

---

## Lab5- Socket programming

### TCP Server-Client implementation in C

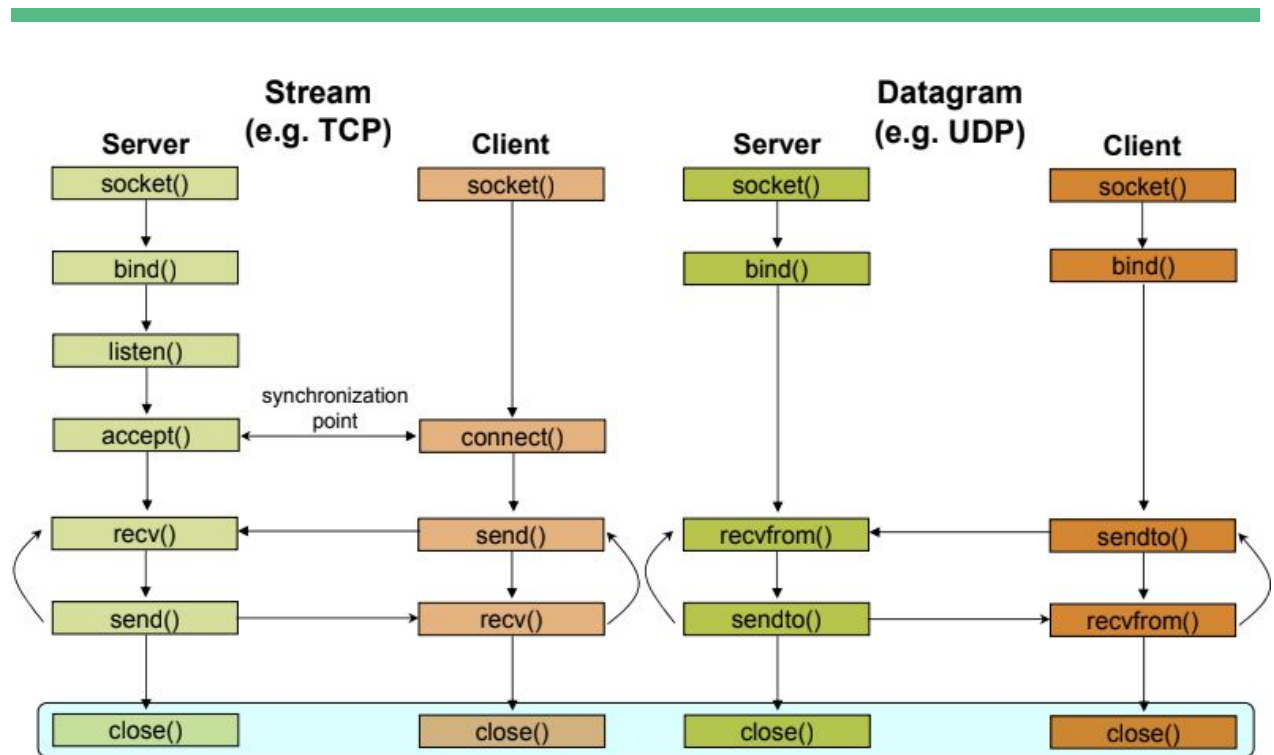
Socket programming is a way of connecting two nodes on a network to communicate with each other. One socket(node) listens on a particular port at an IP, while another socket reaches out to the other to form a connection. The server forms the listener socket while the client reaches out to the server. TCP/IP provides end-to-end connectivity specifying how data should be formatted, addressed, transmitted, routed, and received at the destination can be used on the internet, and in stand-alone private networks, it is organized into layers. The entire process can be broken down into the following steps:

#### **TCP Server –**

1. using `create()`, Create a TCP socket.
2. using `bind()`, bind the socket to the server address.
3. using `listen()`, but the server socket in a passive mode, where it waits for the client to approach the server to make a connection
4. using `accept()`, At this point, the connection is established between client and server, and they are ready to transfer data.
5. Go back to Step 3.

#### **TCP Client –**

1. Create a TCP socket.
2. connect newly created client socket to the server.



```
Activities Terminal Thu 6:51 PM
harsh@harshs: ~/Desktop/ANCP
File Edit View Search Terminal Help
harsh@harshs:~/Desktop/ANCP$ ./server
Socket successfully created..
Socket successfully binded..
Server listening..
server accept the client...
From client: Hello
    To client : Hi
From client: How are you
    To client : I am fine
From client: exit
    To client : exit
Server Exit...
harsh@harshs:~/Desktop/ANCP$

harsh@harshs:~/Desktop/ANCP$ ./client
Socket successfully created..
connected to the server..
Enter the string : Hello
From Server : Hi
Enter the string : How are you
From Server : I am fine
Enter the string : exit
From Server : exit
Client Exit...
harsh@harshs:~/Desktop/ANCP$
```

---

## Conclusion

Through this practical application, we learned about TCP and implemented a basic communication setup between client and server.