IJARSCT
International Journal of Advanced Research in Science, Communication and Technology
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
ISSN: 2581-9429
Volume 5, Issue 1, December 2025
Impact Factor: 7.67

# Collaborative White Board Synchronization in Networks

**Saketh R, Sudeep Reddy N, Suraj, Dr. Nirmala H**
Department Information Science and Engineering
Global Academy of Technology, Bengaluru, India
sakethr1ga22is134@gmail.com, sudeepreddy1ga22is164@gmail.com
Suraj1ga22is167@gmail.com, dr.nirmala@gat.ac.in

**Abstract:** *The rapid growth of remote learning, distributed teams, and online collaboration has created a strong demand for interactive tools that allow multiple users to work together in real time. Traditional collaborative whiteboard platforms are primarily web-based and often on complex backend infrastructure, which can introduce latency, require constant connectivity and provide limited control over the underlying communication processes. These limitations highlight the need for a lightweight, secure, and efficient system that supports seamless cooperation over computer networks. The proposed project, Collaborative Whiteboard Synchronization Using Computer Networks, addresses this need by designing and implementing a real-time, secure, synchronized whiteboard application using socket programming enhanced with SSL/TLS encryption. The system is based on a client– server architecture where the server acts as a central communication hub, responsible for managing multiple client connections and broadcasting drawing updates to all participants. Communication between clients and server takes place over TCP sockets wrapped with SSL/TLS certificates to ensure confidentiality, integrity and authenticated data exchange. This secure channel prevents unauthorized access, guarantees data protection and makes the system suitable for academic, professional and organizational environments where sensitive information may be shared. Each client application features a Tkinterbased graphical user interface that provides an intuitive whiteboard where users can draw, annotate, and interact using different colours and brush thicknesses. When a user draws on the white board, the system captures the corresponding mouse events and converts them into drawing commands comprising coordinates, colour information and stroke style. These commands are transmitted to the server, which instantly distributes them to all other clients connected to the session. This broadcast mechanism ensures that every participant views a consistent and synchronized canvas, regardless of the number of users or the order in which drawing actions occur. The design also supports collaborative editing features such as UNDO, REDO, and CLEAR, allowing users to modifier reset the canvas collectively. These commands are shared with the same level of synchronizations normal drawing actions, ensuring uniformity across all whiteboard instances..*

**Keywords**: *SSL/TLS encryption*

## I. INTRODUCTION

The increasing reliance on digital communication and remote collaboration has transformed the way people learn, work, and share information. In educational institutions, corporate environments, and online communities, the need for tools that allow real-time interaction and brainstorming has grown significantly. Among these tools, digital whiteboards have emerged as an effective medium for visual communication, enabling users to draw, annotate, and exchange ideas instantly. However, many existing solutions depend heavily on web technologies and cloud-based infrastructures, which often introduce latency, require constant internet connectivity, and do not provide developers with full control over low-level network operations. Such systems may also lack strong, customizable security features, making them less suitable for sensitive or closed network collaboration. To overcome these limitations, a more direct, transparent,

IJARSCT
International Journal of Advanced Research in Science, Communication and Technology
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
ISSN: 2581-9429 Volume 5, Issue 1, December 2025 Impact Factor: 7.67

and secure communication approach is required—one that leverages computer networks at the socket level while still delivering seamless real-timeshared interaction. In this context, the project "Collaborative Whiteboard Synchronization" Using Computer
Networks" presents a real-time multi-user whiteboard application built using Python socket programming and SSL/TLS encryption. This system is designed to operate efficiently over LAN or controlled network environments and provides a synchronized drawing experience where multiple clients share a common digital canvas. When a user draws on the whiteboard, the system captures the drawing action, converts it into structured coordinate data, and transmits it securely to a central server. The server then broadcasts this data to all connected clients, ensuring that every participant sees the same content in real time. This mechanism creates a
smooth, low-latency collaborative environment without relying on external servers, third-party services, or high-level web protocols.

## II. PROPOSED SYSTEM

The diagram shows how a secure collaborative whiteboard works using a Client–Server model with SSL/TLS.
Client A & Client B (Tkinter GUI)
• User draws on the canvas (mouse input).
• The client sends drawing data (x1, y1, x2, y2, colour, thickness) to the server securely using SSL/TLS.
• The client also receives drawing updates from the server and displays them on its canvas.
• Special commands like UNDO, REDO, CLEAR are also sent to the server. Server (SSL Socket) • Uses SSL certificate to secure communication.
• Listens for client connections.
• Receives drawing actions from any client.
• Broadcasts those actions to all connected clients. Overall Working
• Client A draws → sends to Server→ Servers ends to Client B.
• Client B draws → sends to Server→ Servers ends to Client A.
• This keeps all users' whiteboards Synchronized in real time.

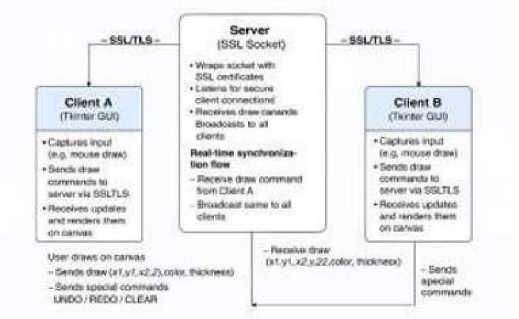



Fig. 2.System Architecture and Workflow

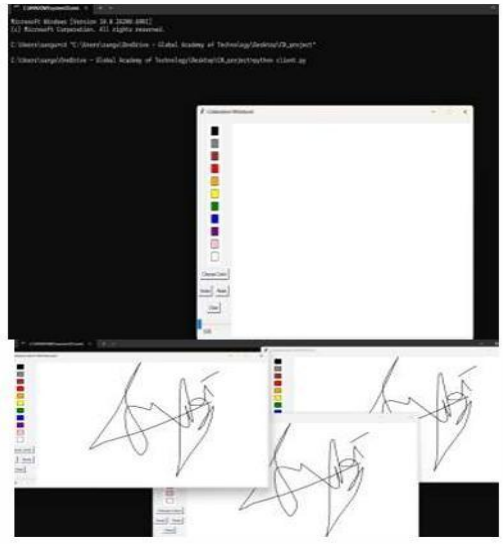
## III. METHODOLOGY

The implementation of the Collaborative Whiteboard Synchronization System is carried out using Python and follows a secure, multithreaded client–server architecture built on top of TCP socket programming with SSL/TLS encryption. The server acts as the backbone of the application, responsible for managing secure connections, receiving drawing data from users, and broadcasting updates to all connected clients. To achieve this, the server initializes a TCP socket, loads SSL certificates, and wraps the socket using an
SSL context to ensure that all communication is encrypted and authenticated. When the server starts, it listens on a dedicated port for incoming client requests. As new clients join, the server spawns a separate thread for each connection, allowing multiple users to interact simultaneously without affecting the responsiveness of the system. On

IJARSCT
International Journal of Advanced Research in Science, Communication and Technology
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
ISSN: 2581-9429
Volume 5, Issue 1, December 2025
Impact Factor: 7.67

the client side, the application is implemented with a Tkinter graphical interface that provides an intuitive whiteboard environment. When the user draws on the mouse events and immediately converts them into drawing coordinates that represent the start and end points of each line synchronization with the rest of the users. SSL/TLS ensures that all segment. These coordinates, along with metadata such as brush colour and thickness, are serialized into a structured message format and transmitted to the server over a secure SSL/TLS connection. The use of encryption is crucial, as it ensures that the transmitted drawing data cannot be intercepted or modified by unauthorized parties, which is essential for secure collaborative environments such as academic discussions or professional brainstorming sessions. Real-time synchronization is achieved through continuous communication between clients and the server. Each client runs a dedicated background thread that listens for incoming messages from the server. Whenever a drawing action or special command—such as UNDO, REDO, or CLEAR—is received, the client updates its canvas accordingly, ensuring all participants always share an identical view of the whiteboard. The server acts purely as a message relay, broadcasting updates to every connected user, which prevents inconsistencies and ensures that the state of the canvas stays synchronized across all clients. Special commands are implemented as simple text-based instructions sent over the same SSL channel. For example, when a user presses the CLEAR button, the client sends a "CLEAR" message to the server, which then broadcasts this command to all other users. Each client interprets the command and modifies the canvas appropriately. Undo and redo functionalities are supported by maintaining a stack-like structure that records the sequence of drawing actions, allowing users to revert or restore changes in a collaborative manner. To ensure scalability and responsiveness, both the server and the client leverage Python's multithreading capabilities. The server creates a new thread for every client, preventing one slow or inactive connection from interrupting the entire system. The client uses two independent threads—one for drawing actions and one for receiving updates—so that communication and user interaction occur seamlessly in parallel. This architectural design minimizes latency and allows the system to support many users in a live session. Finally, SSL/TLS integration ensures secure data transmission throughout the communication pipeline. SSL certificates and keys are generated using tools like OpenSSL, and the client verifies the server's certificate before initiating communication. This mechanism prevents man-in-the-middle attacks and ensures that all connected users are communicating with an authenticated server. Through the combined use of socket programming, GUI event handling, secure encryption, and multithreaded communication, the project achieves a robust and efficient real time collaborative whiteboard system capable of supporting interactive teamwork across computer networks.

## IV. RESULTS AND DISCUSSIONS



IJARSCT
International Journal of Advanced Research in Science, Communication and Technology
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
ISSN: 2581-9429 Volume 5, Issue 1, December 2025 Impact Factor: 7.67

The image shows multiple client windows of the Collaborative Whiteboard application running at the same time. A drawing made on one client is instantly reflected on all other connected clients. This demonstrates that the whiteboard synchronization is working correctly— every user sees the same sketch in real time. The server receives drawing data from one client and broadcasts it to all others, keeping every canvas updated simultaneously.

## V. FUTURE SCOPE

1. User Authentication &Access Control Adding login systems to ensure secure participation and personalized user sessions.
2. Advanced Drawing Capabilities Incorporating features such as text tools, shapes, layers, erasers, and diagramming tools for richer collaboration.
3. Web-Based Implementation Developing a browser-accessible version using Web Sockets to remove the need for local installation.
4. Cloud Storage &Session Management Enabling users to save, load, and share previous whiteboard sessions online.
5. Voice, Video & Chat Integration Adding communication channels for complete virtual collaboration.
6. Mobile and Tablet Support Extending the system to Android/iOS devices to enhance accessibility.
7. AI Features Integrating handwriting recognition, shape correction, and intelligent auto-suggestions.
8. High Scalability& Distributed Architecture Using load balancing, distributed servers, or CRDT/OT algorithms to support large groups of users without conflicts.

## VI. CONCLUSION

The Collaborative Whiteboard Synchronization Using Computer Networks project successfully demonstrates how secure, real-time collaboration can be achieved through socket programming and SSL/TLS communication. By integrating a Tkinterbased graphical interface with a multithreaded client–server architecture, the system provides a smooth and synchronized drawing experience across multiple users. Each drawing action is captured, transmitted securely, and instantly broadcasted to all connected clients, ensuring that every participant views the same canvas without delay. The use of SSL/TLS strengthens data confidentiality and integrity, making the system suitable for educational, professional, and remote teamwork environments. Overall, the project highlights the practical application of networking concepts, secure communication protocols, and event-driven GUI design to create an efficient and interactive collaborative tool.

## REFERENCES

**[1].** Tanenbaum, A. S., & Wetherall, D. J. Computer Networks (5th ed.). Pearson Education, 2010.
**[2].** Forouzan,B. A. Data Communications and Networking (4th ed.). McGraw-Hill, 2007.
**[3].** Stevens, W. R. UNIX Network Programming: The Sockets Networking API (Vol. 1). Addison Wesley, 2003.
**[4].** Beej,B.Beej'sGuideto Network Programming Using Internet Sockets. beej.us/guide, 2019.
**[5].** OpenSSL Project OpenSSL documentation. https://www.openssl.org/docs/
**[6].** 6.Rescorla, E. SSL and TLS: Designing and Building Secure Systems. Addison-Wesley, 2000.
**[7].** Fielding,R., et al. RFC 5246: The Transport Layer Security (TLS) Protocol Version 1.2. IETF, 2008.
**[8].** Shneiderman.B, Designing the User Interface: Strategies for Effective Human– Computer Interaction. Pearson, 2016.
**[9].** Ousterhout J. K. Tcl and the Tk Tool kit. Addison-Wesley, 2010.
**[10].** Li, X., & Li, Z. "Real-Time Collaboration Techniques Based on Sockets and Web Sockets." International Journal of Computer Applications,2018.
**[11].** Ahmed, M., et al. "A Survey on Real-Time Collaborative Editing Systems." Journal of Systems and Software, 2016.
**[12].** Tan K, & Wu, H. "Efficient Multi-User Synchronization in Distributed Whiteboard Systems." IEEE Conference on Distributed Computing Systems, 2019