

**Aim** - Write a Program to transfer the content of a requested file from server to client using TCP/IP sockets.

**OBJECTIVES**

- 1. To provides end-to-end data communication specifying how data should be packetized, addressed, transmitted, routed, and received.
- 2. To bound a socket to a port number so that the TCP layer can identify the application that data is destined to be sent to.
- 3. To Transfer the requested file using sockets in TCP/IP .

**THEORY**

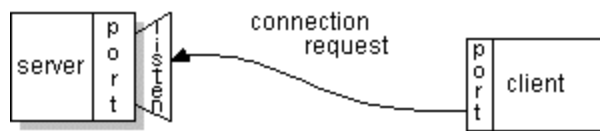
TCP/IP, or the Transmission Control Protocol/Internet Protocol, is a suite of communication protocols used to interconnect network devices on the internet. TCP/IP can also be used as a communications protocol in a private network (an intranet or an extranet).  
A socket is one endpoint of a two-way communication link between two programs running on the network. A socket is bound to a port number so that the TCP layer can identify the application that data is destined to be sent to. An endpoint is a combination of an IP address and a port number.

**Primitive of Socket-**

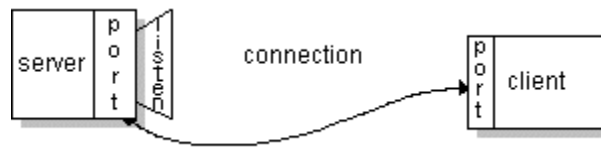
Primitive	Meaning
Socket	Create a new communication endpoint
Bind	Attach a local address to a socket
Listen	Announce willingness to accept connections
Accept	Block caller until a connection request arrives
Connect	Actively attempt to establish a connection
Send	Send some data over the connection
Receive	Receive some data over the connection
Close	Release the connection

**METHODOLOGY**

Normally, a server runs on a specific computer and has a socket that is bound to a specific port number. The server just waits, listening to the socket for a client to make a connection request. On the client-side: The client knows the hostname of the machine on which the server is running and the port number on which the server is listening. To make a connection request, the client tries to rendezvous with the server on the server's machine and port. The client also needs to identify itself to the server so it binds to a local port number that it will use during this connection. This is usually assigned by the system.



If everything goes well, the server accepts the connection. Upon acceptance, the server gets a new socket bound to the same local port and also has its remote endpoint set to the address and port of the client. It needs a new socket so that it can continue to listen to the original socket for connection requests while tending to the needs of the connected client.



On the client side, if the connection is accepted, a socket is successfully created and the client can use the socket to communicate with the server. The client and server can now communicate by writing to or reading from their sockets.

## OUTPUT

```
Terminal
-----CLIENT SIDE-----
Socket Connection Active
file descriptor 3 received

Please enter file name to receive:
demofile.txt

-----Data Received-----
Under the halo of the moon .

-----
Please enter file name to receive:
|
```

```
Terminal
-----SERVER SIDE-----
Socket Connection Established
Successfully binded!

Waiting for file name...

File Name Received: demofile.txt

File Successfully opened!

Waiting for file name...
|
```

## CONCLUSION

A server creates a socket, binds the socket to an IP address and port number and then listens for incoming connections. When a client connects to the server, a new socket is created for communication with the client.