Design and Implementation of a LAN Chat System Based on C Language

Zhong Wang, Huaqiong Duan

Jincheng College of Sichuan University, School of Computer and Software Engineering, Chengdu 611731, Sichuan, China
**Correspondence Author*

Abstract: ***The 21st century is the era of networks; the popularization of the Internet has fundamentally transformed traditional modes of communication. LAN chat systems are a product of computer networks reaching a certain stage of development. This paper first describes the background of LAN chat system design, the current state of research, the tools and related technologies employed to implement the system, then presents a detailed account of the overall system design, and briefly introduces the implementation principles of several key modules.***

Keywords: LAN; Instant messaging; Modularity; Multithreading; C language.

1. INTRODUCTION

Since the beginning of the 21st century, the accelerating global informatization process has driven rapid advances in computer network technology, spawning a continuous stream of network applications—such as the increasingly popular Internet-based online transaction processing, mobile communications, and artificial intelligence. LAN chat technology has likewise developed swiftly. To handle internal corporate affairs more efficiently and to better safeguard information security, more and more enterprises are now focusing on building internal LAN systems. Thanks to these internal LANs, companies can streamline information flows and accelerate internal information sharing, thereby improving work efficiency. However, as enterprises grow, their business volume increases; if the company has not anticipated this in advance, the original LAN system may become overburdened, leading to information bottlenecks. In such cases, the LAN itself can become an obstacle to corporate development.

To resolve the above contradiction, many professionals have proposed solutions—for example, increasing network bandwidth and boosting server throughput. Yet, from a cost perspective, these measures require secure and reliable software support; otherwise, they not only fail to solve the problem but also impose additional burdens on the enterprise. In light of these issues, this paper presents a design and implementation method for a LAN communication software based on the C/S architecture, employing C language combined with multithreading technology, thereby establishing a secure and efficient communication mechanism for enterprises.

Jiang et al. (2025) introduced Investment Advisory Robotics 2.0, leveraging deep neural networks to provide personalized financial guidance, showcasing the potential of AI in finance [1]. Wang (2025) proposed a method for joint training of propensity and prediction models via targeted learning, addressing the challenge of recommendation systems dealing with data missing not at random [2]. Ding and Wu (2024) conducted a systematic review on self-supervised learning for biomedical signal processing, focusing on ECG and PPG signals, which contributes significantly to the healthcare domain [3]. Restrepo et al. (2024) explored multimodal deep learning in low-resource settings, employing a vector embedding alignment approach for healthcare applications, highlighting the importance of adaptable AI solutions [4]. Xie and Chen (2025) developed InVis, an interactive neural visualization system for human-centered data interpretation, enhancing the accessibility and interpretability of complex data [5]. Their subsequent work, CoreViz, introduced a context-aware reasoning and visualization engine for business intelligence dashboards, further advancing data visualization techniques [6]. Zhu (2025) proposed TraceLM, a framework for temporal root-cause analysis using contextual embedding language models, providing insights into temporal data patterns [7]. Zhang (2025) introduced CrossPlatformStack, enabling high availability and safe deployment for products across meta services, addressing scalability and reliability challenges [8]. Hu (2025) presented GenPlayAds, a procedural playable 3D ad creation system via generative models, showcasing the creative potential of AI in advertising [9]. Wang and Shih (2024) proposed a hybrid multi-modal recommendation system integrating MMOE and XGBoost, enhancing personalization and accuracy in recommendation tasks [10]. Fu et al. (2025) explored adversarial prompt optimization in LLMs, presenting HijackNet's approach to improving robustness and defense evasion capabilities [11]. Zheng et al. (2025) introduced FinGPT-Agent, an advanced framework for multimodal research report generation, incorporating

task-adaptive optimization and hierarchical attention mechanisms [12]. Weng et al. (2025) developed SafeGen-X, a comprehensive framework aimed at enhancing security, compliance, and robustness in large language models [13]. Chen et al. (2025) introduced SyntheClean, a method for enhancing large-scale multimodal models through adaptive data synthesis and cleaning [14]. Jiang et al. (2025) developed a knowledge-enhanced multi-task learning model for domain-specific question answering, demonstrating the effectiveness of integrating external knowledge sources [15]. Zhuo et al. (2025) proposed an intelligent-aware Transformer with domain adaptation and contextual reasoning capabilities for question answering tasks [16]. Zhang et al. (2025) explored dynamic attention-guided video generation from text, utilizing multi-scale synthesis and LoRA optimization techniques [17]. Zhao et al. (2025) introduced KET-GPT, a modular framework for precision knowledge updates in pretrained language models [18]. Shih et al. (2025) developed DST-GFN, a dual-stage Transformer network with gated fusion for pairwise user preference prediction in dialogue systems [19]. Li et al. (2025) proposed MLIF-Net, a multimodal fusion approach combining vision transformers and large language models for AI image detection [20]. Chen et al. (2024) introduced Bimcv-r, a landmark dataset for 3D CT text-image retrieval, providing a valuable resource for medical image analysis [21]. Sun et al. (2025) focused on constructing an Automated Machine Learning (AutoML) framework based on large language models, aiming to streamline the machine learning pipeline [22]. Pal et al. (2025) proposed an AI-based credit risk assessment and intelligent matching mechanism in supply chain finance, enhancing financial decision-making processes [23]. These studies collectively contribute to the advancement of AI technologies across various domains, highlighting the diversity and depth of current research efforts.

2. THE CONCEPT OF LAN CHAT

LAN chat software, abbreviated as LCS (LAN chat software), is a network communication tool service aimed at end users within an enterprise LAN. Users can engage in real-time two-way or multi-party communication via terminals equipped with instant messaging. Communication content includes text, interfaces, voice and video, and file transfers. Several LAN chat software products targeting enterprises have already appeared on the Chinese market, including Tencent's RTX, IBM's Lotus Sametime, and Tongruanlianhe's GoCom. Among these, WiseUC by China Yingtong, imo, Chuanwo, Shiyexian, and Feiqiu/Feige are available free of charge.

The name Local Area Network (LAN) itself implies its geographically limited scope, generally covering only a few kilometers. In an era when information security is so highly valued, data exchange over a LAN is more secure, and the network is highly specialized, featuring a stable and standardized topology. Because of its limited coverage, a LAN typically offers much higher transmission speeds than a Wide Area Network (WAN).

3. CURRENT RESEARCH AT HOME AND ABROAD

Starting in the 1990s, instant messaging software began to emerge. Entering the 21st century, with the development of mobile phones—especially the advent and popularization of smartphones led by Android—network communication shifted from PCs to mobile clients.

LAN-based instant messaging tools are essentially scaled-down versions of Internet instant messengers. Early foreign products such as ICQ, MSN, and Yahoo Messenger are examples. When ICQ was all the rage, domestic imitators appeared, including Tencent QQ and Sina UC. These chat programs implemented sophisticated features and supported large numbers of concurrent users through multithreading and multi-server technologies. Today's WAN instant messengers generally adopt UDP or TCP protocol suites, communicate via TCP/IP, and have reached technological maturity. Their evolution has greatly expanded the concept of chat tools and is expected to become the main platform for people to connect and entertain over the Internet.

As the world's first instant messaging tool, ICQ offers more extensive functionality than Tencent QQ or MSN, yet its excessive complexity led to user attrition. Today ICQ holds a certain share of the foreign market but relatively little in China. Enterprise instant messaging, aimed at corporate end-users, provides services such as rich-media chat, message push, and OA approval. Stripped of entertainment elements, it emphasizes practicality, security, stability, and scalability. Although a LAN's structure is more complex than a WAN's, leveraging certain protocols and communication tools makes its implementation simpler, so research into LAN-based instant messaging remains highly promising.

4. KEY TECHNOLOGIES

4.1 C Language

C is a computer programming language that combines the features of both high-level and assembly languages. It can serve as a system-design language for writing system applications, or as an application-design language for writing programs independent of computer hardware. As one of today's mainstream programming languages, C has long been the essential first language for many programming enthusiasts and professionals, and numerous modern languages are developed on the basis of C. At present, C is used in a wide range of fields—not only in software development but also in various scientific research projects, such as server-side development, system software, graphics processing, microcontroller and embedded-device development. In short, C can be applied to virtually every area of program development.

4.2 Multithreading Technology

Multithreading technology refers to the technique of executing multiple threads concurrently through software or hardware. Computers with multithreading capability can execute several threads at the same time thanks to hardware support, thereby improving overall processing performance. This involves understanding the relationship between processes and threads—both are operating-system concepts. A process is, in simple terms, a running program, whereas a thread is merely an execution unit within that process; their relationship is somewhat like containment. Systems capable of multithreading include symmetric multiprocessors, multi-core processors, and processors supporting chip-level multithreading or simultaneous multithreading. The concept of programming with these independently running program fragments (threads) within a single program is called "multithreading."

5. SYSTEM REQUIREMENTS ANALYSIS AND HIGH-LEVEL DESIGN

5.1 Requirements Analysis

The LAN chat software is designed from the perspectives of security and usability. It not only satisfies practical chat requirements but also implements the functions the system needs, reducing the problems of blocked or insecure information exchange caused by a company's continuous growth, thereby preventing unnecessary losses.

In terms of system performance, the client and server must maintain a stable connection at all times. Unlike ordinary web browsing, the interaction between client and server is not a simple request-response sequence but a bidirectional relationship: while the client sends messages to the server, the server simultaneously pushes chat messages from others to the client. Therefore, a stable connection must be established to avoid message loss or delayed reception.

Regarding speed performance, the program must be lightweight and fast—low hardware requirements, low memory usage, and rapid response. Even when the client sends a large volume of connection requests at once, the server must still maintain a high-performance concurrent processing mechanism to prevent errors such as message congestion that could affect system operation.

Of course, robust fault tolerance and a clear modular design are also indispensable. Good fault tolerance ensures that users can exit the system gracefully in case of incorrect operations or system errors, while a clear modular design makes future maintenance and extension faster and easier.

5.2 Implementation Content

1) Start the server: The server must be started before users can register, log in, or use the chat features.

2) User login: Users with an existing username and password can log in directly; otherwise, they must register and complete their personal information first. After logging in, the server will display the user; when the user wants to log in again, they send a request to the server and, after passing login verification, can log in once more.

3) User registration: First-time users must complete their personal information to register an account; after successful registration, they can return to the login screen and log in.

4) User offline: To go offline, the user simply closes the window; the background simultaneously executes an information security protection mechanism to prevent personal data leakage.

5) Send chat messages: Once in the chat interface, users can start chatting. There are two modes: group chat and private chat (only the designated user receives the message). Both modes are more secure than typical WAN-based chat applications.

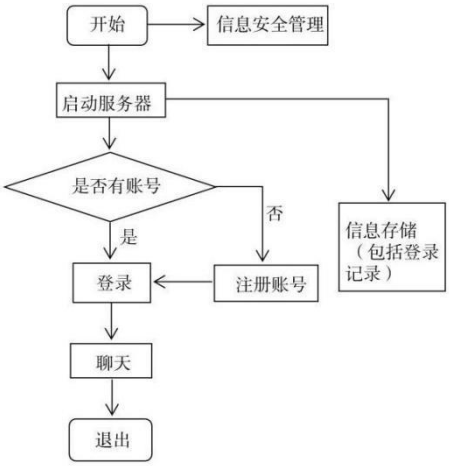

Figure 1: Program Flowchart

5.3 Server-Side Overview Design

This chat tool uses a C/S architecture: the client can only start after the server is launched. Therefore, the server must handle the following functions.

- Process user registration requests
- Process user login requests
- Handle user logout requests
- Handle user connection requests
- Store communication data
- Information security assurance
- Display the above current user information

5.4 Client Overview Design

Functions the client needs to implement:

- User login interface implementation
- User registration interface implementation
- User chat interface implementation
- Group chat interface implementation
- Private chat interface implementation

5.5 Module Organization

The entire LAN chat system mainly includes the following modules: registration module, login module, logout module, chat module, server module, storage module, and information security module. Among them, the chat module includes the group chat module and the private chat module. The calling relationships among the modules are shown in Figure 2.

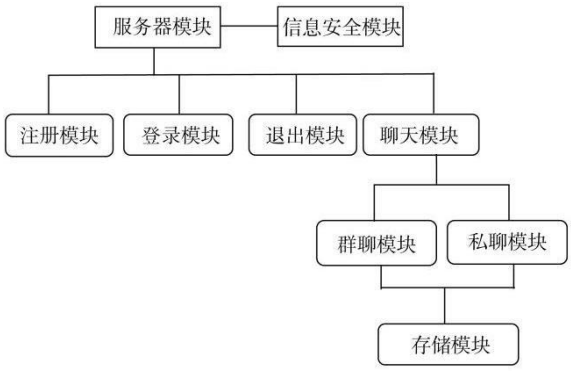

Figure 2: Software Module Organization Diagram

6. CORE MODULE IMPLEMENTATION PRINCIPLES

6.1 Implementation of the Server Module

To implement a LAN chat system, the server must first be set up; this requires the following five steps:

① Create a socket.
② Bind an address to the INET BSD socket.
③ Bind the socket to the address structure.
④ Listen on the INET BSD socket.
⑤ Use accept to receive connection requests from clients.

Once the server is set up, whenever it receives data from a client, it automatically spawns a thread to handle it; the server selects the appropriate function according to the protocol carried in the structure sent by the client.

6.2 Implementation of the Chat Module

The client sends data to the server in the form @username:, after which the server parses the data. Based on the #who and #chat protocols, it determines whether the message is for private chat, then extracts the target username, adds a timestamp, and forwards the message to that user.

7. CONCLUSION

As operating systems become increasingly bloated and complex, and application software ever richer and more powerful, chat systems will inevitably demand higher CPU computing power. Multi-threaded (multi-core) CPUs can decompose large, complex computational tasks into smaller ones and assign them to different cores (threads) for parallel processing, ensuring information security while greatly reducing the computational load on each thread and improving efficiency. In today's era of rapid global economic and Internet development, a simple yet effective LAN communication system has great practical value and development prospects; therefore, we should devote greater attention to its research and development in the future.

REFERENCES

[1] Jiang, G., Yang, J., Zhao, S., Chen, H., Zhong, Y., & Gong, C. (2025). Investment Advisory Robotics 2.0: Leveraging Deep Neural Networks for Personalized Financial Guidance. Preprints. https://doi.org/10.20944/preprints202504.1735.v1
[2] Wang, Hao. "Joint Training of Propensity Model and Prediction Model via Targeted Learning for Recommendation on Data Missing Not at Random." AAAI 2025 Workshop on Artificial Intelligence with Causal Techniques. 2025.
[3] Ding, C.; Wu, C. Self-Supervised Learning for Biomedical Signal Processing: A Systematic Review on ECG and PPG Signals. medRxiv 2024.

[4] D. Restrepo, C. Wu, S.A. Cajas, L.F. Nakayama, L.A. Celi, D.M. López. Multimodal deep learning for low-resource settings: A vector embedding alignment approach for healthcare applications. (2024), 10.1101/2024.06.03.24308401
[5] Xie, Minhui, and Shujian Chen. "InVis: Interactive Neural Visualization System for Human-Centered Data Interpretation." Authorea Preprints (2025).
[6] Xie, Minhui, and Shujian Chen. "CoreViz: Context-Aware Reasoning and Visualization Engine for Business Intelligence Dashboards." Authorea Preprints (2025).
[7] Zhu, Bingxin. "TraceLM: Temporal Root-Cause Analysis with Contextual Embedding Language Models." (2025).
[8] Zhang, Yuhan. "CrossPlatformStack: Enabling High Availability and Safe Deployment for Products Across Meta Services." (2025).
[9] Hu, Xiao. "GenPlayAds: Procedural Playable 3D Ad Creation via Generative Model." (2025).
[10] Wang, Yang, and Kowei Shih. "Hybrid multi-modal recommendation system: Integrating mmoe and xgboost for enhanced personalization and accuracy." 2024 4th International Conference on Artificial Intelligence, Robotics, and Communication (ICAIRC). IEEE, 2024.
[11] Fu, Lei, et al. "Adversarial Prompt Optimization in LLMs: HijackNet's Approach to Robustness and Defense Evasion." 2025 4th International Symposium on Computer Applications and Information Technology (ISCAIT). IEEE, 2025.
[12] Zheng, Haoran, et al. "FinGPT-Agent: An Advanced Framework for Multimodal Research Report Generation with Task-Adaptive Optimization and Hierarchical Attention." (2025).
[13] Weng, Yijie, et al. "SafeGen-X: A Comprehensive Framework for Enhancing Security, Compliance, and Robustness in Large Language Models." 2025 8th International Conference on Advanced Algorithms and Control Engineering (ICAACE). IEEE, 2025.
[14] Chen, Yang, et al. "SyntheClean: Enhancing Large-Scale Multimodal Models via Adaptive Data Synthesis and Cleaning." 2025 5th International Conference on Artificial Intelligence and Industrial Technology Applications (AIITA). IEEE, 2025.
[15] Jiang, Gaozhe, et al. "A Knowledge-Enhanced Multi-Task Learning Model for Domain-Specific Question Answering." 2025 7th International Conference on Information Science, Electrical and Automation Engineering (ISEAE). IEEE, 2025.
[16] Zhuo, Jiayang, et al. "An Intelligent-Aware Transformer with Domain Adaptation and Contextual Reasoning for Question Answering." 2025 IEEE 6th International Seminar on Artificial Intelligence, Networking and Information Technology (AINIT). IEEE, 2025.
[17] Zhang, Hanlu, et al. "Dynamic Attention-Guided Video Generation from Text with Multi-Scale Synthesis and LoRA Optimization." 2025 4th International Symposium on Computer Applications and Information Technology (ISCAIT). IEEE, 2025.
[18] Zhao, Shihao, et al. "KET-GPT: A Modular Framework for Precision Knowledge Updates in Pretrained Language Models." 2025 IEEE 6th International Seminar on Artificial Intelligence, Networking and Information Technology (AINIT). IEEE, 2025.
[19] Shih, Kowei, et al. "DST-GFN: A Dual-Stage Transformer Network with Gated Fusion for Pairwise User Preference Prediction in Dialogue Systems." 2025 8th International Conference on Advanced Electronic Materials, Computers and Software Engineering (AEMCSE). IEEE, 2025.
[20] Li, Xuan, et al. "MLIF-Net: Multimodal Fusion of Vision Transformers and Large Language Models for AI Image Detection." 2025 8th International Conference on Advanced Algorithms and Control Engineering (ICAACE). IEEE, 2025.
[21] Chen, Yinda, et al. "Bimcv-r: A landmark dataset for 3d ct text-image retrieval." International Conference on Medical Image Computing and Computer-Assisted Intervention. Cham: Springer Nature Switzerland, 2024.
[22] Sun, N., Yu, Z., Jiang, N., & Wang, Y. (2025). Construction of Automated Machine Learning (AutoML) Framework Based on Large LanguageModels.
[23] Pal, P. et al. 2025. AI-Based Credit Risk Assessment and Intelligent Matching Mechanism in Supply Chain Finance. Journal of Theory and Practice in Economics and Management. 2, 3 (May 2025), 1–9.