SYNOPSIS
Report on

Server Client Programming

Using Socket And TCP/IP

by

Siddharth - 2100290140134

Session:2022-2023 (4th Semester)

Under the supervision of

Ms Shalika
Assistant Professor

KIET Group of Institutions, Delhi-NCR, Ghaziabad

DEPARTMENT OF COMPUTER APPLICATIONS
KIET GROUP OF INSTITUTIONS, DELHI-NCR, GHAZIABAD-201206
(MARCH- 2023)

ABSTRACT

The project aims to develop a client-server communication system using sockets and TCP/IP protocols. The system allows multiple clients to connect to a server and exchange data in real-time. The project involves designing and implementing a server application that listens for incoming client connections and manages the communication between clients. The clients can connect to the server using their IP address and port number, and the server can authenticate and authorize clients before allowing them to send or receive data. The system employs socket programming to establish reliable connections and TCP/IP protocol for error-free and ordered data transfer. The project also involves testing the system's performance and scalability under different conditions to ensure its effectiveness and efficiency in handling large-scale communication. The resulting system provides a robust and flexible communication platform that can be used in various domains, including gaming, messaging, and data streaming applications.

C and C++ Programming languages can be used to implement a client-server communication system using sockets and TCP/IP protocols.C and C++ are low-level languages that offer direct access to the operating system's networking capabilities. They are well-suited for developing high-performance, low-latency network applications.

Keywords:client,server,TCP protocol,IP protocol,Socket

TABLE OF CONTENTS

1 - INTRODUCTION

The Client-Server Communication using Sockets and TCP/IP protocol is a commonly used technique in computer networking to facilitate communication between two different programs running on different machines. This project aims to develop a client-server application using C programming language, which will allow two separate programs to exchange data over a network using sockets and the TCP/IP protocol.

The client-server model is a widely used distributed application architecture that enables two or more computers to communicate over a network. In this model, one computer acts as a server that provides services, and other computers, known as clients, request these services. The communication between the client and the server is done using sockets, which are endpoints for communication between two processes on a network.

This project will use C programming language, a popular choice for system-level programming due to its low-level control and efficient memory management. The TCP/IP protocol is used for data transmission over a network, providing reliable, ordered, and error-checked delivery of data packets.

The project will involve designing and implementing a client-server application that allows the client to send requests to the server and receive responses. The client will establish a connection to the server using sockets, send data to the server, and wait for a response. The server will listen for incoming connections, receive requests from the client, process them, and send a response back.

Overall, this project will provide hands-on experience in developing a client-server application using sockets and the TCP/IP protocol in C programming language.

2 - Literature Review

Client Server Communication:

The client directly sends his request to the server asking for an appropriate service, and later server does the work and returns immediately the data or error code as a response back to the client. The server offers various services to the clients based on its client's request [1].

Sockets and Socket Programming:

In distributed computing environment, client server application (consist of client program and server program) can be designed using socket programming. Socket [2] is a two-way communication link between client and server programs that are running in a network environment. It has an effective communication mechanism between two computers.

Communication Protocol

The stream communication protocol, popularly known as TCP (transfer control protocol) is a connection-oriented protocol. This means that in order to establish communication over this type of protocol, a connection must be established first between two identified sockets [3]. One socket listens for a connection request (main responsibility of the server) and the other socket asks for a connection (main responsibility of the client) in order to establish connections. Once they have already established the connection, it can be used to transmit data in both directions [4].

3 - Project Objective

The project objectives of client-server communication using socket, TCP/IP may vary depending on the specific use case or application, but some common objectives may include:

1. Enabling real-time communication between client and server: The primary objective of client-server communication using socket and TCP/IP is to enable real-time communication between the client and server. This is useful in applications such as online gaming, chat applications, and video conferencing.

2. Data transfer: Another objective is to transfer data between the client and server securely and reliably. This can include transferring files, messages, or any other type of data.

3. Scalability: The project may aim to develop a scalable client-server architecture that can handle a large number of clients simultaneously while maintaining high performance and reliability.

4. Security: Ensuring the security of client-server communication is another objective. This can include implementing encryption to protect data in transit, authentication to ensure only authorized clients can access the server, and other security measures to prevent unauthorized access and data breaches.

5. Cross-platform compatibility: The project may aim to develop a client-server communication system that works across multiple platforms, such as Windows, macOS, Linux, and mobile devices.

Overall, the objective of client-server communication using socket and TCP/IP is to create a reliable, secure, and efficient way for clients and servers to communicate over a network, enabling real-time communication and data transfer between the two.

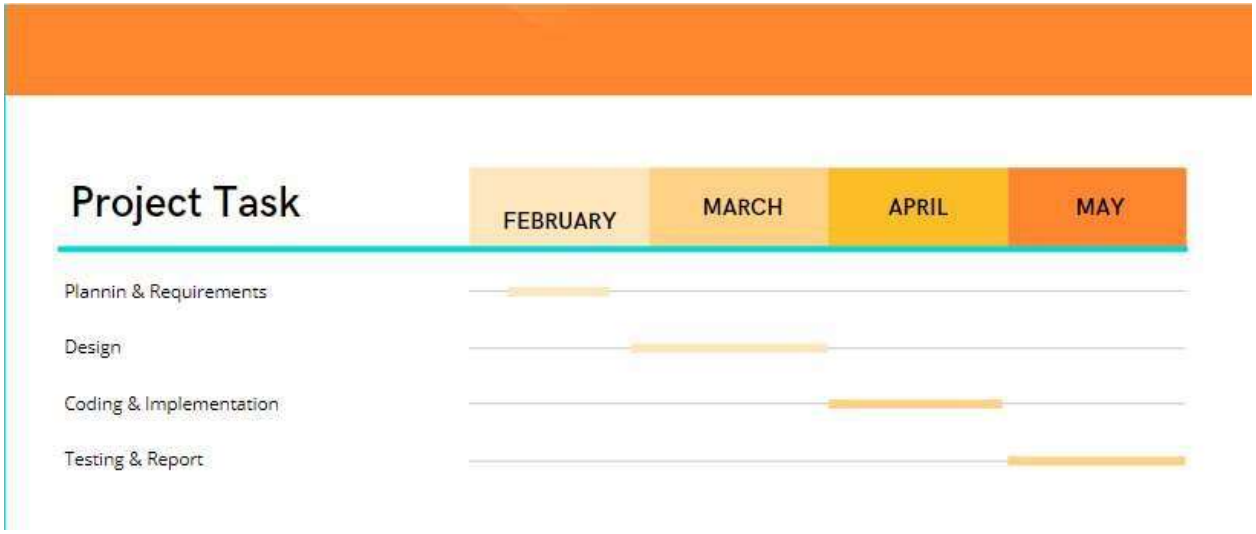
4 - Research Methodology

1. **Socket Creation Module:** This module is responsible for creating a socket on the server side and on the client side. This involves configuring the socket's protocol, address family, and other settings.
2. **Connection Module:** This module handles the connection between the client and server by establishing the socket connection, initiating the data transfer, and managing the connection lifecycle.
3. **Data Transfer Module:** This module is responsible for sending and receiving data between the client and server. It handles data serialization, buffering, and other tasks related to the transmission of data over the network.
4. **Error Handling Module:** This module is responsible for handling errors that can occur during the communication process, such as connection failures, data transmission errors, and other network-related issues.
5. **Security Module:** This module is responsible for implementing security features, such as encryption and authentication, to ensure the confidentiality and integrity of the data being transmitted.
6. **User Interface Module:** This module handles the user interface for the client application, allowing users to interact with the server and view the data being transmitted.
7. **File Transfer Module:** This module is responsible for transferring files between the client and server, handling file serialization, buffering, and other tasks related to the transmission of files over the network.
8. **Logging and Monitoring Module:** This module logs and monitors all network communication activities, providing feedback and alerts to the user or administrator in case of errors or abnormal events.

5 - Project Outcomes

1. **Reliable Communication:** TCP/IP provides reliable communication between the client and server. The protocol ensures that all data packets are transmitted in sequence and that none are lost or corrupted during transmission.
2. **Platform Independence:** The socket-based communication with TCP/IP is platform-independent. This means that the client and server can communicate irrespective of the underlying operating system.
3. **Bi-Directional Communication:** TCP/IP allows bi-directional communication between the client and server. Both parties can send and receive data.
4. **Stream-Oriented Communication:** TCP/IP provides stream-oriented communication between the client and server. This means that the data is sent as a continuous stream of bytes and is received in the same way.
5. **Security:** TCP/IP provides security through various encryption mechanisms. This ensures that the data sent between the client and server is not intercepted by unauthorized parties.
6. **Scalability:** Socket-based communication with TCP/IP is scalable. This means that the communication can be extended to include multiple clients and servers, making it suitable for large-scale applications.
7. **Interoperability:** TCP/IP provides interoperability between different systems and platforms. This means that clients and servers can communicate with each other, regardless of the hardware or software they are running.

6 - Proposed Time Duration



REFERENCES

1. Xue, M. and Zhu, C. , “The Socket Programming and Software Design for Communication Based on Client/Server”,” 2009 Pacific-Asia Conference on Circuits, Communications and Systems, pp. 774-777,2009
2. J. McManis, S. Hashmi, M Ahsan, and J. Haider, “Developing Intelligent Software Interface for Wireless Monitoring of Vehicle Speed and Management of Associated Data”, IET Wireless Sensor Systems,2016
3. IDG Communications (2017). Sockets Programming in Java World from IDG. [Online]. Available at:https://www.javaworld.com/article/2077322/core-java/core-java-sockets-programming-in-java-a-tutorial.html. [Accessed 17-July-2017]
4. T. Duong-Ba, T. Nguyen, 4B. Rose, D. Tan “Distributed Client Server Assignment for Computing, Volume 2, Issue 4, pp. 422 – 435,December 201