UDC [004.732+004.432+004.738.5]-027.31

LAN INSTANT MESSAGE COMMUNICATION APPLICATION DESIGN BASED ON TCP

ZHANG BOWEN, ZHANG RONGLIANG, N.V. KHAJYNAVA

Belarusian State University of Informatics and Radioelectronics, Republic of Belarus

Received March 19, 2023

Abstract. The use of LAN Message Communication through TCP/IP provides a reliable, secure, and cost-effective means of communication among company or organizational staff members. This study includes the added functionality of file and picture transfers, which addresses issues related to time, cost, and information safety. The protocol offers a range of communication options, facilitating information exchange among individuals. Developed as a standalone application using C++ programming language and QT platform, the proposed protocol has been tested in the LANs of our institute.

Keywords: LAN, communication, TCP, UDP, IP.

Introduction

The evolution of communication technology has revolutionized the way we interact with each other. From the traditional modes of communication such as mail, telephone, and telegraph to the internet and digital convergence, communication has come a long way. The introduction of the internet has given rise to various means of communication, including instant messaging. In this context, the focus of this essay is on the necessity and design of a LAN instant message application based on TCP and C++. The need for such an application arises from the growing concern of data and message safety during transmission.

While conventional means of communication are still prevalent in many organizations, the advances in electronic messaging services and their synchronous nature can bring about different advantages, including improving organization activities, saving costs, energy, and safety of information. In this paper, we will explore the development of a LAN instant message application based on TCP and C++, discussing its features, advantages, and implementation. We will also examine the risks of information leakage and the necessity of such an application in today's digital age. The paper will be structured as follows: Section 1 will provide an introduction of the topic, discussing the need for secure communication tools. Section 2 of this research about the structure design of the client and server. Section 3 describes how the client connected to server. Section 4 is the project’s protocol. Section 5 gives the output of project’s implantation. Finally, it is the conclusion of the research or the project [1].

The reason – In today's interconnected world, communication is essential for the smooth functioning of any organization. However, with the increasing amount of data and messages being transmitted, the risk of information leakage has also grown exponentially. In recent times, we have witnessed several high-profile cases of data breaches, where sensitive information has been compromised, resulting in significant losses for companies and individuals.

For instance, in 2017, Equifax, one of the largest credit reporting agencies in the US, suffered a data breach that exposed the personal information of 147 million people. This breach was caused by a vulnerability in the company's web application, which hackers exploited to gain unauthorized access to sensitive data. The breach resulted in the loss of sensitive personal and financial information, including names, social security numbers, and credit card numbers. Similarly, in 2020, the video conferencing app, Zoom, faced severe backlash over privacy and security concerns. The app's popularity skyrocketed due to the COVID-19 pandemic, but it soon became evident that the app was not adequately secured, and data breaches were rampant. Zoom's user data was sold on the dark web, and users' personal information was leaked, leading to severe consequences for individuals and companies alike. In such a scenario, the need for secure communication tools cannot be overemphasized. This is where a LAN instant message application based on TCP using C++ comes into play. Such an application can help organizations communicate securely within their internal network, minimizing the risk of data breaches and information leakage.

In the wake of this, it is an urge to use a safer communication way to transmit information. That is the reason why we design this LAN instant message communication application.

Server-Client Structure Design

To realize the program, Client/Server model is adapted. In the client/server programming model, a server program awaits and fulfills requests from client programs, which might be running in the same or other computers.

In this structure (Figure 1), server must be running first and waiting (listening) for a connection, server listening at specific port number and create thread for each client connection. Each client has the same port number that the server listening at it and hostname of server (or IP).

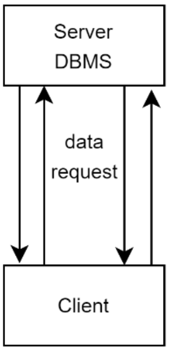

Figure 1. Structure of C/S

However, technically in the Client/Server model, a client cannot reach other clients directly. In other words, a client cannot send data to other clients directly. As Figure1 shown, a client can send data to another client or clients by send the request to server and then server send the data to the destination or target clients. How this information transmission works between server and client is based on TCP socket. Once the connection is established, data can be transmitted.

Another question arise – how do server and clients know what kind of information or action should server to respond to clients. That is the key point of this structure. Every Clients connected to the server have a matched socket to server and every operation made by clients will send an encode to server. Then server will get the request and give a corresponding response.

For instance, a client sends a message to server. The client sends an encode pattern like "@sendPersonalMesssage#ToFriendOne#content to server". Then the server will according the prefix of the encode "@sendPersonalMessage" to store the message in the database and simultaneously send an encode pattern like "@sendPersonalMessage#From#FriendTwo#content" to the target client. With the help of this encode pattern, TCP and Signal-emit from C++, communication is accomplished perfectly.

Database Design

As shown at the Figure 2, "many to many" relation is between entity user and group, which means a user can in many groups while a group can have many users.

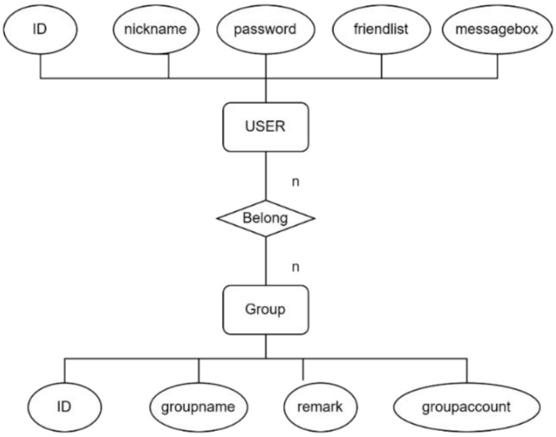

Figure 2. User and group

As shown at the Figure 3, “one to many” relation is between entity user and friend, which means a user, can have many friends. Each user has a friend table for its own and it has all friends contact.

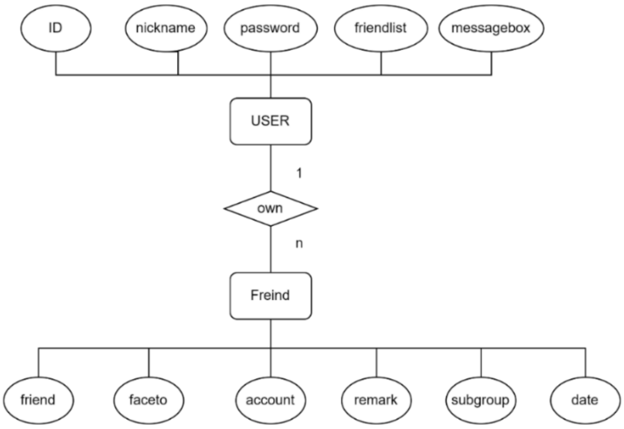

Figure 3. User and friend

As shown at the Figure 4, “one to many” relation is between entity user and message, which means a user, can have many messages from friends or groups. Each user has a messagebox table for its own, it has all messages from friends, and groups send to the user.

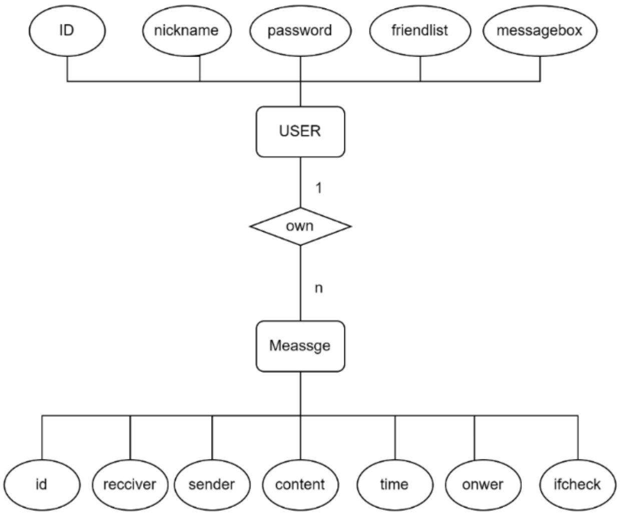

Figure 4. User and message

Comparative analysis of TCP and UDP

Two Internet Protocols can be chosen: TCP and UDP. After the comparative analysis, the TCP is used to implement data transmission.

TCP is connection-oriented while UDP is connectionless. For TCP, the client sends a synchronization request, the server sends back an acknowledgment, and the client returns a synchronization acknowledgment in response. Comparatively, for UDP, the message is sent out, without as much regard for the recipient, without considering the destination. Connectionless transport protocols can lose a minimal number of packets.

TCP sends data in a particular sequence, whereas there is no fixed order for UDP protocol. TCP send and receive data with sequence numbers. This allows the system to track the specific order in which data is transmitted, maintaining the desired sequence. On the contrary, UDP does not follow a sequencing mechanism. The protocol has no way of telling which data packets should come first, and if they are received in the wrong order. UDP also drops any data packet that it is unable to process. However, UDP is faster and more efficient than TCP User datagram protocol does not need an established connection to start sending packets. Therefore, it saves the time typically required to turn on the server and place it in a “passive open,” listening state. It allows data transmission to begin faster immediately or extended latency time. There is also no need to put the packets in sequence or send and receive acknowledgments, saving time. UDP is also more efficient in terms of bandwidth. Once the data is in motion from the server to the client, TCP engages in many error check mechanisms, acknowledgment processes, and sequencing measures, which occupy a lot of bandwidth. In contrast, UDP quickly gets the data stream from one computing location to another without a lot of checks and balances. UDP is suitable for live and real-time data transmission, which TCP cannot support.

Despite its inherently unreliable nature, UDP continues to be a staple for online operations. This is because it is ideal for real-time data transmissions, where the loss of a few packets does not matter.

TCP is best for use cases where data integrity matters more than transmission speed. It will ensure that files and web pages arrive intact and can even be helpful for real-time analytics and content delivery networks, where dropped packets would fudge the outcomes. In comparison, UDP is suitable for media transmissions, such as video calling and online gaming [2].

Through what has been mentioned above, TCP is more reliable but less efficient and slower in transmitting. For Lan Instant Message Communication application, files and chat messages are the content that being transmitted. The transmission is not strictly real-time. Data integrity should come first. Therefore, it considers more reliability than efficiency. In this paper, TCP is used for the Internet Protocol of the application.

Conclusion

The proposed TCP protocol solved the instant communication problems among staff members and reduces time of communication at low cost. Simultaneously the LAN instant message communication application can guarantee the safety of the data transmission since it is not connected to the network. However, data leakage is still at risk, since Information leakage channels exist in any information space. A leakage channel in the most general sense is understood as an uncontrolled way of transmitting information. As a result, an attacker can gain unauthorized access to the confidential company data he needs. However, physical access by plugging unknown external devices still may cause a leakage. Therefore, organization application regulations should be combined with the application to reduce the risk of leakage.

References

1. Design and implement chat program using TCP/IP Mohammed A. Ahmed, Sara Ammar Rafea, Lara Moufaq Falah, Liqaa Samir Abd Ullah

2. TCP and UDP: Understanding 10 Key Differences Chiradeep BasuMallick [Electronic resource]. URL: https://www.spiceworks.com/tech/networking/articles/tcp-vs-udp/#_003