

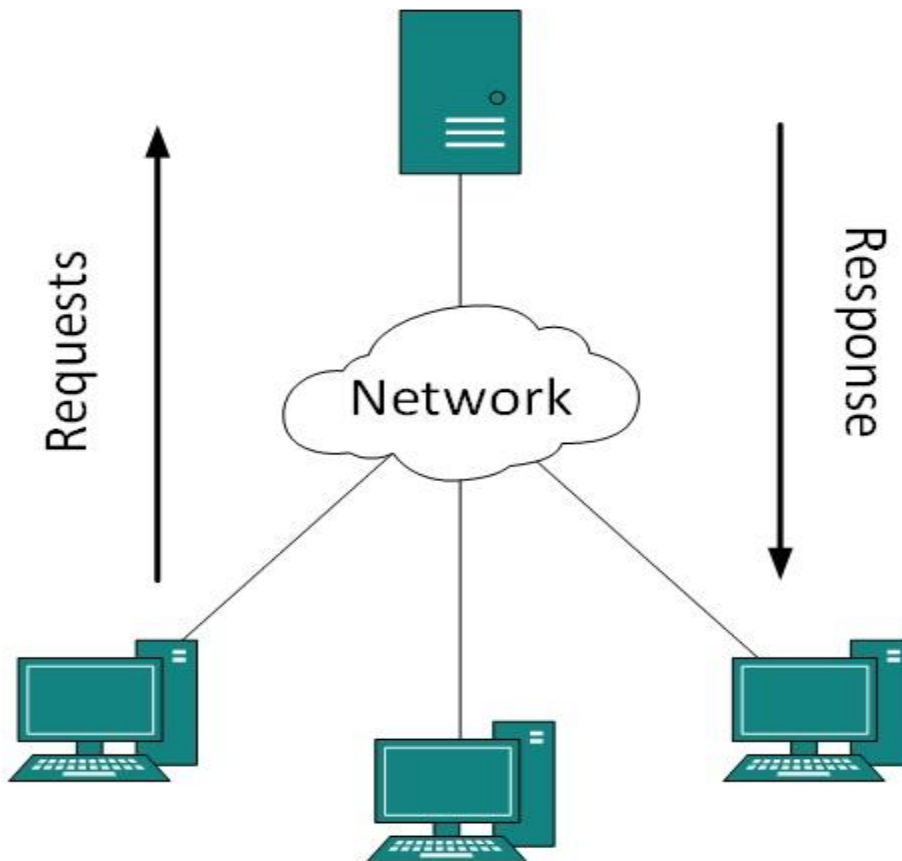
# Client Server Model

## Introduction

Two remote application process can communicate in mainly two different fashions:

- **Peer-to-peer:** Both remote processes are at same level and exchange data using some shared resource.
- **Client-Server:** One the remote process acts as Client and requests some resource from another application process acting as Server.

In client-server model, any process can act as Server or Client. This not the machine or size of the machine or its computing power which makes it server but it is the feature of serving request that makes it server.



[Image: Client Server Model]

A system can act as Server and Client simultaneously. That is, one process is acting as Server and another is acting as a client. This may also happen that both client and server processes reside on the same machine.

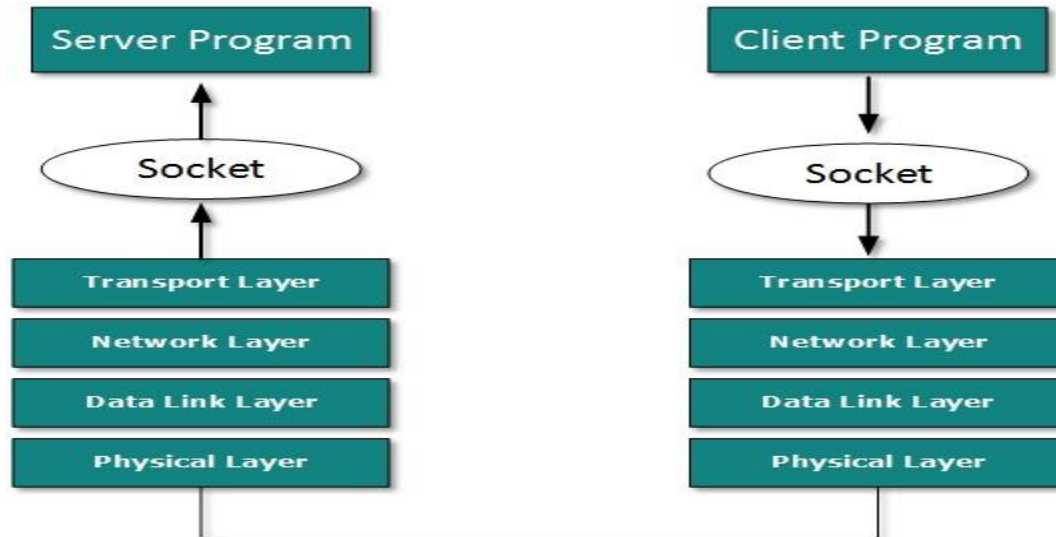
## Communication:

Two processes in client-server model can interact in various ways:

- Sockets
- Remote Procedure Calls (RPC)

## Sockets:

In this paradigm, the process acting as Server opens a socket using a well-known (or known by client) port and waits until some client request comes. The second process acting as Client also opens a socket but instead of waiting for an incoming request, client process 'requests first'.



[Image: Socket]

When the request is reached to server, it is served. It can either be an information sharing or resource request.

## Remote Procedure Call

This is a mechanism where one process interacts with another by means of procedure calls. One process (client) calls the procedure lying on remote host. The process on remote host is said to be Server. Both processes are allocated stubs. This communication happens in the following way:

- The client process calls the client stub. It passes all the parameters pertaining to program local to it.
- All parameters are then packed (marshalling) and a system call is made to send them to other side of the network is made.
- Kernel sends the data over the network and the other end receives it.
- The remote host passes data to the server stub where it is unmarshalled.
- The parameters are passed to the procedure and the procedure is then executed.
- The result is sent back to the client in the same manner.

Source:

[http://www.tutorialspoint.com/data\\_communication\\_computer\\_network/client\\_server\\_model.htm](http://www.tutorialspoint.com/data_communication_computer_network/client_server_model.htm)