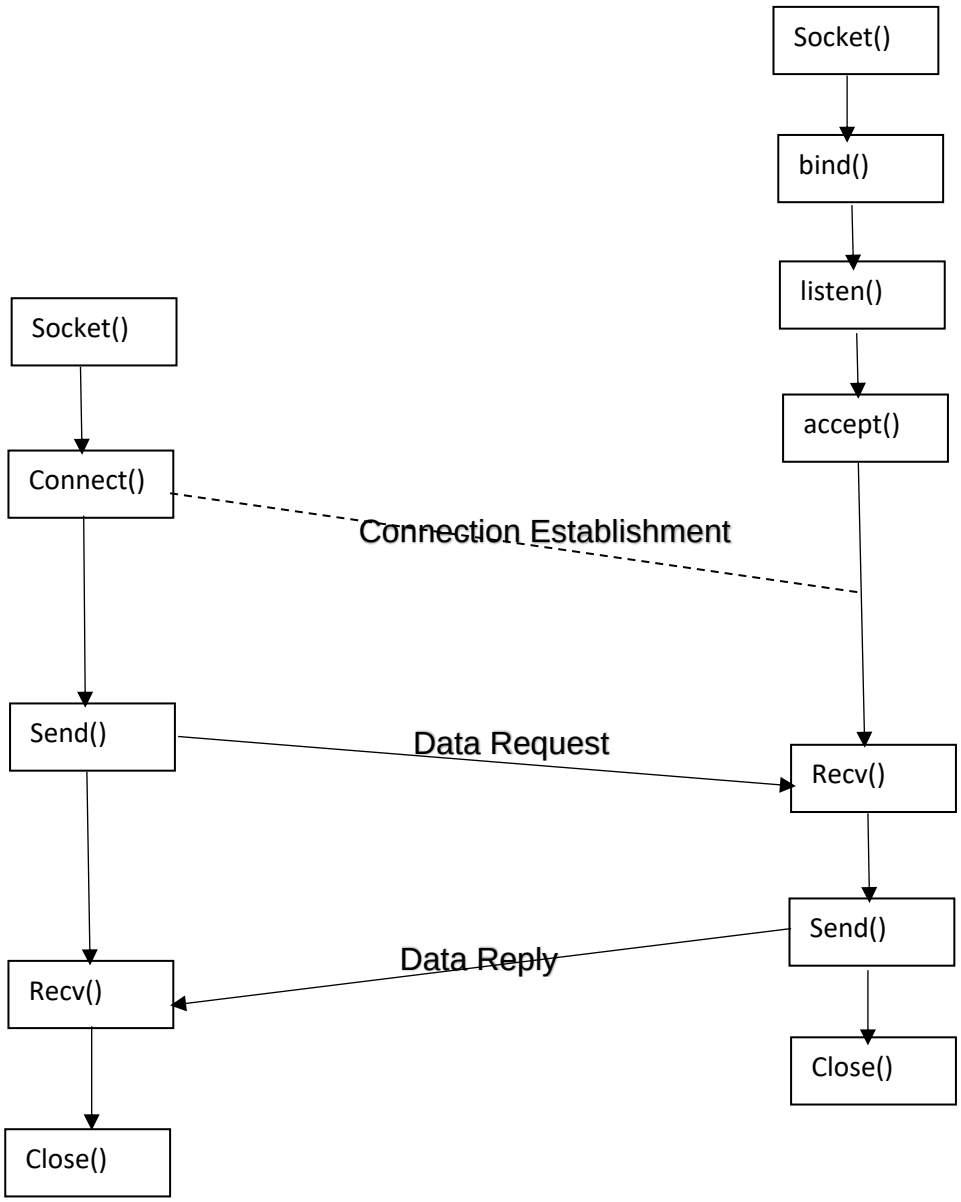
CLIENT SERVER APPLICATION:

Developing **a client-server application** using **TCP** in C++ for Windows involves creating two separate programs: **a client program** that sends requests to a server, and **a server program** that listens for requests from clients and responds to them

TCP Client **TCP Server**



Block diagram of TCP server-client communication

This implementation is designed to accept connections from multiple clients at the same time. Each time a new client connection is accepted, a new thread is spawned to handle that connection. As a result, multiple clients can connect to the server and send messages simultaneously without interfering with each other. Each client connection is handled independently in its own thread.

We are using WinSock network Library available in C++ for Windows to use sockets. In order to use Windows threads, we need to include the <windows.h> header file, and use the CreateThread() function to create threads.

Create the server application –

1)Initialize Winsock using the WSAStartup() function.

2)Create a socket using the socket() function and specify the type of socket (in this case, TCP) and the address family.

3)Bind the socket to a specific IP address and port using the bind() function.

4)Listen for incoming connections using the listen() function.

5)Accept incoming connections using the accept() function, which creates a new socket for the incoming client connection.

6)In this code, each time a client connects, a new thread is created to handle the communication with that client.

7)The handleClient() function is the thread entry point, and is called for each connected client, and it enters a loop to read messages from the client and echo them back.

8)The mentioned function then receive data from the client using the recv() function and send data back to the client using the send() function.

9) We also need to use the CloseHandle function to close the thread handle after the thread has exited.

10)Close the socket and clean up the Winsock environment using the closesocket() and WSACleanup() functions.

Create the client application:

1)Initialize Winsock using the WSAStartup() function.

2)Create a socket using the socket() function and specify the type of socket (TCP) and the address family.

3)Connect to the server using the connect() function and specify the server's IP address and port number.

4)In this code, the CreateThread() function is used to create a new thread to send messages to the server.

5)The sendMessages() function is run in a separate thread to send messages to the server data to the server using the send() function and receive data back from the server using the recv() function.

6)Inside the mentioned function, we display the round trip time for each message, the running average round trip time for all messages, and the running throughput rate for all messages.

7)The main thread waits for the sendMessages() thread to complete using the join() function.WaitForSingleObject

8)Close the socket and clean up the Winsock environment using the closesocket() and WSACleanup() functions