AIM :-

To implement Transmission contact protocol usin Socket programming in c language.

PROBLEM ANALYSIS:-

TCP contains two main parts

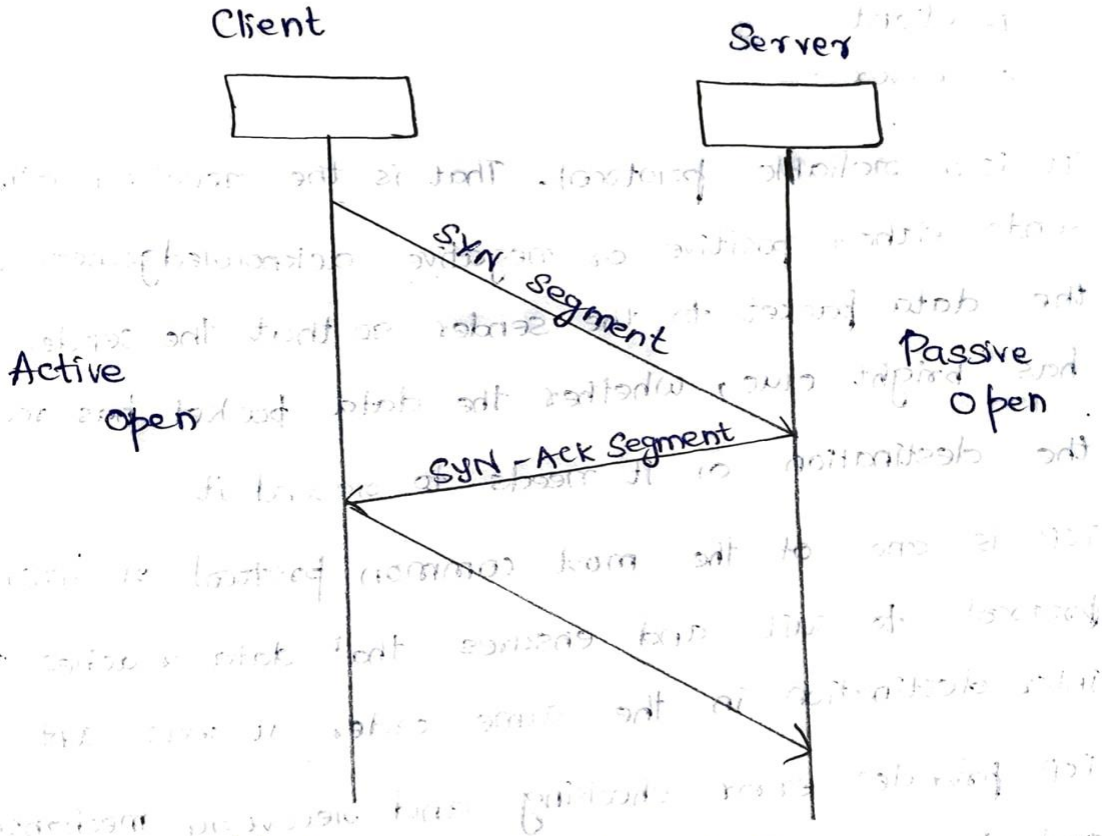
1. client
2. Server.

* It is a reliable protocol. That is the receiver al sends either positive or negative acknowledgement the data packet to the sender so that the sender has bright clue, whether the data packet has rec the destination or it needs to resend it.
* TCP is one of the most common protocol of inter protocol to suit and ensures that data reaches t inter destination in the same order it was sent.
* TCP provides error checking and recovery mechanism and provides end to end communications and also provides flow control and quality of service and i operates in client | server point to point note.

WORKING of TCP

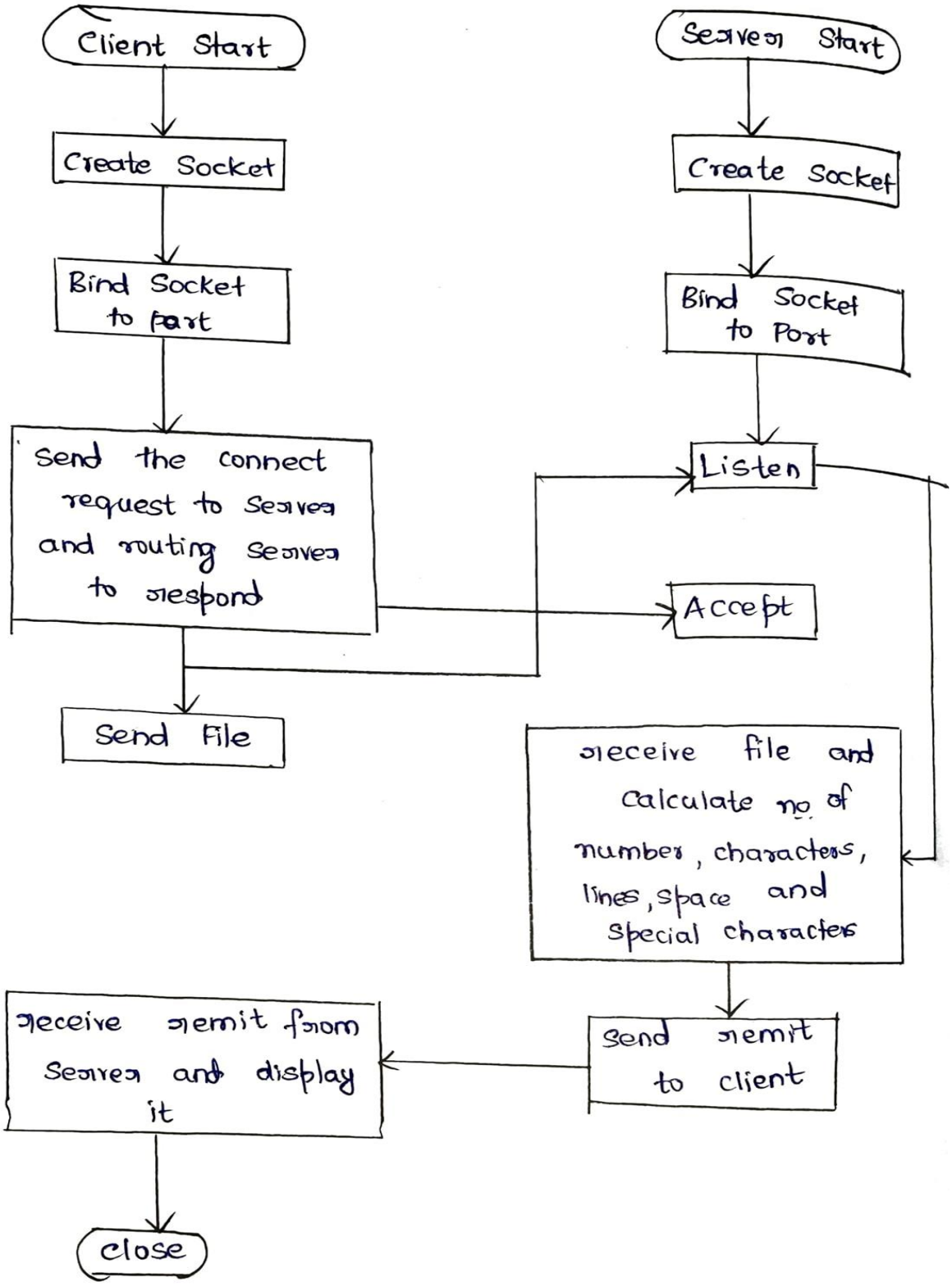
In TCP the connection is established by using 3-way hand-shaking client sends the segment wi its sequence number; the server in return sends its segments with its own sequence number as

well as the ack sequence number in one more client. Sequence number when the client receive the ack and its segment, then it sends the Ack to the server. In this way connection is established between client and server.



WORKING OF THE TCP

FLOWCHART :-


Client Start
Server Start
Create Socket
Create Socket
Bind Socket to Port
Bind Socket to Port
Send the connect request to Server and routing Server to respond
Listen
Accept
Send File
receive file and calculate no. of number, characters, lines, space and special characters
receive remit from Server and display it
Send remit to client
close

Refer server.py

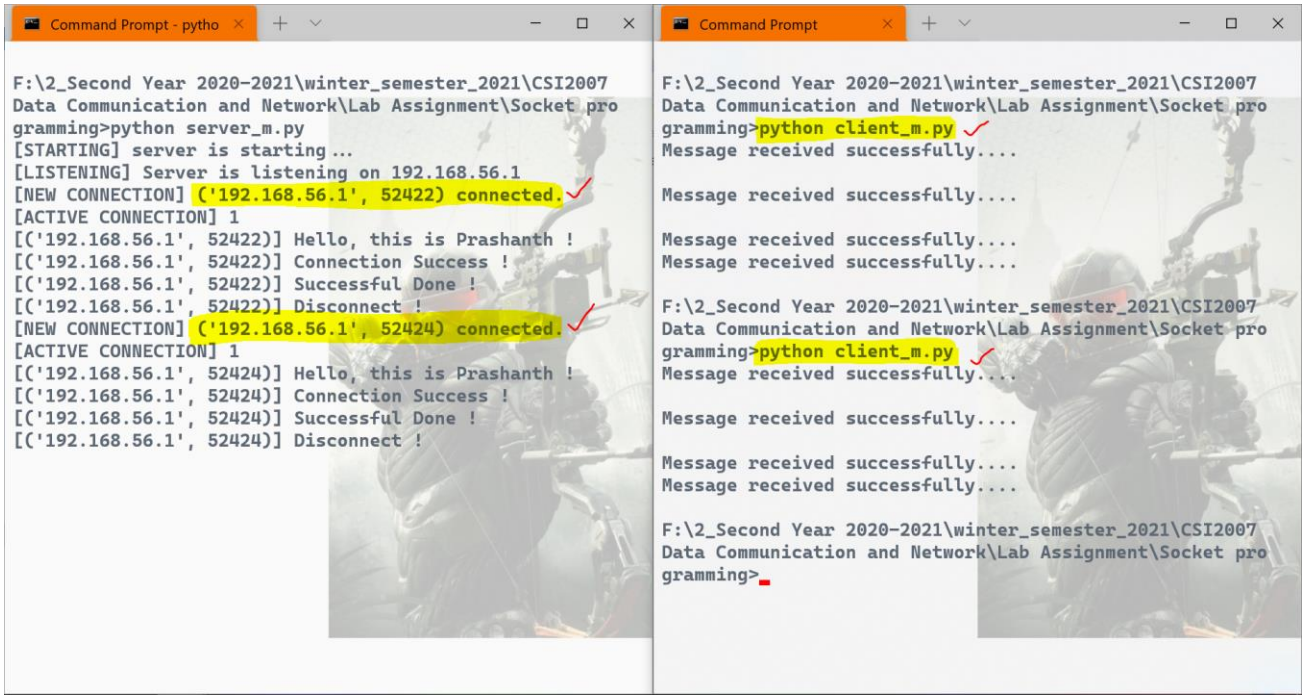
```python
import socket as soc
import threading as th

'''
1) Pick the port
2) Pick the server
3) Pick the socket
4) Bind the socket
'''
HEADER = 64
PORT = 9999
# or use loopback address [localhost]
SERVER = soc.gethostbyname(soc.gethostname())
# gethostname returns device name  gethostby name returns it ip address
ADDR = (SERVER, PORT)
FORMAT = "utf-8"
DISCONNECT MESSAGE = "Disconnect !"
RESPONSE_MESSAGE = "Message received successfully...."

# Socket creation
# first argument --> famiily (categories) 2nd -->ways of sending data (protocols)
server = soc.socket(soc.AF_INET, soc.SOCK_STREAM)
# the first argument tells the what type of ip address or type of address that we gonna be acceptir
server.setsockopt(soc.SOL_SOCKET, soc.SO_REUSEADDR, 1) #for immediate use of that port
server.bind(ADDR)


def handle_client(client_conn, addr):
    print(f"[NEW CONNECTION] {addr} connected.")
    connected = True
    while connected:
        msg_length = client_conn.recv(HEADER).decode(FORMAT)
        # conn.recv() takes argument of length of the msg
        # every time we send msg need to encode in byte format
        if msg_length:
            msg_length = int(msg_length)
            msg = client_conn.recv(msg_length).decode(FORMAT)
```

```python
            if msg == DISCONNECT_MESSAGE:
                connected = False
            print(f"[{addr}] {msg}")
            response_msg_length = len(RESPONSE_MESSAGE)
            send_length = str(response_msg_length).encode(FORMAT)
            send_length += b' ' * (HEADER - len(send_length))
            client_conn.send(send_length)
            client_conn.send(RESPONSE_MESSAGE.encode(FORMAT))

    client_conn.close()


def start():
    # listen
    server.listen()
    print(f"[LISTENING] Server is listening on {SERVER}")
    while True:  # continue to listen until turn off or it crashes
        # this line waits for new connection to occur & it stores the address as well as the port
        client_conn, addr = server.accept()
        # this function will return the socket_object & address-port tuple
        thread = th.Thread(target=handle_client, args=(client_conn, addr))
        thread.start()
        # to exclude the main thread (this paython program ) we sub 1
        print(f"[ACTIVE CONNECTION] {th.activeCount()-1}")


print("[STARTING] server is starting...")
start()
```

Refer client.py

```python
import socket as soc

HEADER = 64
PORT = 9999
FORMAT = "utf-8"
DISCONNECT_MESSAGE = "Disconnect !"
SERVER = soc.gethostbyname(soc.gethostname())
ADDR = (SERVER, PORT)
client = soc.socket(soc.AF_INET, soc.SOCK_STREAM)
client.connect(ADDR)


def send(msg):
    message = msg.encode(FORMAT)
    msg_length = len(message)
    send_length = str(msg_length).encode(FORMAT)
    send_length += b' ' * (HEADER - len(send_length))
    # b' ' -->byte format specifier padding
    client.send(send_length)
    client.send(message)
    msg_length = client.recv(HEADER).decode(FORMAT)

    msg_length = int(msg_length)
    msg = client.recv(msg_length).decode(FORMAT)
    print(msg)


send("Hello, this is Prashanth !")
input()
send("Connection Success !")
input()
send("Successful Done !")
send(DISCONNECT_MESSAGE)
```

Output



Server is always open, and the client tries to access the server twice.