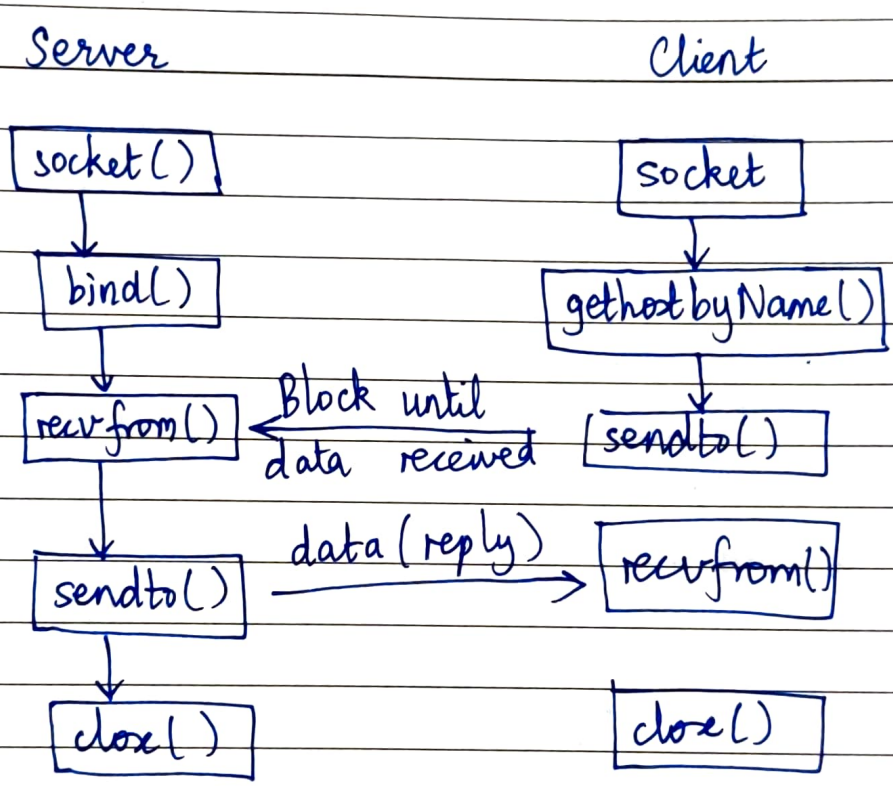
Parameters to Create Socket - The socket function is used to create a new socket descriptor. It takes 3 arguments.

(i) for the standard socket programming, the 1st argument is always AF_INET.

2) The 2nd argument is typically SOCK_DGRAM for UDP socket, ~~for stream~~ Socket for 1st and end arguments are always AF_INET and SOCK_RAW.

3) The 3rd argument varies depending on what the socket will be used for.

Describe the socket primitives with diagram and explanation of each primitive.



sock = Socket(domain type protocol) → allocates a new socket in the kernel, returns handle to this socket.

bind(sock, addr) → associates sock with address.

3. Write logic for your Client and Server.

Logic for Client program -

1. Create the socket object for carrying the data.
2. Create while loop
3. Convert input string to byte array
4. Create the datagram packet for sending data.
5. Invoke send call to actually send data.
6. Break the loop.

Logic for Server Program -

1. Create socket to listen at port 1234.
2. Create a datagram packet to receive the data.
3. Receive the data in byte buffer.
4. Exit if the client sends datagram_socket.
5. Clear the buffer after every message.