Optimising Peer-to-Peer network Communication: Reilable UDP for Improved Latency

Nilesh Jadhav
Computer Engineering
Vishwakarma institute of Technology
Pune, India
nilesh.jadhav22@vit.edu

Krishna Nilesh Jaiswal
Artificial Intelligence and Data Science
Vishwakarma institute of Technology
Pune, India
krishna.jaiswal22@vit.edu

Ravindra Jadhav
Electronics and Telecommunicarion
Vishwakarma institute of Technology
Pune, India
ravindra.jadhav22@vit.edu

Sumedh Kokane
Computer Engineering
Vishwakarma institute of Technology
Pune, India
sumedh.kokane22@vit.edu

Viomesh Singh
Artificial Intelligence and Data Science
Vishwakarma institute of Technology
Pune, India
viomesh.singh@vit.edu

***Abstract*-Effective communication in network games is essential to mitigate latency issues, thereby enhancing user satisfaction. Traditional solutions often entail costly hardware upgrades or internet speed enhancements, placing financial strain on users. To address this challenge, we propose leveraging a reliable user datagram protocol (RUDP) as an alternative to conventional transmission control protocol (TCP) or user datagram protocol (UDP) systems. Despite RUDP's recognized potential, the scarcity of large-scale testbeds impedes direct experimentation. Thus, our research adopts a small-scale testbed to evaluate performance and develop an RUDP-based communication model. Using the Java tool, we analyze the model's efficacy in improving network communication performance. This approach not only aims to optimize communication experience but also considers energy and cost-effective photonics technologies for access/metro networks, thereby offering a promising avenue for addressing latency issues in network communication.**

Keywords— RUDP, UDP TCP, Java, Communication Model.

I. Introduction

Peer-to-Peer (P2P) systems signify a fundamental departure from conventional centralized computing models, introducing a decentralized framework where every node holds equal importance. This departure challenges the traditional client-server architecture by dispersing control and functionality across the network. Unlike centralized systems where a single entity orchestrates resource allocation and management, P2P networks distribute these responsibilities among all participating nodes[1].

At the heart of P2P systems lies the concept of collaborative resource sharing, where each node acts both as a consumer and a provider of resources. Whether it's file sharing, computational power, or content delivery, P2P networks harness the collective capabilities of all connected devices[2]. This approach not only enhances resource utilization but also fosters resilience and scalability, as the network's performance isn't reliant on the availability of a single centralized entity. The decentralized nature of P2P systems promotes inclusivity, accommodating nodes of varying types and sizes. Whether it's a household computer or an enterprise-level server, each node contributes to the network's overall functionality without the need for hierarchical organization[3]. This egalitarian structure ensures that no single node holds undue influence over others, fostering a collaborative and cooperative environment where all participants are valued equally[4].

In essence, P2P systems redefine the traditional notions of computing by distributing control, functionality, and resources across a network of peers[5]. By embracing decentralization and equal participation, these systems offer a flexible and robust framework for resource sharing and critical functions, challenging the dominance of centralized models in distributed computing[6].

P2P computing has stirred considerable debate within the technological community, with differing perspectives on its novelty and significance. Definitions vary, but generally, P2P systems enable direct exchange of files, resources, and services between systems, often operating on the periphery of the Internet. Key requirements include operational quality computing devices, independent addressing systems, and adaptability to variable connectivity[7]. At its essence, Peer-to-Peer (P2P) networking harnesses the diverse resources located at the periphery of the Internet. These resources span a spectrum, including storage capacities, computational capabilities, digital content, and even the human presence itself. By tapping into these distributed resources, P2P systems create a decentralized ecosystem where users can seamlessly share and access information without the need for central oversight [8].

However, operating within such decentralized environments poses several challenges. One significant hurdle is the inherent instability of connectivity. Unlike traditional client-server models where communication is routed through centralized servers with stable connections, P2P networks rely on direct peer-to-peer connections. This can result in intermittent connectivity issues, especially in scenarios where nodes enter or leave the network dynamically [9].

The reliance on peer-to-peer connections leads to the issue of unpredictable IP addresses. In P2P networks, each node typically communicates directly with others, bypassing the need for centralized routing systems. Consequently, IP addresses may change frequently, making it challenging to establish and maintain connections reliably [10]. Despite these challenges, P2P nodes operate autonomously, free from the constraints of centralized control. By circumventing the Domain Name System (DNS) and reducing dependence on central servers, P2P networks foster a distributed infrastructure where each node contributes to the overall functionality [11]. This autonomy not only enhances resilience against single points of failure but also promotes scalability and efficiency by distributing tasks across the network. This paper aims to provide a comprehensive overview of P2P networks, encompassing various classifications, representative examples, platforms, applications, security concerns, and simulation tools [12]. By delving into the intricacies of P2P systems, we aim to deepen understanding and foster discussion around this evolving and dynamic field of distributed computing.

II. Literature Review

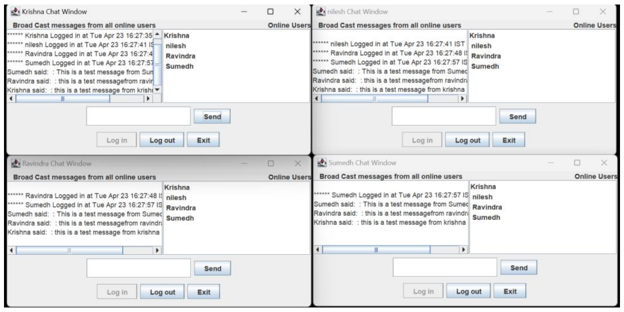
Over the past few years, there has been a noticeable decline in the popularity of multiplayer network-based games, with gamers increasingly gravitating towards massive multiplayer online games (MMPOG)/MMORPG. Griwodz Carsten proposed a proxy-based gaming architecture aimed at categorizing gaming traffic according to its urgency or relevance, as a means to address scalability issues. Similarly, another architecture was introduced with the objective of alleviating traffic congestion on central game servers. Studies have compared the average load on servers based on the number of users involved, suggesting that peer-to-peer architecture can effectively reduce server load, resulting in lower latency compared to fully centralized server architectures. However, despite its benefits, peer-to-peer architecture may struggle with resolving issues related to state inconsistency[13]. In response, the concept of a central arbiter architecture, blending aspects of both peer-to-peer and centralized architectures, was proposed to leverage the strengths of each approach. Over the past decade, numerous studies have focused on identifying drawbacks such as jittering, delays, and packet losses in multiplayer and wireless games through simulation-based investigations.

One recent survey [14] clearly shows that network latency is one of the most critical elements relevant to user satisfaction in wireless games. network latency is one of the most critical elements relevant to user satisfaction in wireless games. An improved Quality of Service (QoS) means a better game performance based on descriptions [15] where it is been proven that the Universal Mobile Telecommunications System (UMTS) exhibits a better performance level in interactive real-time games than that of General Packet Radio Service (GPRS) as it, on some level, surmounts the problems of overprovisioning which often cause delays and jittering during the game process. It was suggested that statistical multiplexing and QoS together ensure an improved multiple game flow. More wireless gaming architectures have been introduced in recent years . However, some of the present global leading wireless game providers are focusing on implementing their services on Enhanced General Packet Radio Service (EGPRS) that could be well-suited for the cross-layer design proposed in this study. Producing or launching wireless games with a reduced overhead is one of the major considerations which game providers should not take lightly as it would be costly to change the existing protocol stack for the intermediate nodes not connected to the game being played. Currently, either UDP or RUDP are often used for games [19], with the latter usually playing the role of continuously delivering the status of the current gaming information and also being suitable for lightweight game packet delivery. On the basis that TCP is a relatively slow protocol because of its complex congestion control algorithms and byte-oriented window scheme, this is considered a protocol that is not suited for wireless online games that accommodate multiple numbers of users simultaneously without causing delays.

In addition, TCP cannot even deal with moderate traffic congestion when delivering packets and this makes it quite clear that it is not at all suitable for MMPOG games. The connectionless protocol UDP, however, operates on the top of the IP layer and exchange data directly and recovers data when necessary, which seldom happens. As both are universal protocols widely adopted for most network devices and equipment, they are quite convenient for application developers. RUDP supplements UDP's error recovery function through a one-time acknowledgment scheme. Currently, both UDP and RUDP have become a major default protocol for wireless gaming infrastructures . Other protocols including the game transport protocol (GTP) have not been successful as their commercialization was always questioned. In the final analysis, it is clear that existing end-to-end QoS approaches for multimedia traffic do not work well with wireless games. As traffic generated during wireless games can be quite heavy depending on the number of participants and indicate an extreme value distribution , some new approaches that can effectively and efficiently deal with QoS-related issues should be devised to handle traffic [16].

III. Methodology

This project demonstrates the implementation of a UDP peer-to-peer system, allowing users to securely communicate and exchange data. At the core of this system, users are required to log in to access its features. To facilitate this process, users are prompted to input their credentials, namely their username and password. This login mechanism ensures that only authenticated users can access the system, enhancing security and privacy. In our UDP peer-to-peer project, once users are online and visible to each other, any user can broadcast a message to all others. Every client can see this message, including information about who sent the message and at what timestamp as shown in Figure 1. This process is designed to achieve low latency, ensuring smooth and real-time communication among peers.



```
****** Krishna Logged in at Tue Apr 23 16:27:35
****** nilesh Logged in at Tue Apr 23 16:27:41 IS
****** Ravindra Logged in at Tue Apr 23 16:27:4
****** Sumedh Logged in at Tue Apr 23 16:27:57
Sumedh said:  : This is a test message from Sun
Ravindra said:  : this is a test messagefrom ravir
Krishna said:  : this is a test message from krishi
```

Krishna
nilesh
Ravindra
Sumedh

```
****** nilesh Logged in at Tue Apr 23 16:27:41 IST
****** Ravindra Logged in at Tue Apr 23 16:27:48 IS
****** Sumedh Logged in at Tue Apr 23 16:27:57 IS
Sumedh said:  : This is a test message from Sumec
Ravindra said:  : this is a test messagefrom ravindr
Krishna said:  : this is a test message from krishna
```

Krishna
nilesh
Ravindra
Sumedh

```
****** Ravindra Logged in at Tue Apr 23 16:27:48 IS
****** Sumedh Logged in at Tue Apr 23 16:27:57 IS
Sumedh said:  : This is a test message from Sumec
Ravindra said:  : this is a test messagefrom ravindr
Krishna said:  : this is a test message from krishna
```

Krishna
nilesh
Ravindra
Sumedh

```
****** Sumedh Logged in at Tue Apr 23 16:27:57 IS
Sumedh said:  : This is a test message from Sumec
Ravindra said:  : this is a test messagefrom ravindr
Krishna said:  : this is a test message from krishna
```

Krishna
nilesh
Ravindra
Sumedh

Fig. 1: Implementation of Project

The broadcasting mechanism in our peer-to-peer system allows any user to send a message to all other connected users simultaneously. This is particularly important for real-time communication and maintaining an engaging user experience. When a user sends a message, it is broadcasted to all other users, ensuring everyone stays updated and connected. User-friendly buttons for "Log in" and "Exit" functions are provided, streamlining the user experience. This design enables smooth navigation within the application, making it intuitive and straightforward for users to interact with. By integrating these features, we aim to enhance both the usability and security of our application, providing a seamless experience for users.

Once logged in by entering the username and password, the program sends a notification to all users, indicating that a particular user with a specific nickname is now online. This notification ensures that all users are informed about the active users. Upon receiving the notification, all users can see the list of active users. This feature enhances the interactivity of the system and facilitates communication between users. To implement this feature, the system maintains an updated list of active users. When a user logs in, the system broadcasts a message to all connected peers, informing them about the newly logged-in user. This message includes the nickname of the user who just logged in. By providing this feature, our application creates a dynamic and engaging environment, allowing users to easily identify and interact with other active users. It enhances the overall experience of the peer-to-peer system and fosters seamless communication among users. Once logged in by entering the username and password, the program sends a notification to all users, indicating that a particular user with a specific nickname is now online. This notification ensures that all users are informed about the active users. Upon receiving the notification, all users can see the list of active users. This feature enhances the interactivity of the system and facilitates communication between users. To implement this feature, the system maintains an updated list of active users. When a user logs in, the system broadcasts a message to all connected peers, informing them about the newly logged-in user. This message includes the nickname of the user who just logged in. By providing this feature, our application creates a dynamic and engaging environment, allowing users to easily identify and interact with other active users. It enhances the overall experience of the peer-to-peer system and fosters seamless communication among users.

Broadcasting is a network communication method where a single sender transmits data to all other devices or nodes in the network. It's particularly useful in scenarios where the same information needs to be shared with multiple recipients simultaneously. In the context of our UDP peer-to-peer project, the broadcasting mechanism plays a pivotal role in informing all connected peers about the newly logged-in user. When a user logs in by entering the username and password, the system initiates the broadcasting mechanism. A broadcast message containing information about the newly logged-in user, including their nickname, is prepared. The system then sends the broadcast message to all connected peers using the UDP (User Datagram Protocol) broadcasting mechanism. Upon receiving the broadcast message, each peer updates the list of active users, making it available for all connected peers to view.

The broadcasting mechanism in our peer-to-peer system enables any user to send a message to all other connected users simultaneously, which is vital for real-time communication and maintaining an engaging user experience. Once a user sends a message, it is broadcasted to all other users, ensuring everyone stays updated and connected. This broadcasting mechanism serves as the backbone of the communication system, facilitating seamless interaction between peers. The process begins when a user initiates a message. The system then takes charge of broadcasting the message to all connected peers using the UDP (User Datagram Protocol) broadcasting mechanism, ensuring each connected user receives the message almost instantly. All connected peers are continuously listening for broadcast messages. When a broadcast message is received, the system processes it and displays the message to the users. Along with the message content, the system displays who sent the message and the timestamp to all connected users. This provides users with the ability to track the origin and timing of each message, enhancing the overall user experience. By implementing this broadcasting mechanism, our peer-to-peer system fosters an environment of immediate and seamless communication. It significantly reduces the delay in message delivery, ensuring that users experience minimal latency. This feature is particularly useful for real-time collaboration and interactive discussions. The broadcasting mechanism, combined with low-latency implementation, ensures that users can communicate effectively and enjoy a smooth and uninterrupted experience within our peer-to-peer system. The broadcasting mechanism in our peer-to-peer system enables any user to send a message to all other connected users simultaneously, which is vital for real-time communication and maintaining an engaging user experience. Once a user sends a message, it is broadcasted to all other users, ensuring everyone stays updated and connected. This broadcasting mechanism serves as the backbone of the communication system, facilitating seamless interaction between peers. The process begins when a user initiates a message. The system then takes charge of broadcasting the message to all connected peers using the UDP (User Datagram Protocol) broadcasting mechanism, ensuring each connected user receives the message almost instantly. All connected peers are continuously listening for broadcast messages. When a broadcast message is received, the system processes it and displays the message to the users. Along with the message content, the system displays who sent the message and the timestamp to all connected users. This provides users with the ability to track the origin and timing of each message, enhancing the overall user experience. By implementing this broadcasting mechanism, our peer-to-peer system fosters an environment of immediate and seamless communication. It significantly reduces the delay in message delivery, ensuring that users experience minimal latency. This feature is particularly useful for real-time collaboration and interactive discussions. The broadcasting mechanism, combined with low-latency implementation, ensures that users can communicate effectively and enjoy a smooth and uninterrupted experience within our peer-to-peer system.

IV. Future Scope

Looking ahead, there are several avenues to explore for the future development of our UDP peer-to-peer project. One key enhancement could be the implementation of end-to-end encryption, which would significantly bolster security and privacy within the system. This feature would ensure that messages are encrypted before being broadcast, allowing only the intended recipients to decipher them, thus adding an extra layer of protection. Additionally, integrating file-sharing capabilities would further expand the functionality of the system. Enabling users to securely share files within the peer-to-peer network would make the system more versatile and appealing to users who need to exchange files. Moreover, enhancing user authentication and authorization mechanisms would provide a more secure environment. By implementing multi-factor authentication and access control, the system would ensure that only authorized users can access and interact with it, thereby strengthening the security posture of the application. These developments would make our peer-to-peer system more robust, secure, and feature-rich, catering to the evolving needs of users in a dynamic digital landscape.

V. Conclusion

The broadcasting mechanism in our peer-to-peer system allows any user to send a message to all other connected users simultaneously. This is particularly important for real-time communication and maintaining an engaging user experience. When a user sends a message, it is broadcast to all other users, ensuring everyone stays updated and connected. In this context, the broadcasting mechanism serves as the backbone of the communication system, enabling seamless interaction between peers. The process begins when a user initiates a message. The system then takes charge of broadcasting the message to all connected peers using the UDP (User Datagram Protocol) broadcasting mechanism. This ensures that each connected user receives the message almost instantly. All connected peers are continuously listening for broadcast messages. When a broadcast message is received, the system processes it and displays the message to the users. Along with the message content, the system displays who sent the message and the timestamp to all connected users. This provides users with the ability to track the origin and timing of each message, enhancing the overall user experience. By implementing this broadcasting mechanism, our peer-to-peer system fosters an environment of immediate and seamless communication. It significantly reduces the delay in message delivery, ensuring that users experience minimal latency. This feature is particularly useful for real-time collaboration and interactive discussions. The broadcasting mechanism, combined with low-latency implementation, ensures that users can communicate effectively and enjoy a smooth and uninterrupted experience within our peer-to-peer system.

References

[1] Peer-to-Peer: Harnessing the Benefits of a Disruptive Technology. OReilly, 2000.

[2] S. Deering, D. Estrin, D. Farinacci, V. Jacobson, C. Liu and L. Wei, "The PIM Architecture for Wide-Area Multicast Routing IEEE/ACM Transaction on Networking", vol. 4, no. 2, April 1996.

[3] S. Floyd, V. Jacobson, C-G Liu, S. McCanne and L. Zhang, "Reliable Multicast Framework for light-weight sessions and application level framing", Processings from SigCOMM95, September 1995.

[4] A. J. Ballardie, P. F. Francis and J. Crowcroft, "Core based trees", Proc. ACM SIGCOMM, 1993.

[5] M. W. Bern and R. L. Graham, "The shortest-network problem", Scientific American, pp. 84-89, Jan. 1989.

[6] F. Thompson (2023, December 01). "The Rise of Serverless Functions in Web Development." Smashing Magazine.

[7] B. Cain, A. Thyagarajan and S. Deering, Internet Group Management Protocol, July 1995.

[8] S. Casner, "Second ietf internet audiocast", Internet Society News, vol. 1, no. 3, pp. 23, 1992.

[9] D. D. Clark, "The design philosophy of the darpa internet protocols", Proc. ACM SIGCOMM, 1988.

[10] Y. K. Dalal and R. M. Metcalfe, "Reverse path forwarding of broadcast packets", Commun. ACM, vol. 21, no. 12, pp. 1040-1048, 1978.

[11] S. Deering and D. Cheriton, "Multicast routing in a datagram internetworks and extended lans", ACM Trans. Comput. Syst., pp. 85-111, May 1990.

[12] S. Deering, D. Estrin, D. Farinacci and V. Jacobson, Protocol Independent Multicast (PIM) Dense Mode Protocol: Specification, Mar. 1994.

[13] S. Deering, D. Estrin, D. Farinacci, V. Jacobson, C. Liu and L. Wei, Protocol Independent Multicast (PIM): Motivation and Architecture, Nov. 1994.

[14] S. Deering, D. Estrin, D. Farinacci, V. Jacobson, C. Liu and L. Wei, Protocol Independent Multicast (PIM) Sparse Mode Protocol: Specification, Sept. 1995.

[15] S. Floyd, V. Jacobson, C. Liu, S. McCanne and L. Zhang, "A reliable multicast framework for light-weight sessions and application level framing", Proc. ACM SIGCOMM, 1995.

[16] R. Frederick, "Left audio & videocast", Internet Society News, vol. 1, no. 4, pp. 19, 1993.