VIETNAM NATIONAL UNIVERSITY, HO CHI MINH CITY
HO CHI MINH UNIVERSITY OF TECHNOLOGY
COMPUTER SCIENCE & ENGINEERING DEPARTMENT

ASSIGMENT 1 REPORT

Subject: Computer Networks

CHAT APPLICATION

Supervisor:	PhD. Nguyen Le Duy Lai
Students:	
Nguyen Phuc Gia Khiem	2211573
Nguyen Truong Thai Bao	2210251

Ho Chi Minh City, November 19, 2024

CONTENTS

1 Theoretical basis **1**

1.1 Key Features of TCP/IP 1

1.2 Layers of the TCP/IP Model 1

1.2.1 Application Layer 1

1.2.2 Transport Layer 2

1.2.3 Internet Layer 2

1.2.4 Network Interface Layer 2

1.3 Key Characteristics of TCP/IP 2

1.4 Applications of TCP/IP 3

1.5 Advantages of TCP/IP 3

1.6 Limitations of TCP/IP 3

2 Chat application **4**

2.1 Description 4

2.1.1 Business context 4

2.1.2 Components 4

2.1.3 Network connection types 4

2.1.4 Functions 5

LIST OF FIGURES

Chapter 1. THEORETICAL BASIS

Introduction

The **Transmission Control Protocol/Internet Protocol (TCP/IP)** is a foundational set of protocols that govern communication across the internet and most private networks. It provides standardized methods for data transmission, ensuring reliable and efficient communication between interconnected devices.

1.1 Key Features of TCP/IP

- **Modularity**: TCP/IP is structured as a suite of protocols organized into layers, each with specific responsibilities.
- **Interoperability**: It enables communication across diverse hardware and operating systems.
- **Scalability**: Designed to accommodate networks of all sizes, from small local-area networks (LANs) to the global internet.
- **Robustness**: Ensures reliable data delivery even in challenging network conditions.

1.2 Layers of the TCP/IP Model

TCP/IP operates through a four-layer model, each responsible for specific aspects of communication:

1.2.1 Application Layer

Provides network services directly to end-users. Common protocols include:

- HTTP

- FTP
- SMTP
- DNS

1.2.2 Transport Layer

Manages end-to-end communication and data delivery. Key protocols include:

- **TCP (Transmission Control Protocol)**: Reliable, connection-oriented communication.
- **UDP (User Datagram Protocol)**: Faster, connectionless communication for applications where speed is prioritized over reliability.

1.2.3 Internet Layer

Handles addressing, routing, and packaging of data. Key protocols include:

- **IP (Internet Protocol)**: Defines logical addressing and ensures data is delivered to the correct destination.
- **ICMP (Internet Control Message Protocol)**: Reports errors and diagnostics.
- **ARP (Address Resolution Protocol)**: Maps IP addresses to physical (MAC) addresses.

1.2.4 Network Interface Layer

Handles physical transmission of data over the network medium and interfaces with network hardware like Ethernet or Wi-Fi.

1.3 Key Characteristics of TCP/IP

- **Connection-Oriented Protocol**: TCP establishes a connection before transmitting data, ensuring that packets are delivered in order and without errors.
- **IP Addressing**: Every device on a TCP/IP network has a unique IP address for identification and routing.

- **Error Handling**: TCP/IP protocols include mechanisms for error detection, correction, and retransmission.
- **End-to-End Communication**: Ensures data can traverse intermediate networks seamlessly.

1.4 Applications of TCP/IP

- **Internet Communication**: Forms the backbone of the global internet.
- **Email Transmission**: Protocols like SMTP and IMAP rely on TCP/IP.
- **File Transfers**: FTP uses TCP for secure and reliable file sharing.
- **Web Browsing**: HTTP and HTTPS depend on TCP/IP for data exchange.

1.5 Advantages of TCP/IP

- Widely adopted and universally compatible.
- Reliable communication through robust error detection and retransmission.
- Scalable for expanding networks.

1.6 Limitations of TCP/IP

- Overhead due to extensive error-checking and acknowledgment mechanisms.
- Less efficient for real-time applications where latency is critical.

Conclusion

The TCP/IP protocol suite underpins modern networking by providing a versatile and reliable framework for communication. Its modular design and compatibility with diverse systems have made it the standard for global internet and private network communications.

Chapter 2. CHAT APPLICATION

2.1 Description

The chat application is developed in C++ language using **asio.hpp** library for network connections via TCP/IP protocol.

2.1.1 Business context

This application allows multiple users access to the same chat session, this means that users can see all global messages from other users. Besides that, the application provides some features for directed messages to a specific user as well as support to create a private chat between two particular users when they send a request.

2.1.2 Components

The application has two types of components:

1. **Server**: The host machine of the chat session, always listens to the requests of the users.
2. **User**: The client machine, joins the chat session and interacts together. In private connections, a user roles both as a user and a server to communicate directly with one another.

2.1.3 Network connection types

There are two types of network connections developed in this chat application:

1. **Client - Server**: The relationship between user and server, the server always listens to the requests of the users and sends responses.
2. **Peer to Peer (P2P)**: The relationship between two users in a private connection, they can communicate together without the observation of the server, each of them is a client and a server, also.

2.1.4 Functions

Server

1. **Start**: The server can be started by using the command in the Terminal

   ```
   .\server <port>
   ```

2. **Listen to requests**: The server has a **acceptor** that allow it to always listen to the requests of the users.
3. **Response to the requests**: After receiving the requests, the server will process them and send the corresponding responses to the users.

User

In Client - Server relationship:

1. `\h`: Get the help from the server about the prefixes.
2. `\m <user> <message>`: Send the directed `message` to `user` via server, other users cannot see it.
3. `\b <message>`: Send a broadcast message that all users accessing to the server can see.
4. `\q`: Quit the server.
5. `\l`: List all current users accessing to the server.
6. `\p <user>`: Request a private connections to `user`.