**GOVERNMENT COLLEGE OF ENGINEERING, JALGAON**
(An Autonomous Institute of Government of Maharashtra)
Department of Computer Engineering

Name: Vaibhav Bharat Sontakke PRN No: 1942207

Class: T. Y. B. Tech. Batch: L4

Date of Performance: _____ Date of Completion: _____

Subject: DSL Sign. of Teacher with Date: _____

# Experiment No.: 07

**Aim:** Program for Distributed chat server using TCP sockets.

**Title:** Design a distributed Program for Distributed chat server using TCP sockets.

Software Requirements: Ubuntu OS, Eclipse IDE 4.11 **Theory :**

## Client/Server Communication

At a basic level, network-based systems consist of a server, client, and a media for communication as shown in Fig. 1.1. A computer running a program that makes a request for services is called client machine. A computer running a program that offers requested services from one or more clients is called server machine. The media for communication can be wired or wireless network.

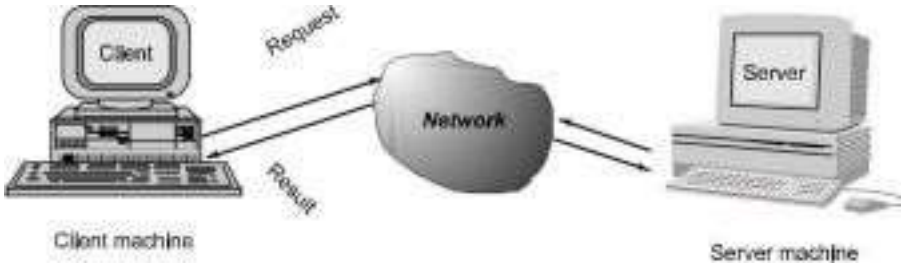



**Fig.1.1 Client – Server communication**

The client-server model is a distributed communication framework of network processes among service requestors, clients and service providers. The client-server connection is established through a network or the Internet.

The client-server model is a core network computing concept also building functionality for email exchange and Web/ database access. Web technologies and protocols built around the client-server model are:

- Hypertext Transfer Protocol (HTTP)
- Domain Name System (DNS)
- Simple Mail Transfer Protocol (SMTP)
- Telnet

Clients include Web browsers, chat applications, and email software, among others. Servers include Web, database, application, chat and email, etc.

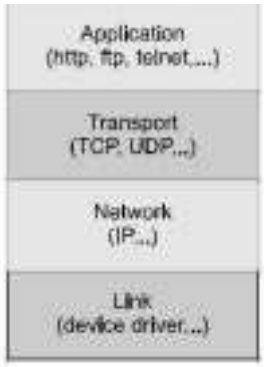


A server manages most processes and stores all data. A client requests specified data or processes. The server relays process output to the client. Clients sometimes handle processing, but require server data resources for completion.

The client-server model differs from a peer-to-peer (P2P) model where communicating systems are the client or server, each with equal status and responsibilities. The P2P model is decentralized networking. The client-server model is centralized networking.

One client-server model drawback is having too many client requests underrun a server and lead to improper functioning or total shutdown. Hackers often use such tactics to terminate specific organizational services through distributed denial-of-service (DDoS) attacks.

Generally, programs running on client machines make requests to a program (often called as server program) running on aserver machine. They involve networking services provided by the transport layer, which is part of the Internet software stack,often called *TCP/IP (Transport Control Protocol/Internet Protocol)*
*stack*, shown in Fig. 1.2. The transport layer comprises two types of protocols, *TCP (Transport Control Protocol)* and *UDP*
*(User Datagram Protocol)*. The most widely usedprogramming interfaces for these protocols are sockets.




- TCP is a connection-oriented protocol that provides areliable flow of data between two computers. Example applicationsthat use such services are HTTP, FTP, and Telnet.

**Fig 1.2 TCP/IP so ware stack**

UDP is a protocol that sends independent packets of data, called *datagrams*, from one computer to another with no guarantees about arrival and sequencing. Example applications that use such services include Clock server and Ping. The TCP and UDP protocols use ports to map incoming data to a particular process running on a computer. Port is represented by a positive (16-bit) integer value. Some ports have been reserved to support common/well known services:

∑ FTP 21/tcp
∑ telnet 23/tcp

∑ SMTP 25/tcp
∑ Login 513/tcp
∑ http 80/tcp,udp
443/tcp,ud
∑ https p

User-level process/services generally use port number value >= 1024. **Object-Oriented Programming with Java**

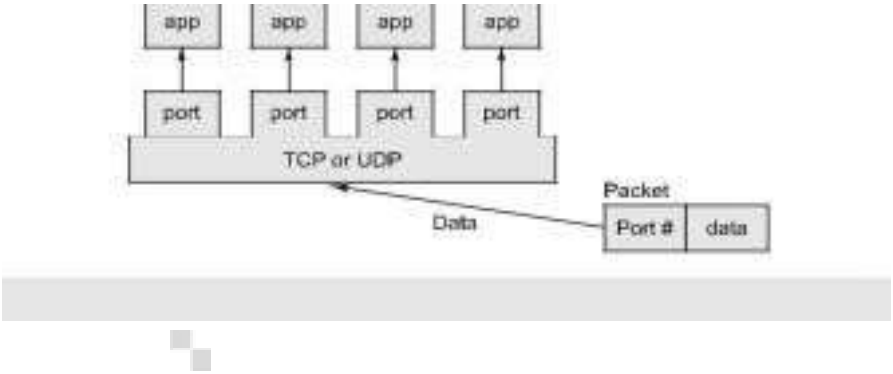





**Fig. 1.3 TCP/UDP mapping of incoming packets to appropriate port/process**

Object-oriented Java technologies—Sockets, threads, RMI, clustering, Web services—have emerged as leading solutions for creating portable, efficient, and maintainable large and complex Internet applications.

**ALGORITHM:**

**Steps:**

**TCP Server**

Step1: Create a socket
Step2: Bind it to the operating system. Step3:
Listen over it.
Step4: Accept connections.
Step5: Receive data from client and send it back to client.
Step6: Close the socket.

**TCP Client** Step1:
Create a socket.
Step2: connect to the server using connect().
Step3: send data to server and receive data from the server.
Step4: Close the socket.

**Conclusion:**

In this practical, We learnt about the implementation of a distributedchat server using

.....

**Miss. Archana Chitte**

**Name & Sign of Course Teacher**