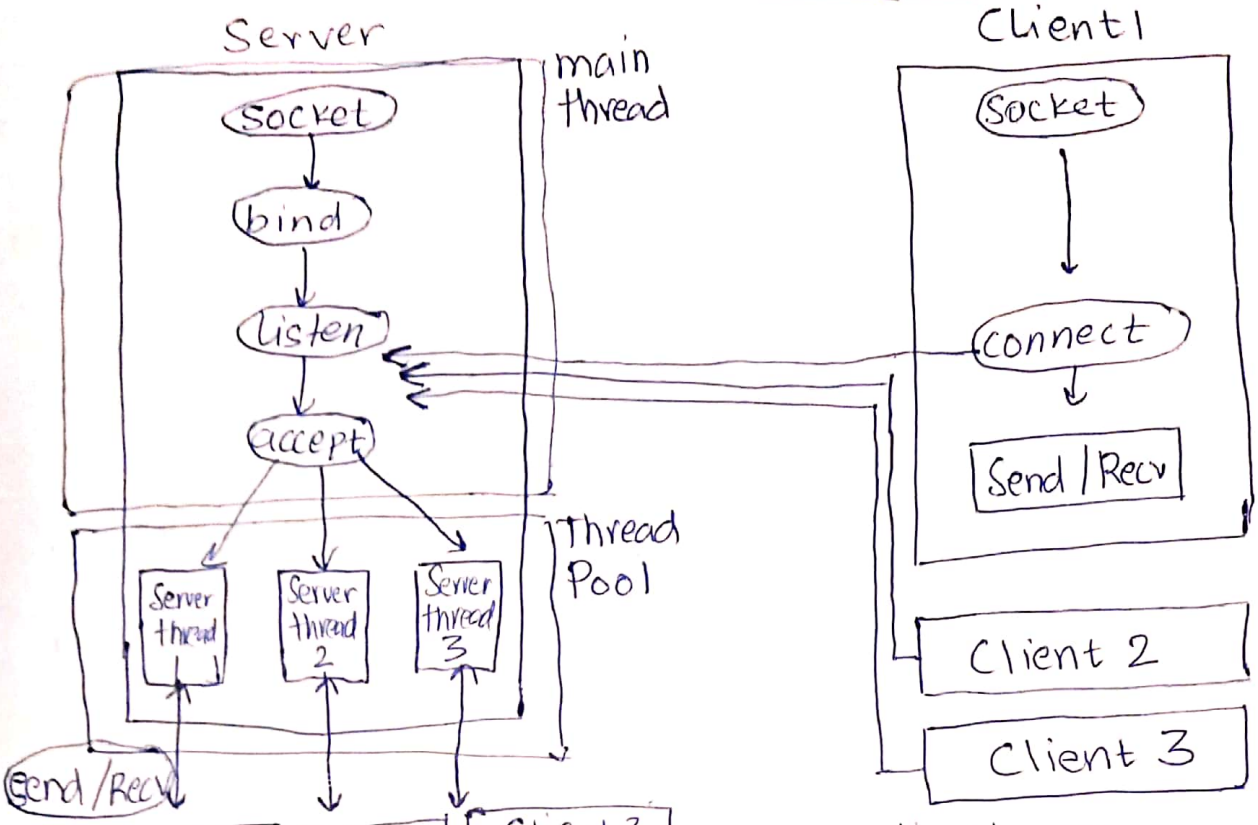
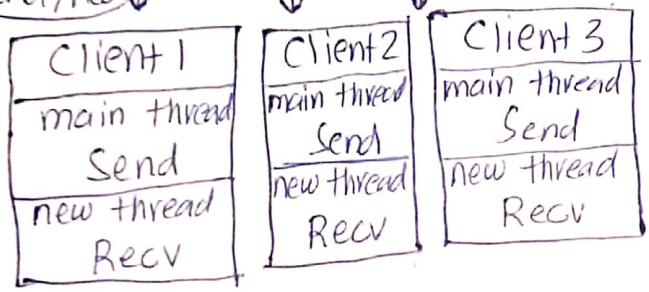
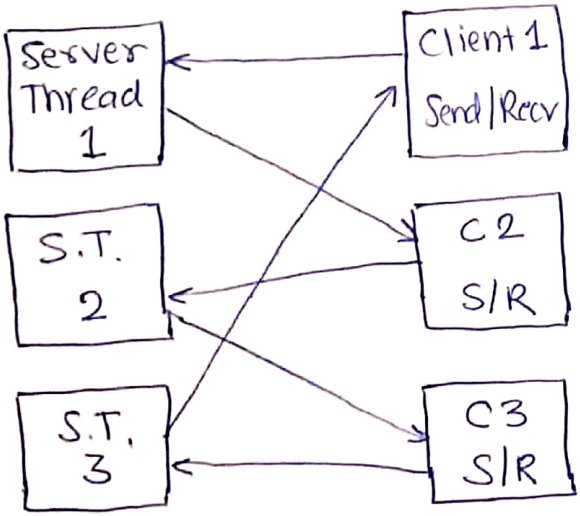
① Flowchart for TCP multi-threaded Server-Client





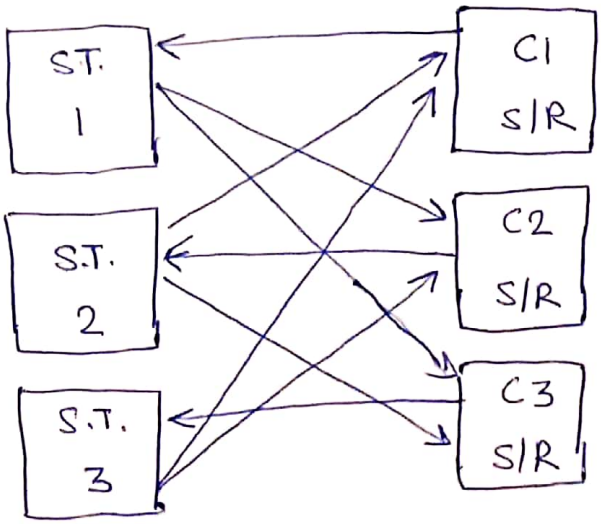
Each client on different socket handled by a different server thread.

For ONE to ONE:



shows message forwarding from server in cyclic manner

For Broadcast.



Shows sending of message to remaining clients.

TCP Pseudocode :

~~Client~~ : Server.

- Create Socket
- Bind ~~Connect~~ Socket ~~Bind~~
- Start listening
- Accept Client in a client structure and keep indexing them
- Create new server thread as clients keep connecting
- Threads call the function to send / Recv.

loop

Client Structure

```
index
SOCKID
sockaddr_in clientAddr
```

- For one-one

  send to Client(index+1)% total clients

  Recv from all Clients

For Broadcast

Send to all clients with index != current Client

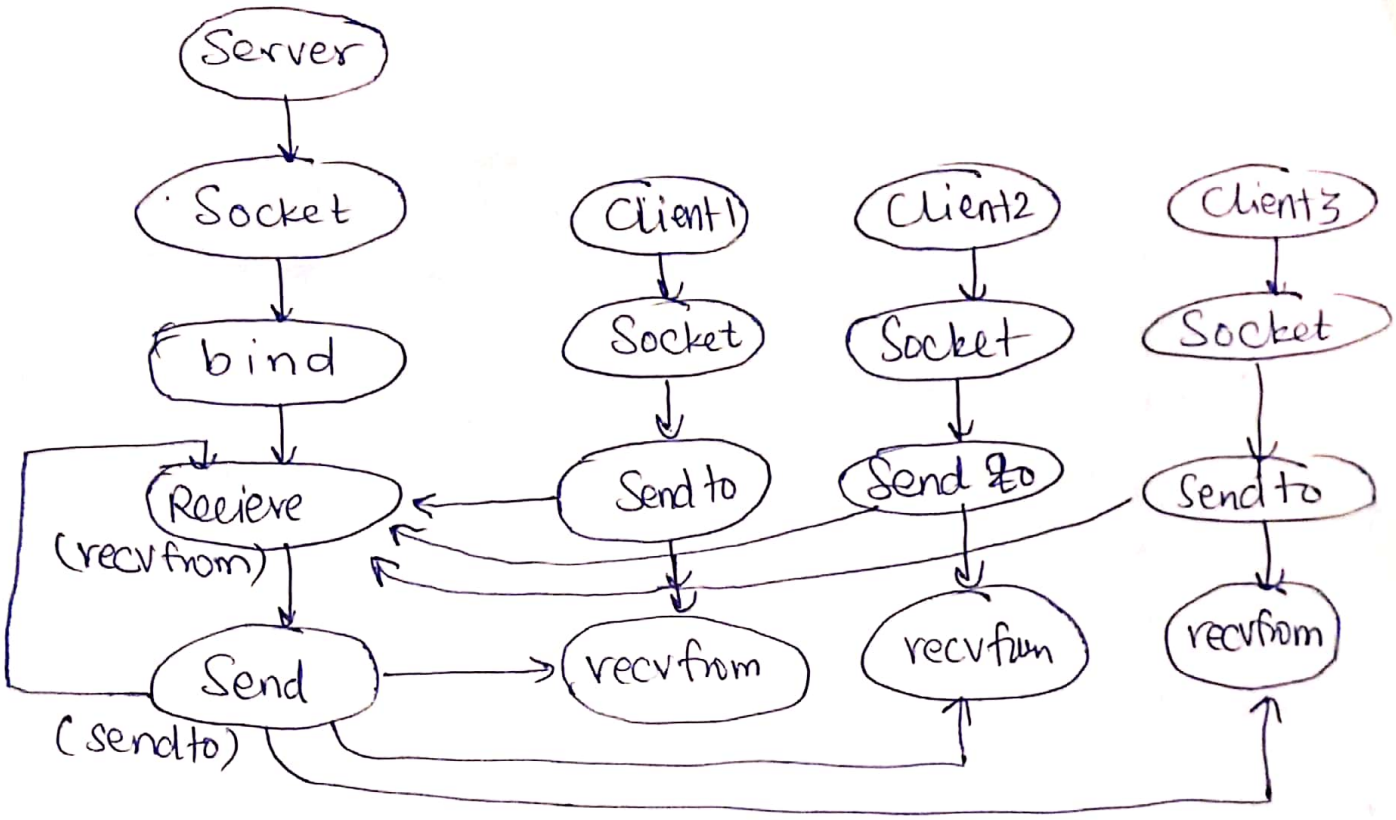
Recv from all Clients

Client :

- Create Socket
- Connect to server
- Wait till server forwards the message in the main thread.
- Make a new thread for recieving messages
- Do the reading in main thread.
- This logic works for both broadcast and one-one mode

loop

(2) Flowchart for Multi-client UDP Server:



For ONE-to-ONE:

incomming message

sin-portX

find index

Client Array

0	sin-port 1
1	sin-port 2
2	sin-port 3
% Size	

sin-portX = Array[index+1]

Send reply to this sin-portX By changing

cliaddr.sin_port = sinportX

For BROADCAST:

incomming message

sin-portX

find index

Loop

Send message to remaining (total clients) - 1 by changing sin-port with index

UDP Pseudocode:

Server

- Create socket
- Bind socket
- Read temporary messages and index the sin_port in cliaddr in an array to identify clients.
- Now when any client sends the message you can identify its index by comparing its sin_port value with the array storing all sin_port values.
- Broadcast: reply to all clients other than the current index by changing sin_port value of cliaddr using indexing array and for loop.
- One to One: ~~Find the~~ Use the index you found out, increase it by 1 in cyclic way. Change sin_port value to this index's value and send the message.

Client.

- Create Socket
- Send temporary message to server for identification
- Loop:
  Send text messages in main theaad
- Use new thread to reciere messages
- Loop:
  Reciene messages in the function called by new thread