DMPedia Lecture Notes in Computer Science & Engineering
Volume 1, Issue 2, pp.451-460, 2026
ICICCS-2026

A Comprehensive WebRTC-Based Platform for Remote Assessment and Synchronous Technical Collaboration

Varun Kumar R, Bala G, Ms.S.Babitha

Department of Information Technology, Hindustan Institute of Technology and Science

Kelambakkam, Tamil Nadu 603103, India

Email: enquiretovarun@gmail.com, bxg8491@gmail.com, babi2289@gmail.com

Abstract

With the rapidly growing technology, communication and collaboration have become much simpler and more efficient. Web Real-Time Communication (WEBRTC) enables secure, peer-to-peer (P2P) communication between two IP addresses. Common vulnerabilities include exposure of private IP addresses, which may lead to Man-in-the-Middle (MitM) attacks, and a lack of end-to-end encryption (E2EE), which makes data visible or vulnerable to interception, resulting in reduced security and privacy. Most video conferencing apps use P2P mesh, which helps transmit video directly between IP addresses. It also has the drawback of instability when the stream reaches more than 4 users. Selective Forwarding Unit (SFU) architectures are widely used for forwarding individual streams of data; however, they lack decoding on the client side, which leads to high CPU consumption. By addressing these identified gaps, our Project aims to contribute to the development of NextGen video conferencing applications that not only allow authenticated participants to attend video conferences assigned by the host or admin, but also enhance security and privacy.

Keywords: Web Real Time Communication (WebRTC), peer-to-peer (P2P), Man-in-the-Middle (MitM) attacks, Selective Forwarding Unit (SFU).

I. Introduction

In the current video conferencing app, security and role-based authentication and authorisation are configured manually. Still, our goal is to ensure authentication and authorisation for automatic analysis. By analysing the current user ID with the already predefined user list provided by the admin or host, the participant can join the meeting. The current key features provided by us are:

- Email ID-based authentication to gain or revoke access to join the meeting.
- Accounting for all the meetings held in a month by addressing when the meeting occurred, how long the meeting lasted, and how many active participants attended the meeting.
- Past held important sections are accessible to their participant if the host allows them access.
- In-built code compiler and online assessment conductor for examination listed as features.
- Like the traditional video conferencing app tools like white board, chat box and screen sharing options also available.

The proposed system for our product is to ensure security and enhance the user experience by providing a simplified UI and quality-of-life tools.

II. Literature Review

The rapid growth of remote communication technologies has increased reliance on real-time video conferencing systems for education, business collaboration, and social interaction. WebRTC has become a key enabling technology for browser-based communication due to its peer-to-peer architecture, low latency, and built-in encryption support. However, ensuring secure, scalable, and high-quality conferencing remains a major research focus.

Proceedings of ICICCS-2026 (Poster Papers), DMPedia Lecture Notes in Computer Science & Engineering.
ISBN: 978-81-993813-1-5

Security considerations form the foundation of modern conferencing systems. Tian and Jiang [1] proposed a WebRTC-based encrypted media communication framework to protect real-time data transmission against unauthorised access by implementing the SFU (selective forwarding unit) method, which encrypts and decrypts media such as audio and video files, preventing data loss.

Li [2] analysed potential risks in conferencing platforms and provided a methodology for breaking down the conferencing system into components (client, server, network, storage) and identifying vulnerabilities at each layer to classify threats and estimate impact, thereby helping avoid MitM (man-in-the-middle) attacks.

Sharma et al. [3] introduced a WebRTC-based file transfer system that demonstrates secure direct communication between peers without centralised storage; instead of connecting through a server, direct channelling helps reduce security risks and significantly boost computation time, thereby reducing latency.

Deshmukh et al. [4] implemented a student discussion platform using WebRTC to facilitate interactive communication, which also provides a secure path for sending and receiving text data via the WebRTC media transfer protocol, so that messages are encrypted while sending and decrypted upon receipt to avoid data loss.

Solanki and Senapati [5] proposed a method to reduce latency in multilingual broadcasting by enhancing WebRTC streaming distribution mechanisms, improving routing and media handling, and supporting multilingual data flow for audio/video streams alongside language-processing components. This system maintains stability and latency during deployment.

s. Ishola and Mailer [6] introduce a novel sampling method, customised stratified-cluster sampling, to optimise the active measurement configuration for simulating WebRTC network behaviour, which helps to maintain the Session traversal and relay traversal more efficiently on network routing and helps to reduce the load of network trafficking.

Diaz et al. [7] proposed a method to address the limited codec and video format problem in browsers using WebRTC data channels and support drivers, by encapsulating media into video fragments and pushing the data source into the data channel via additional optimised extensions available in browsers. This method enhances the colour accuracy and gradient of the video on browsers.

Das and Ghosh [8] developed a virtual classroom integrating WebRTC with modern web frameworks to enable secure remote learning and help teachers share assignments, announcements, and course materials. All source materials were stored in Firebase for easy, faster access.

Lu et al [9] proposed a method where WebRTC is used for real-time audio, video and data sharing within most browsers by integrating WebSocket, which allows full-duplex or two-way communication between users, where changes appear immediately on all participants' webpages, so it became an effective solution for modern collaborative work and communication.

Singla et al. [10] demonstrated the scalability of real-time communication by implementing a chat-based application within a video conferencing app to enhance engagement and collaboration. Also, the message was encrypted, so only the participants in the meeting room can read the chat admin can also disable or monitor the chat function.

WebRTC typically uses Google Congestion Control (GCC) to estimate available network bandwidth and adjust the video bitrate accordingly. In large-scale streaming setups that use a Selective Forwarding Unit (SFU) to distribute video to multiple users, GCC can sometimes misjudge network capacity — especially when simulcast traffic is involved. As a result, the system may increase the video bitrate too much, causing network congestion and leading to freezes, skipped video, and poor viewing quality. Sato et al. [11] proposed adaptive video quality switching based on bandwidth probing to improve quality of experience—synchronisation techniques for collaborative media delivery.

Bhalange et al. [12] performed load testing and end-to-end integration tests to demonstrate an average response time of 2 milliseconds for up to 800 requests per second, with a maximum end-to-end connection latency of 8 seconds, and suggested that p2p for media transfer is efficient without centralised infrastructure.

Yu et al [13] Their system uses MQTT to handle communication between devices during call setup and management, reducing signalling data usage while maintaining full video calling functionality. Testing shows that the solution supports reliable video calls, consumes minimal signalling bandwidth, and achieves acceptable call connection delays. Overall, the work demonstrates that MQTT can be an efficient alternative signalling method for mobile video call systems.

Ko et al[[14] instead of relying on traditional rule-based methods, the system uses a trained DRL (Deep Reinforcement Learning) model to calculate exactly how much video bitrate is needed for each participant based on their current network stability. A data synchronisation method between the media server and the DRL system enables the trained model to continuously learn and adjust its decisions to maximise overall meeting quality for all users. Experimental results show that the framework adapts well to changing network conditions and improves video bitrate delivery compared to conventional methods — increasing bitrates by about 5% in stable networks and up to 35% in highly dynamic environments. This demonstrates the potential of AI-driven allocation techniques to enhance user experience in multi-party video conferencing systems.

Ito et al [15] Unlike previous research focused on TCP-based systems, this approach sends streaming data across multiple network paths simultaneously to improve reliability. The framework was tested using a WebRTC-based streaming application while moving through an urban environment across different mobile networks. Results show that it significantly reduces packet loss and increases streaming bitrate without disrupting existing congestion control mechanisms. Overall, the study demonstrates that multipath redundancy can enhance stability and quality in real-time streaming scenarios, making communication more reliable even under varying network conditions.

Apriyani et al. [16] implemented a machine-learning model called the Dual-Signal Transformation Long Short-Term Memory Network directly in the user's browser. The system runs entirely on the client side using Open Neural Network Exchange. Runtime-Web and Web Workers, meaning audio processing happens locally rather than on a server. This helps keep latency low and protects user privacy. This helps to reduce noise in environments during media exchange. It can potentially reduce unwanted noise and serve as a noise-cancellation feature in video conferencing applications.

Shaikh et al. [17] introduced a Drop Bridge a lightweight peer-to-peer file transfer system built on WebRTC. This allows files to be transferred securely and efficiently without being stored on intermediate servers. This design keeps the system simple by avoiding external databases or complex middleware, reducing transfer latency, protecting data through end-to-end encryption, and working even in local or low-connectivity environments. This will be a core feature for decentralised peer discovery and blockchain-based authentication .

III. Proposed System

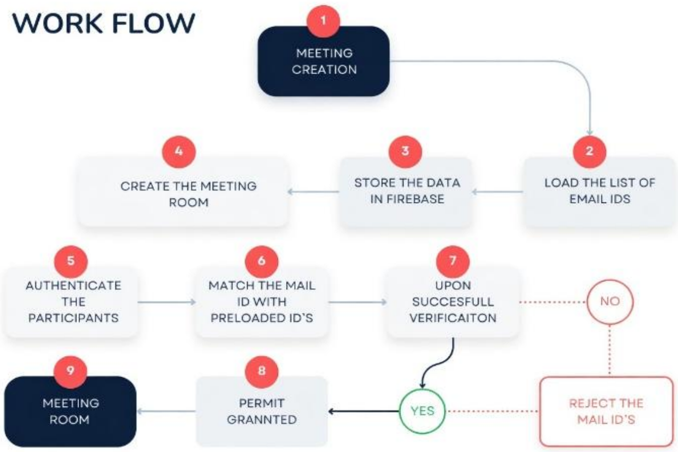

Fig. 1. Working principle

Our proposed system is fully operated and controlled by the host/admin who creates the meeting. First, the admin created a scheduled meeting with the timing and loaded the list of participants into the meeting portal. Upon creating the portal, all participants' data from the meeting was stored in the database for authentication. The meeting lobby is now ready to schedule a time to come. Once the meeting time arrives, if the participants log in with their registered email IDs that match the host's list of participants' email IDs, they will be allowed into the meeting. If the authentication fails, the permission will not be provided as per Fig. 1.

IV. Implementation

A. User profile

In our project panel, we have a user authentication option. With Firebase authentication, we can verify users' identities. This helps us during authentication and entry into the meeting lobby. Additionally, user personalisation, such as profile pictures and usernames, is stored. Dark mode or light mode is stored as browser cookies using JSON Web Tokens and also in Firebase when the account is created. To avoid delays in profile details and improve accessibility, we are using Cloudinary to reduce latency. The user profile also visualises the total number of meeting hours attended over a period as statistics. Furthermore, upcoming meetings are also listed in the user profile as per Fig. 2.

Fig. 2. User profile

B. **Authentication**

User authentication is based on their email ID. If the participant's email ID matches the ID listed by the host, then they will be permitted into the meeting. If an unauthorised participant attempts to attend the meeting, they will be denied access immediately. If the host does not mention the participant list, the meeting will be automatically considered a public meeting, allowing any participants to attend. Data for every scheduled meeting, such as the number of participants, was encrypted to prevent a man-in-the-middle attack Fig.3 .

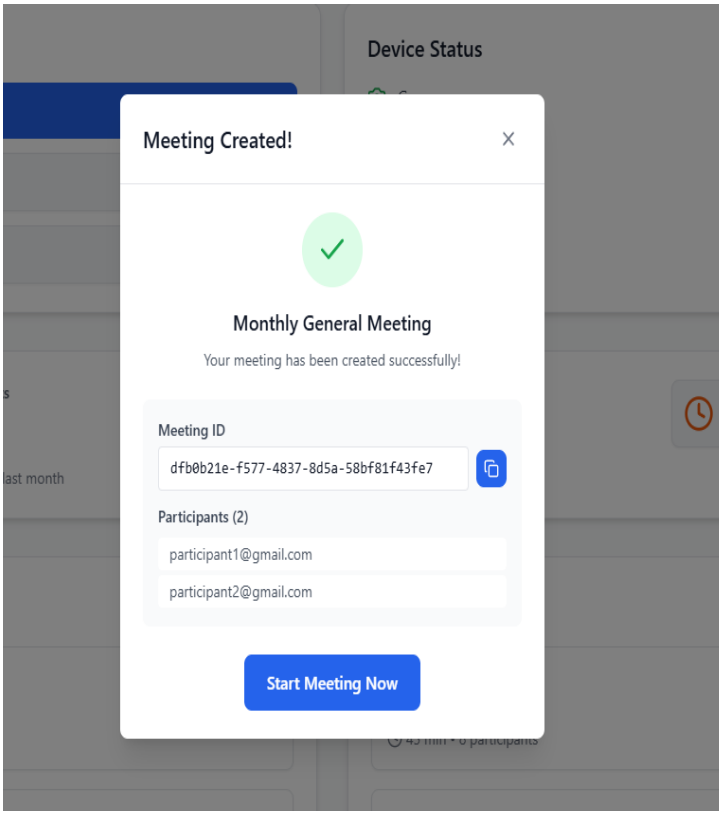

Fig. 3. Lobby creation

C. **Connectivity**

Establishing a session /connectivity between devices is done using WebRTC. With the help of WebRTC, we can transfer real-time data, such as audio, video, and text, within our app. Our application works across major browsers with minimal system requirements Fig. 4.

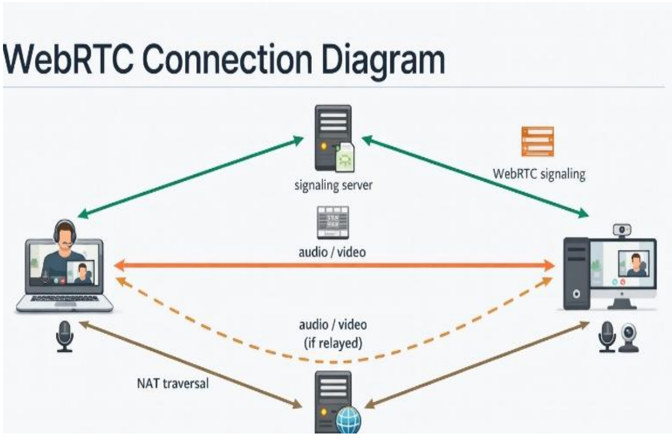

Fig. 4 Working principle of WebRTC

WebRTC API keys enable access to, recording of, and transmission of video/audio and text data between multiple devices via secure peer-to-peer connectivity in browsers.If users were on different IP addresses, NAT (Network Address Translation) firewalls would prevent real-time connections, so WebRTC uses a STUN (Session Traversal Utilities for Network Address Translation) server. This helps translate the given IP address to a public IP for p2p establishment.

In some scenarios, where networks are so restrictive that the STUN (Session Traversal Utilities for NAT) server also can't help translate the IP address. In those cases, WebRTC uses traversal using relay (Traversal Using Relays around NAT) to find the best connection through the traffic.

D. **Meeting Room**

The meeting room is where all meetings are held, and all participants are listed in a grid. Participants are listed in Alphabetical order, and when a participant is on the mic speaking, the participant's ID is shown at the top of the list. The meeting room also includes tools such as live captioning, a whiteboard, a chat system, polls, live recording, and screen sharing, all with a simplified UI Fig. 5.

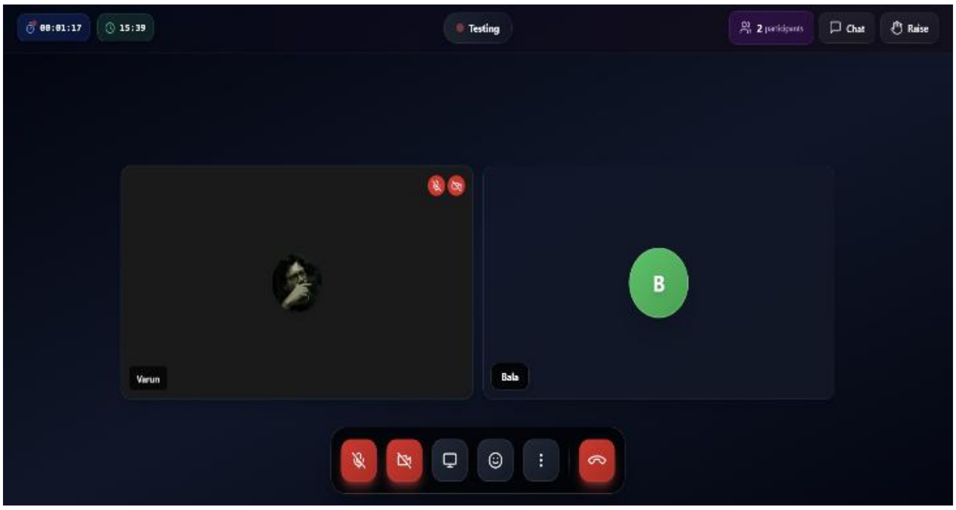

Fig 5 Meeting Room

E. **Storage**

To further improve storage security, we implemented functions such as a user account deletion process that retains user data for up upto 15 business days. users' scheduled data, number of meetings attended, and other data are also stored in the database. Chats during meetings were temporarily stored in a database to reduce the latency. Admins can delete all meeting data from the database if needed.

F. **Settings**

Users can set their appearance in dark or light mode. Notification settings, such as push notification (Receive meeting and chat notifications) screen share notification (Get notified when someone shares their screen), chat sound (Play sound when receiving chat messages) can be sent on/off here and also features like Auto Join Audio(Automatically connect microphone when joining, Auto Join Video (Automatically turn on camera when joining) also be customized in settings. Delete Account erases all users' data from the database.

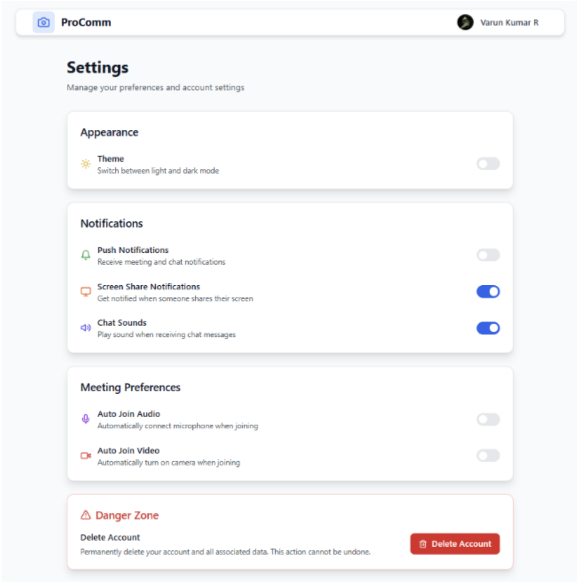

Fig 6 Settings

V. Result

With the successful implementation of our proposed system, which is hosted on the internet and deployed in a versatile manner. Our project was accessible online via a web app, which allowed meeting sessions to be created and scheduled. Upon successful Looby creation, the host can send the meeting link to the participants. Now, registered participants can attend the meeting only after ID verification via Firebase authentication. Using the Joodle API key, we implemented a code editor and an online assessment tool. Our project also allows streaming 1080p video at 60 fps in any browser.

Criteria	ProComm	Microsoft Teams	Zoom meet
Max. quality	1080p	1080p	1080p
Max. Frame Rate	60 frames. /sec.	60 frames. /sec.	60 frames. /sec.
User registration is required to attend the meeting	Yes	No	No
The need to install software	No	No	No
In the built code editor	Yes	No	No
In build assessment mode	Yes	No	No

Features

A. **Online assessment mode**

In our project, we implemented an online assessment mode during host surveillance. Many institutes and schools benefited from this feature, making the online test easier. The host creates the set of questions preloaded into the assessment mode. Participants can't choose answers or write them in the text box; they are later evaluated manually by the host. For text-mode answering, choose-type questions were automatically evaluated as per Fig. 7.

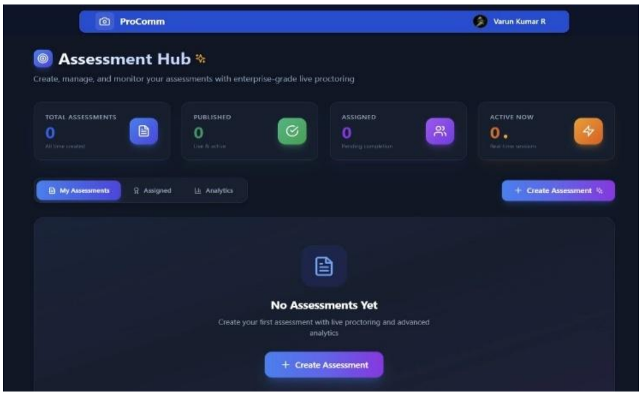

Fig.7. Assessment Mode

B. **Tracker**

It shows how many meetings were held this month, how many hours of meetings were attended, and tracks the records for the month, making it easy to filter by dates and important information from the meetings.

C. **Complier**

In our proposed web app, we implement a compiler to help online code assessment interview conductors benefit from this feature. Similar to the assessment mode, this mode is also under the host's surveillance. Our compiler supports programming languages such as Python, Java, JavaScript, C, and C++. Our compiler can run most data structures and algorithm code types without additional extensions, thanks to the JODI API key Fig. 8.

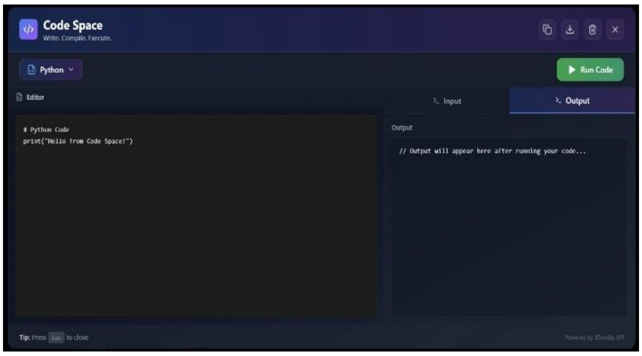

Fig.8. Code Compiler Mode

D. **Other Tools**

Other tools, such as a whiteboard, chatbot, screen share calculator, and Calendar, were also implemented for users' convenience.

VII. Conclusion

WebRTC makes real-time video calls and chats possible without additional tools or plugins. It's good for businesses that want to have real-time meetings or communication, or conduct assessments, by integrating WebRTC with Firebase authentication and API keys. We successfully created a video conferencing system with a secure way of allowing selective participants to attend the meeting or conduct assessments. With more budget and time, we can even optimise and improve the quality and longevity of our project.

References

[1] Y. Tian and Z. Jiang, "Design of Real-Time Media Encrypted Conference System Based on WebRTC," 2024 13th International Conference of Information and Communication Technology (ICTech), Xiamen, China, 2024, pp. 326-331, doi: 10.1109/ICTech63197.2024.00067.

[2] Y. Li, "Security risk analysis of video conference system," 2023 International Conference on Networking, Informatics and Computing (ICNETIC), Palermo, Italy, 2023, pp. 134-138, doi: 10.1109/ICNETIC59568.2023.00034.

[3] V. Sharma, M. K. Pandey, J. Gupta and H. Kishan, "Peer to Peer Communication using Web Sockets and Web RTC," *2024 International Conference on Emerging Innovations and Advanced Computing (INNOCOMP)*, Sonipat, India, 2024, pp. 627-634, doi: 10.1109/INNOCOMP63224.2024.00108.

[4] R. Deshmukh, A. Pawar, D. Wagh, A. Kudale and N. Nand, "Group Discussion Platform for Students Using WebRTC: TalkativeTribe," 2024 2nd International Conference on Sustainable Computing and Smart Systems (ICSCSS), Coimbatore, India, 2024, pp. 881-888, doi: 10.1109/ICSCSS60660.2024.10625193.

[5] J. Solanki and B. Senapati, "Enhancing Real-Time Multilingual Communication in Virtual Meetings through Optimized WebRTC Broadcasting," 2024 IEEE International Conference and Expo on Real Time Communications at IIT (RTC), Chicago, IL, USA, 2024, pp. 1-8, doi: 10.1109/RTC62204.2024.10739086.

[6] H. XU, Y. Du, X. Li, F. Gao, L. Shi and Z. Yin, "Optimized Active Measurement Configurations for Monitoring WebRTcNetwork Behavior," *2024 6th International Conference on Frontier Technologies of Information and Computer (ICFTIC)*, Qingdao, China, 2024, pp. 260-264, doi: 10.1109/ICFTIC64248.2024.10912888.

[7] D. Diaz, D. Stolarz and J. Aguerre, "Toward Real-Time Video Streaming Over WebRTC Data Channels to Support Supplementary Video Codecs and Formats in the Browser," 2024 IEEE International Conference and Expo on Real Time Communications at IIT (RTC), Chicago, IL, USA, 2024, pp. 39-45, doi: 10.1109/RTC62204.2024.10739162.

[8] A. K. Das and S. Ghosh, "Real-Time Virtual Classroom Platform: Integrating React, Firebase, WebRTC, and Socket.io," 2025 8th International Conference on Electronics, Materials Engineering & Nano-Technology (IEMENTech), Kolkata, India, 2025, pp. 1-4, doi: 10.1109/IEMENTech65115.2025.10959621...

[9] P. Sharma, T.Nagpal, G. Shrivastava, & J. D Kumar, “Machine Learning for Fake News Detection A Survey.” In *2023 5th International Conference on Advances in Computing, Communication Control and Networking (ICAC3N)*, 2023 pp. 850-855.

[10] S. Singla, A. Kumar, A. Aryan and H. R. Pathak, "Stranger Chatting App Using Nodejs and WebRTC," 2024 International Conference on Emerging Innovations and Advanced Computing (INNOCOMP), Sonipat, India, 2024, pp. 642-644, doi: 10.1109/INNOCOMP63224.2024.00110.

[11] R. Sato, Y. Kendo, S. Ohzahata and R. Yamamoto, "A Video Quality Switching Method of WebRTC SFU using Bandwidth Probing for QoE Improvement," 2025 International Conference on Meta-Networking (MEET), Tokyo, Japan, 2025, pp. 1-6, doi: 10.1109/MEET67398.2025.11336102.

[12] P. Bhalange, S. Thakur and M. Chen, "Peer-to-Peer Synchronization for Collaborative Media Streaming," 2025 IEEE 8th International Conference on Multimedia Information Processing and Retrieval (MIPR), San Jose, CA, USA, 2025, pp. 213-218, doi: 10.1109/MIPR67560.2025.00041.

[13] H.Sharma, P.Kumar, & K. Sharma, “Security Solutions for the Internet of Things Using Machine Learning and Deep Learning: Current Trends and Future Directions,” *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery*, 2026,*16*(1), e70059.

[14] K. Ko, S. Ryu, N. Van Tu and J. W. -K. Hong, "Optimizing Video Conferencing QoS: A DRL-based Bitrate Allocation Framework," NOMS 2024-2024 IEEE Network Operations and Management Symposium, Seoul, Korea, Republic of, 2024, pp. 1-10, doi: 10.1109/NOMS59830.2024.10575889.

[15] K. Ahuja, A. Nayyar, & K. Sharma,"Comprehensive Guide to Heterogeneous Networks" Academic Press,2022.

[16] M. E. Apriyani, M. N. R. Maulana and E. N. Hamdana, "Real-Time Evaluation of Client-Side DTLN Noise Suppression for WebRTC in Diverse Noise Environments," 2025 8th International Conference of Computer and Informatics Engineering (IC2IE), Depok, Indonesia, 2025, pp. 1-6, doi: 10.1109/IC2IE67206.2025.11283371.

[17] R. N. Shaikh, A. Poojary, Suraj, S. Bhandary and Sumanth, "DropBridge-A-WebRTC-Based-P2P-File-Transfer-Tool," 2025 International Conference on Future Technologies (ICFT), Sangli, India, 2025, pp. 1-6, doi: 10.1109/ICFT66708.2025.11336315.