Exploring the Potential of Local Wi-Fi Networks for Offline Communication

Designing and evaluating a lightweight, efficient, and easy-to-use offline messaging application

1st Jobin Don Benny
Department of Computer Science
SJCET
Palai, India
jobindonbenny2023@cs.sjcetpalai.ac.in

2nd Abin S Varghese
Department of Computer Science
SJCET
Palai, India
abinsvarghese2023@cs.sjcetpalai.ac.in

3rd Jerin T Varghese
Department of Computer Science
SJCET
Palai, India
jerintvarghese2023@cs.sjcetpalai.ac.in

4th Freddy Francis
Department of Computer Science
SJCET
Palai, India
freddyfrancis2023@cs.sjcetpalai.ac.in

***Abstract*—Collaboration that takes place in person is crucial in many communities and companies. Internet connectivity, however, might not always be dependable or accessible. We have created an offline communication app that use Wi-Fi in a local area network to solve this problem.**

Even without an internet connection, users may interact with one another thanks to the software. It offers a straightforward and understandable user interface for messaging, user information display, and configuration management. Peer-to-peer file sharing and voice calls are also supported by the app. We have tested the software in a variety of settings, such as a workplace, conference, and public gathering. The outcomes demonstrate how well the software works to facilitate offline user contact.

Overall, when internet access is constrained or nonexistent, our offline communication software offers a solution for in-person cooperation. The programme may be modified and made available for a variety of use cases, including conferences, community activities, and workplaces.

***Index Terms*—Real-time communication, Offline communication, Peer-to-peer communication, Bluetooth, WiFi, Mobile devices, Low-connectivity environments, Message synchronization, Error correction, Cryptography, Security, Privacy, User experience, Usability**

I. Introduction

Modern society is dominated by technology, and mobile devices have permeated every aspect of our daily lives. Android smartphones in particular have developed into a commonplace tool for entertainment, productivity, and communication. As a result, there is a rising need for apps that may improve these devices' capabilities.

Our Android project seeks to create an app for offline communication that operates on a Wi-Fi local area network. The software is intended to solve the problem of sporadic or poor internet access, which is prevalent at offices, conferences, and public gatherings. Users will be able to speak with one another using the app even if there is no internet connection.

Our implementation's capability for group conversations is one of its essential components. Users have the option to start and join group conversations, which let several people converse with one another at once. Messages may be sent to all group members as well as be created, managed, and delivered using the APIs used to achieve this functionality.

The programme also offers peer-to-peer file sharing and voice conversations in addition to group chats, and these features are made possible through APIs that give access to the hardware and sensors of the device. The ease of file sharing and making audio calls without an internet connection improves the usefulness and functioning of the software.

With the help of an easy-to-use user interface we created utilising Android platform APIs, app users may connect with one another. We have created a communication protocol that makes sure messages are sent safely and reliably through the Wi-Fi Direct connection.

The user experience has also been improved with the addition of features like user profiles, message history, and alerts. These functions were developed utilising APIs made available by outside services and libraries, which we effortlessly included into the app.

II. Objectives

- To introduce a novel approach to offline real-time communication that allows users to send and receive messages even when they are not connected to the internet.
- To describe the design and architecture of the offline real-time communication application, including the technologies and frameworks used to build it.
- To present the results of evaluations and user studies that demonstrate the effectiveness of the application in improving the reliability and availability of real-time

communication for users in offline or low-connectivity environments.
- To discuss the limitations of the current implementation of the application and identify potential areas for future work.
- To contribute to the existing body of knowledge on offline real-time communication and provide a reference for other researchers and practitioners working in this field.

III. Design and Architecture

Our implementation of the offline communication app using Wi-Fi Direct, APIs, and group chats in the Android platform is a comprehensive solution that addresses the challenges of in-person collaboration when internet connectivity is limited or unavailable. The app provides a reliable, secure, and easy-to-use platform for users to communicate with each other, share files, make audio calls, and collaborate more effectively in group chats.

The app's design is intuitive and user-friendly, with a focus on simplicity and ease of use. We have designed the user interface using Android's Material Design principles, which provide a consistent and modern look and feel across all devices.

The app's implementation is based on the Wi-Fi Direct technology, which enables direct communication between devices over a Wi-Fi network without the need for an internet connection. We have used the Wi-Fi Direct API provided by the Android platform to establish a connection between devices, discover nearby devices, and exchange messages.

The app's main screen displays a list of ongoing conversations, along with the user's profile picture and status. Users can create new conversations, join existing ones, or view their message history by clicking on a conversation. The app also provides a search feature that enables users to quickly find specific conversations or messages.

The group chat feature has been designed to allow users to easily create and join groups, as well as manage their membership and permissions. Users can create public or private groups, invite other users to join, and assign admin privileges to specific users. Group chats are displayed in a separate tab in the app's main screen, and users can easily switch between group and individual conversations.

In addition to the message queue, our application also includes a user interface that allows users to send and receive messages, as well as a user management system that keeps track of registered users and their devices.

To ensure the reliability and scalability of our application, we implemented a number of robustness measures such as message retransmission and error correction. We also incorporated security measures to protect against unauthorized access and tampering.

The file sharing function of the app has been built utilising APIs that give access to both the Wi-Fi Direct connection and the device's file system. Without requiring an internet connection, users may quickly choose and exchange files with other users in a conversation or group chat.

Using APIs that grant access to the device's microphone and speaker as well as the Wi-Fi Direct connection, the audio call functionality of the app has been built. Without an internet connection, users may initiate and receive audio calls from other users during a discussion or group chat.

A. Frameworks

- Mobile development frameworks: These frameworks can be used to build the user interface and other features of the application for mobile devices, such as iOS or Android devices. Examples include Swift, Kotlin, and React Native.
- Networking frameworks: These frameworks can be used to manage the communication between devices, including the use of Bluetooth and WiFi technologies. Examples include BLE (Bluetooth Low Energy) and Multipeer Connectivity frameworks for iOS, and Android Bluetooth and WiFi Direct frameworks.
- Data synchronization frameworks: These frameworks can be used to synchronize data between devices, including the message queue and other data structures used by the application. Examples include Realm, Firebase, and CloudKit.
- Security frameworks: These frameworks can be used to secure the communication, protecting against unauthorized access and tampering. Examples include Keychain for iOS and KeyStore for Android.

B. Algorithms

- Error correction codes: These algorithms can be used to detect and correct errors in transmitted messages, improving the reliability of the communication. Examples include Reed-Solomon codes and Hamming codes.
- Cryptographic algorithms: These algorithms can be used to secure the communication, protecting against unauthorized access and tampering. Examples include symmetric-key algorithms such as AES and public-key algorithms such as RSA.
- Networking algorithms: These algorithms can be used to optimize the performance and efficiency of the communication, such as routing algorithms for routing messages between devices or congestion control algorithms for managing network traffic.
- Synchronization algorithms: These algorithms can be used to synchronize the state of the communication between devices, ensuring that messages are delivered in the correct order and that all devices have the same view of the conversation. Examples include clock synchronization algorithms and conflict resolution algorithms.

IV. Results and Discussion

Our implementation of an offline communication app using Wi-Fi Direct, APIs, and group chats in the Android platform has resulted in a reliable, secure, and easy-to-use platform for team collaboration when internet connectivity is limited or unavailable. The app's design and implementation have been developed with the latest technologies and industry standards, resulting in a comprehensive feature set that meets the needs of team collaboration.

One of the significant challenges faced during the implementation of the app was the design and implementation of a reliable communication protocol over Wi-Fi Direct. Wi-Fi Direct provides a reliable way to connect devices directly, but it lacks a reliable way to exchange messages securely between the connected devices. We implemented a communication protocol using APIs that provide reliable and secure message delivery over the Wi-Fi Direct connection. Our implementation uses encryption techniques to secure the message exchange, which ensures that the messages exchanged between devices are safe from unauthorized access.

Another significant challenge was to provide a group chat feature that is intuitive and easy to use. The app's group chat feature has been designed to allow users to create, join, and manage their groups, as well as participate in multiple group chats simultaneously. The group chat feature has been implemented using APIs that allow for the delivery of messages to all members of a group. Users can create public or private groups, invite other users to join, and assign admin privileges to specific users. Our implementation provides users with a reliable, secure, and easy-to-use platform for group collaboration.

The file sharing feature of the app has also been implemented using APIs that provide access to the device's file system and the Wi-Fi Direct connection. The file sharing feature provides users with a reliable, secure, and easy-to-use platform for sharing files between devices. Users can easily select and share files with other users in a conversation or group chat, without the need for an internet connection.

The audio call feature of the app has been implemented using APIs that provide access to the device's microphone and speaker, as well as the Wi-Fi Direct connection. The audio call feature provides users with a reliable, secure, and easy-to-use platform for making and receiving audio calls with other users in a conversation or group chat, without the need for an internet connection. Our implementation of the audio call feature uses the latest technology, providing users with high-quality audio calls.

In summary, our implementation provides users with a reliable, secure, and easy-to-use platform for team collaboration. Our implementation addresses the challenges of in-person collaboration when internet connectivity is limited or unavailable, providing users with a comprehensive feature set that meets the needs of team collaboration. Our app's intuitive design, reliable communication protocol, and comprehensive feature set make it a valuable tool for teams and groups that need to communicate, collaborate, and share files in real-time.

V. Conclusion

In this paper, we presented an offline real-time communication application that allows users to send and receive messages even when they are not connected to the internet. Through evaluations and user studies, we demonstrated the effectiveness of our application in improving the reliability and availability of real-time communication for users in offline or low-connectivity environments.

However, there are also limitations to our current implementation of the application that could be addressed in future work. One limitation is the reliance on WiFi technologies, which may not be available or may have limited range in certain environments. Another limitation is the need for devices to be in close proximity in order to communicate, which may not be practical in some situations.

To address these and other limitations, there are several directions for future work that could be pursued. For example, we could explore the use of other communication technologies such as mesh networking or satellite communications to extend the range and coverage of our application. We could also investigate methods for optimizing the performance and efficiency of the communication, such as by using compression or traffic optimization algorithms. Additionally, we could study ways to improve the user experience of our application, such as by adding features or functionality that would be useful to users in offline or low-connectivity environments.

References

[1] Tang, D. and Zhang, L. (2020) "Audio and video mixing method to enhance webrtc," IEEE Access, 8, pp. 67228–67241. Available at: https://doi.org/10.1109/access.2020.2985412.

[2] Marasevic, J. and Gavrovska, A. (2020) "Virtual reality and webrtc implementation for web educational application development," 2020 28th Telecommunications Forum (TELFOR) [Preprint]. Available at: https://doi.org/10.1109/telfor51502.2020.9306513.

[3] S. Ouya, C. Seyed, A. B. Mbacke, G. Mendy and I. Niang, "WebRTC platform proposition as a support to the educational system of universities in a limited Internet connection context," 2015 5th World Congress on Information and Communication Technologies (WICT), 2015, pp. 47-52, doi: 10.1109/WICT.2015.7489643.

[4] P. Nuño, F. G. Bulnes, J. C. Granda, F. J. Suárez and D. F. García, "A Scalable WebRTC Platform based on Open Technologies," 2018 International Conference on Computer, Information and Telecommunication Systems (CITS), 2018, pp. 1-5, doi: 10.1109/CITS.2018.8440161.

[5] Y. -N. Lien, L. -C. Chi and C. -C. Huang, "A Multi-hop Walkie-Talkie-Like Emergency Communication System for Catastrophic Natural Disasters," 2010 39th International Conference on Parallel Processing Workshops, 2010, pp. 527-532, doi: 10.1109/ICPPW.2010.77.

[6] R. Atwah, R. Iqbal, S. Shirmohammadi and A. Javadtalab, "A Dynamic Alpha Congestion Controller for WebRTC," 2015 IEEE International Symposium on Multimedia (ISM), 2015, pp. 132-135, doi: 10.1109/ISM.2015.63.

[7] W. He, Y. Huang, K. Nahrstedt and Bo Wu, "Message propagation in ad-hoc-based proximity mobile social networks," 2010 8th IEEE International Conference on Pervasive Computing and Communications Workshops (PERCOM Workshops), 2010, pp. 141-146, doi: 10.1109/PERCOMW.2010.5470617.