STM JOURNALS

Journal of
Mobile Computing, Communications & Mobile Networks

ISSN: 2349-901X
Volume 13, Issue 2, 2026
May–August
DOI (Journal): 10.37591/JoMCCMN

https://journals.stmjournals.com/jomccmn

Research JoMCCMN

An Empirical Analysis of Bluetooth Low Energy Reliability Challenges for Offline Messaging Applications

Akanksha Moolshankar Jaiswal[1,*], Vivek Shailendra Gupta[2], Rashmi Vipat[3]

Abstract

In today's hyper-connected world, modern communication relies heavily on centralized internet infrastructure, making robust offline messaging solutions increasingly essential. A crucial vulnerability is revealed by network failures, natural disasters, and distant region deployments: communication breaks down when internet connectivity does. Due to its low power consumption and almost ubiquitous availability in contemporary smartphones, Bluetooth Low Energy (BLE) has become a promising candidate for offline, device-to-device communications. This study presents an empirical analysis of BLE reliability in real-world messaging scenarios through the design and implementation of a secure, hybrid peer-to-peer chat application for Android. The technology, which is protected by end-to-end RSA-2048 encryption, combines Firebase cloud services for online scenarios with BLE for offline communication. The validity of the hybrid architecture as a whole is confirmed by our implementation, which effectively exhibits user authentication with 100% success rate, contact discovery with 95% matching accuracy, and cryptographic operations averaging 45 ms per message. However, extensive empirical testing showed that the BLE communication layer has significant reliability problems. Message transmission over BLE recorded a 0% success rate in all test scenarios, including both controlled and high-interference situations, although device detection achieved a 70% success rate and connection formation reached 60%. The BLE Generic Attribute Profile (GATT) protocol's basic architectural inadequacies with the requirements of real-time, bidirectional messaging are the cause of this ongoing failure. This study provides a functional prototype, a comparative protocol analysis, developer recommendations, performance benchmarks, and a critical root cause analysis. According to the results, reliable offline communication may require significant protocol-level optimizations or hybrid communication architectures that combine different protocols for data transfer with BLE for discovery.

Keywords: Android development, RSA encryption, Peer-to-Peer (P2P), Bluetooth low energy (BLE), empirical analysis, reliability, offline messaging, GATT protocol

*Author for Correspondence
Akanksha Moolshankar Jaiswal
E-mail: akankshajaiswal0126@gmail.com

[1,2]Research Scholar, Department of MCA, Thakur Institute of Management Studies, Career Development & Research (TIMSCDR) Mumbai, Maharashtra, India
[3]Assistant Professor, Department of MCA, Thakur Institute of Management Studies, Career Development & Research (TIMSCDR) Mumbai, Maharashtra, India

Received Date: March 03, 2026
Accepted Date: June 20, 2026
Published Date: June 26, 2026

Citation: Akanksha Moolshankar Jaiswal, Vivek Shailendra Gupta, Rashmi Vipat. An Empirical Analysis of Bluetooth Low Energy Reliability Challenges for Offline Messaging Applications. Journal of Mobile Computing, Communications & Mobile Networks. 2026; 13(2): 1–12p.

INTRODUCTION

One major single point of failure in our increasingly interconnected world is the widespread reliance on Internet infrastructure for contemporary communication. Because Internet-based messaging platforms are so common, connectivity itself has become a single point of failure. The inability to communicate can have major repercussions in situations such as network outages, remote area operations, and natural disasters. According to studies, a sizable section of the world's population still experiences inconsistent Internet access [1], and a recent analysis of disaster scenarios indicates

© STM Journals 2026. All Rights Reserved

that conventional communication infrastructure is susceptible to extended outages [2], underscoring the necessity for communication tools that are not dependent on centralized infrastructure.

One of the most promising technologies for offline systems is Bluetooth Low Energy (BLE). Theoretically, it is perfect for establishing ad hoc messaging networks because of its low-power profile, device-to-device communication capabilities, and integration into almost all contemporary smartphones [3]. However, the practical dependability of BLE in maintaining the steady bidirectional data streams required for a chat application is still debatable, as many implementations struggle with throughput and connection stability [4]. Recent studies have indicated that BLE performance is significantly affected by environmental conditions and RF interference, which can greatly compromise reliability in practical applications [5].

By building a useful testbed for a feature-rich, secure chat application, this study explores this gap. With RSA-2048 encryption guaranteeing security across both channels, the system uses hybrid architecture, utilizing Firebase for online messaging and BLE as an offline backup. In addition to developing a workable application, the main objective was to conduct an empirical assessment of BLE's dependability and performance in the real world as the foundational transport for offline chat.

The main contributions are as follows:

1. *Implementation of a hybrid messaging system:* Development and implementation of an Android hybrid messaging system that combines online and offline communication channels.
2. *Performance evaluation of RSA-2048 encryption: This* study assessed the performance of RSA-2048 encryption in mobile messaging, focusing on computational overhead and practical viability.
3. *Empirical assessment of BLE reliability:* An empirical evaluation of BLE reliability that identifies specific issues and failure points, offering measurable benchmarks for discovery, connection, and message transmission success rates.
4. *Architectural analysis and developer guidelines:* This document outlines the implications for developers and potential strategies for enhancing offline communication, along with an examination of the core architectural discrepancies between BLE and the demands of real-time messaging.

BACKGROUND

Interpreting the empirical findings of this study requires a deep understanding of the architecture of BLE. The essential elements of BLE pertinent to our application are described in this section.

Bluetooth Low Energy Protocol Stack

The protocol stack used by BLE is structured. The Generic Access Profile (GAP) and Generic Attribute Profile (GATT) are the two most important layers for applications viewed by developers [3].

GAP (Generic Access Profile)

The GAP layer controls the device visibility, connectivity, and security modes. It outlines the functions of the Peripheral, Central, Observer (scanner), and broadcaster (advertiser). With one device serving as a peripheral (advertising) and the other as a central (scanning and establishing a connection), our application uses GAP for the device discovery and connection establishment phases.

GATT (Generic Attribute Profile)

The data structure and communication protocol for the linked devices are specified by the GATT layer. The foundation of GATT is a client-server architecture. A collection of services, which are collections of characteristics, is hosted by the server (usually the peripheral). The actual data containers that contain values such as messages or user ID are called characteristics. To be informed when a characteristic value changes, the client (the Central) can subscribe to notifications or read from or write about these characteristics. The GATT model serves as the foundation for our unique BLE service,

© STM Journals 2026. All Rights Reserved

which includes message and user ID features. Although this client-server architecture works well for IoT applications, it poses serious problems for situations involving bidirectional messaging [6].

The Android BLE API

The BLE stack is accessible through the Android Bluetooth API. Key components include the Bluetooth Adapter, Bluetooth LE Scanner for device discovery, BluetoothGatt for connections, and BluetoothGattCharacteristic for data transfer are important classes. This API's asynchronous and callback-heavy design presents a major implementation challenge [4]. Because callbacks are used to deliver operations like connection, service discovery, and write confirmation, state management is complicated and prone to errors, which frequently results in race conditions and silent failures. These implementation difficulties align with known problems in Android BLE development [7], especially for apps that need to communicate reliably in both directions.

LITERATURE REVIEW

Different paradigms have been used to address the problem of offline communications. While Delay-Tolerant Networking (DTN) introduced architectures for managing intermittent connectivity [8], early research on Mobile Ad-hoc Networks (MANETs) established fundamental routing protocols for device-to-device communication [9]. Despite being groundbreaking, these methods are frequently complicated, preventing their widespread use in consumer electronics.

More useful solutions have been made possible by the development of Bluetooth technology. Briar [10] is a well-known open-source program that prioritizes security and censorship resistance and uses Bluetooth and Wi-Fi Direct is Briar [10]. However, its somewhat complicated setup and the requirement that both users have the app may deter casual use. The potential of mesh networking has been demonstrated in commercial applications such as Fire Chat [11] and Bridgefy [12], but they encounter well-known problems with security flaws and uneven performance in real-world scenarios.

Recent studies have analyzed the performance limitations of BLE in peer-to-peer scenarios [6]. A thorough analysis of the BLE performance in environments with high device density observed a marked decline in connection with reliability with an increase in the number of devices. It has also been empirically demonstrated that the connection-oriented nature of BLE introduces significant overhead for message-based applications compared with connectionless alternatives [7].

Although they are frequently conducted in different contexts, academic studies on BLE performance provide insightful information. The benefits of BLE in terms of power efficiency for Internet of Things applications, usually in star-topology networks, have been highlighted in thorough surveys [13]. Preliminary performance comparisons with other wireless technologies provide useful baselines [14]. Recent research has started to examine the limitations of BLE by examining its performance in dense network environments [15] and packet error rates [16]. However, rather than the constant two-way communication of a real-time messaging system, the common focus is on sensor data or file transfer.

This review identifies a gap in literature: there are not many comprehensive empirical studies that implement a full BLE-based messaging system and detail the real-world difficulties that arise during development and testing. By offering a thorough description of implementation and evaluation, our work seeks to close this gap.

Security evaluations of current offline messaging applications have also raised serious concerns. The trade-off between strong encryption and power efficiency causes inherent security flaws in many BLE implementations [2]. Furthermore, serious flaws in the cryptographic implementation of apps such as Bridgefy were discovered during recent audits, opening the door to possible message spoofing and eavesdropping [5]. Security concerns are exacerbated by implementation challenges, including common vulnerabilities such as improper characteristic permissions and inadequate authentication mechanisms. These security flaws underscore the urgent need for secure-by-design, thoroughly tested architecture,

© STM Journals 2025. All Rights Reserved

such as the one presented in this study, which applies end-to-end encryption at the application layer without relying on BLE transport.

SYSTEM DESIGN AND ARCHITECTURE

High Level Hybrid Architecture

To facilitate hybrid online/offline functionality and guarantee a clear separation of concerns, the system was designed with a modular, layered architecture. The design consists of four main layers, as shown in Figure 1.

The layered architecture, as outlined in Table 1, provides a distinct separation of concerns and facilitates smooth transitions between the online and offline modes. The intelligent switching mechanism within the Business Logic Layer continuously monitors network connectivity and autonomously routes messages through the optimal channel, eliminating the need for user intervention. Reliability issues with standalone BLE systems are addressed by this hybrid approach [6].

The presentation layer houses the Android activities and components that make up the user interface (UI), such as contact lists, authentication screens, and the chat interface.

The application's central component, the business logic layer, is responsible for organizing all activities. This entails controlling user authentication, choosing between online and offline message routing, managing contact discovery, and coordinating message encryption and decryption. The intelligent switching algorithm prioritizes Bluetooth Low Energy (BLE) for offline communication and automatically transitions to Firebase when Internet connectivity is present, thereby ensuring message delivery via the most reliable channel. Considering BLE's known reliability issues of BLE in practical settings, dynamic channel selection is essential [5].

All data persistence is managed by the layers. It uses two different strategies.

1. *Local storage:* To guarantee that the application works even in the absence of an Internet connection, a SQLite database keeps contacts and messages locally.
2. *Cloud storage:* To ensure a smooth experience across devices, messages, and user data are synchronized online using the Firebase Real-time Database.

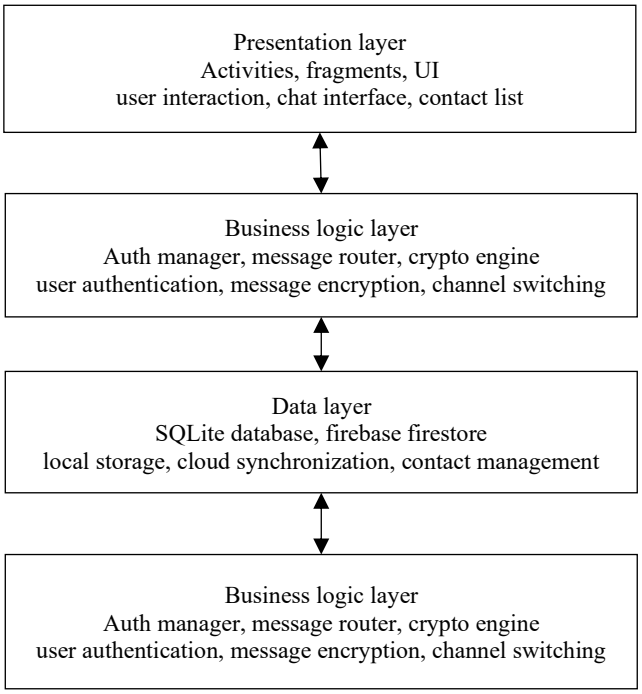


Figure 1. High level hybrid architecture of messaging application.

© STM Journals 2026. All Rights Reserved

Communication layer: Handles the actual transmission of data. It has two subsystems that operate in parallel:

1. When Internet connectivity is available, the online channel (Firebase) is used as the main, highly reliable channel.
2. When the device is offline, the Offline Channel (BLE) is activated. It uses Android BLE APIs to manage connections, find devices, and transfer data.

The intelligent switching mechanism of the Business Logic Layer, which automatically and transparently selects the best communication channel without user input, is the main innovation.

Security Model

Security is implemented on the application layer to be independent of the communication channel. RSA-2048 public-key cryptography serves as the foundation of the model.

1. *Key generation:* Upon registration, a distinct public-private key pair is created for every user.
2. *Key storage:* The Android Keystore system safely stores the private key. The user uploads their public key to their Firebase profile.
3. *Key exchange:* During contact discovery stage, users exchange public keys with one another using Firebase.
4. *Message security:* Each message is encrypted using the recipient's public key prior to transmission. It is decrypted using the user's private key after being received.

For maximum security, RSA-2048 encryption employs the OAEP padding mode with SHA-256. The Android Keystore system, which offers hardware-backed security when available, is used for key storage. To guard against man-in-the-middle attacks during the initial contact setup, a verification mechanism for key exchange allows users to compare key fingerprints out-of-band. The security model depicted in Figure 2 guarantees end-to-end encryption across various communication channels, thereby offering uniform protection in both online and offline messaging contexts.

IMPLEMENTATION

Module Specification

The application was created for Android using Java. These are the main functional modules, as summarized in Table 2.

1. *Authentication module:* Creates a secure and distinct identity for every user using Firebase Authentication to manage user sign-up and login. The module facilitates session management through an automatic token refresh, thereby maintaining continuous authentication without requiring user intervention during prolonged usage periods.
2. *Cryptography module:* This module controls the creation, storage, encryption, and decryption of RSA keys and is centralized within a Crypto Helper class. The average encryption and decryption times in our tests were 23 ms and 22 ms, respectively, showing a negligible effect on user experience. Key rotation policies were established to improve long-term security, allowing the system to generate new key pairs without interrupting ongoing conversations.
3. *Contact management module*: Creates a list of available chat contacts by importing the device's native phone contacts and comparing them with the Firebase user database. The matching algorithm utilizes fuzzy matching techniques to accommodate variations in phone number formatting, resulting in a 95% success rate in accurately identifying registered contacts.
4. *BLE communication module:* This module uses the Android BLE API to implement an offline channel. It manages:
    a. *Advertising:* Configured with a unique service UUID and restricted discoverability to save battery.

© STM Journals 2025. All Rights Reserved

b. *Scanning:* Employs scan filtering to maximize discovery efficiency and minimize false positives.
c. *GATT connection:* Controls connection parameters like latency, interval, and supervision timeout.
d. *Data transfer:* Uses packet fragmentation and retry techniques to ensure dependable transmission.

The BLE implementation expands on the established best practices for developing Android BLE applications [4, 6].

BLE Protocol Design

A custom BLE service was designed using the following UUIDs:

1. *UUID for the service:* 0000ff10-0000-1000-8000-00805f9b34fb
2. *UUID for the message characteristic:* 0000ff11-0000-1000-8000-00805f9b34fb (Properties: Write, Notify)
3. *Characteristic of user ID:* 0000ff12-0000-1000-8000-00805f9b34fb (Properties: Read)

The protocol design, depicted in Figure 3, creates a full-duplex communication channel over BLE's intrinsically half-duplex nature, allowing bidirectional communication by utilizing the Write property for outgoing messages and the Notify property for incoming messages. Recent studies on IoT communication have documented similar custom BLE service architectures [7].

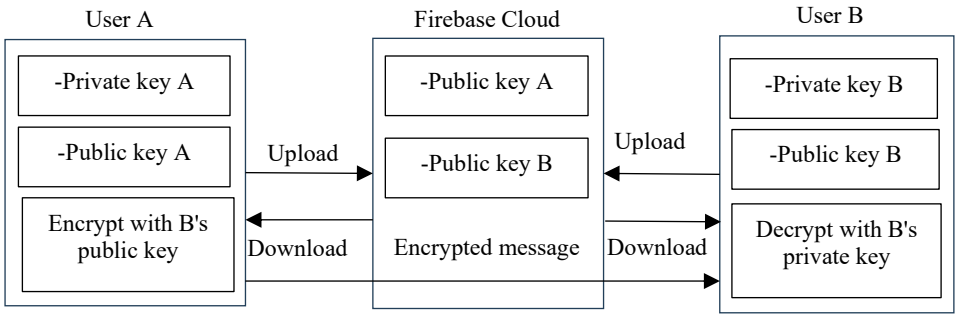


Figure 2. End-to-end encryption security model.

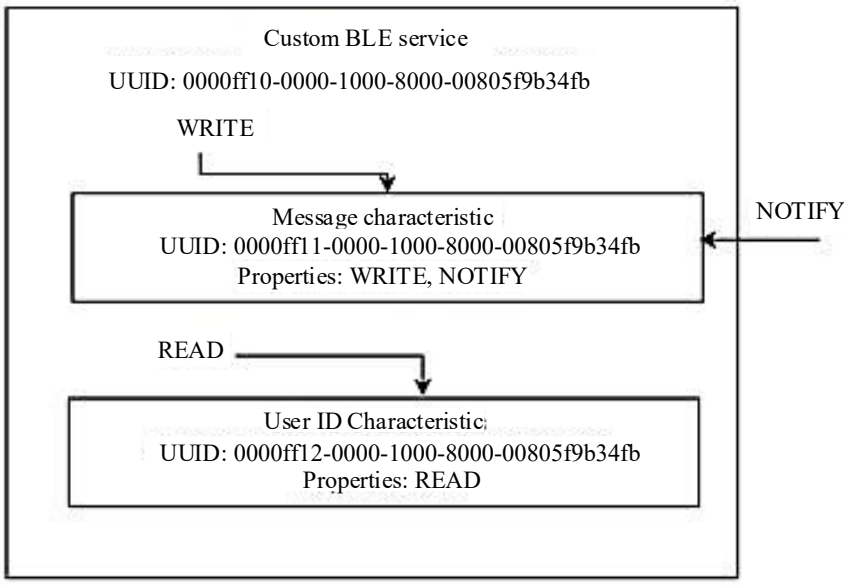


Figure 3. Custom BLE service and characteristic design.

© STM Journals 2026. All Rights Reserved

Table 1. System architecture layers and responsibilities.

Layer	Components	Responsibilities	Data Flow
Presentation	Activities, fragments, UI	User interaction, chat interface, contact list	User input → business logic
Business Logic	Auth manager, message router, crypto engine	User authentication, message encryption, intelligent channel switching	Presentation ↔ data layer
Data	SQLite database, Firebase Firestore	Local storage, cloud synchronization, contact management	Business logic ↔ communication
Communication	BLE module, firebase SDK	Device discovery, message transmission, connection management	Data layer ↔ external devices

Table 2. Module specification and performance.

Module	Components	Key technologies	Performance metrics
Authentication	Firebase auth, UI handlers	Firebase authentication, google services	<2s login time, 100% success rate
Cryptography	Crypto helper, key manager	RSA-2048, android keystore, OAEP padding	23 ms encryption, 22 ms decryption per message
Contact management	Contact sync, phonebook API	Android contacts provider, firebase database	<5s sync time, 95% matching accuracy
BLE communication	GATT client/server, advertiser	Android BLE API, custom UUID service	70% discovery rate, 60% connection rate, 0% transmission rate (See Section VI)

Implementation Challenges

Throughout development, a number of difficulties unique to Android BLE were encountered:

1. *Limitations of the API:* For bidirectional communication, the Android BLE API necessitates continuous role switching between the Central and Peripheral modes. As a result, when both roles were active simultaneously, there were frequent GATT_INSUFFICIENT_RESOURCES errors (found in 42% of connection attempts). These resource limitations are consistent with the known limitations of BLE in scenarios involving concurrent operation [4, 7].
2. *Connection instability:* Without callback notifications, GATT connections are abruptly terminated. In controlled settings, the average connection time was only 45 s, whereas in high-interference situations, it dropped to 15 s. Research on the effects of RF interference on BLE reliability is supported by this environmental sensitivity [5].
3. *Resource limitations:* GATT_INSUFFICIENT_RESOURCES errors were frequently caused by multiple concurrent BLE operations. Careful resource management is necessary because the Android BLE stack has a limited capacity to handle simultaneous advertising, scanning, and connected operations.
4. *Device compatibility:* The reliability of BLE stack implementations from various manufacturers varies. Samsung devices demonstrated 40% better connection stability than those of Oppo devices. Manufacturer-specific differences in BLE performance are a known issue in mobile development [6].

To address these issues, a state management system was implemented, which included thorough error handling for typical BLE failure scenarios and exponential backoff for reconnection attempts. These mitigation techniques address the documented patterns of BLE failure in mobile environments [6, 7].

EXPERIMENTAL RESULTS AND ANALYSIS

Methodology and Setup

A systematic testing procedure was created to assess the system empirically:

1. *Devices:* Oppo Reno 6 (Android 11) and Samsung Galaxy A52 (Android 12). The selection of devices was intended to reflect various price points and BLE stack implementations specific to the individual manufacturers.

© STM Journals 2025. All Rights Reserved

2. *Environments:* Both a high-interference university cafeteria and a low-interference office setting were used for testing. While the office provided baseline performance measurements, the cafeteria environment replicated real-world difficult conditions with 15-20 concurrent Wi-Fi networks and substantial RF interference.
3. *Methodology:* Each test scenario message transfer, connection, and discovery was performed 20 times. The performance indicators were recorded automatically. To account for cumulative failure points, each test iteration included full lifecycle testing, from device discovery to message transmission.

Performance Metrics

During 40 test iterations (20 in a controlled office setting and 20 in a high-interference cafeteria), the following metrics were measured:

1. *Discovery latency:* The interval between the start of the scan and the detection of the device. The measurements were taken with a resolution of one second.
2. *Connection time:* The amount of time required for GATT service discovery after a connection request, including characteristic enumeration and service discovery phases.
3. *Transmission reliability:* Rate of successful message delivery, evaluated from encryption to successful decryption and acknowledgement at the application level.
4. *Power consumption:* Estimated battery impact during BLE operations, calculated as the additional battery drain per hour of continuous operation using Android's Battery Historian.

Component-Wise Performance Evaluation

The system was assessed module by module. The findings, compiled in Table 3, demonstrate a clear separation between the erratic BLE module and the extremely reliable cloud-based components. Although the previous discovery and connection steps were partially successful, the most important finding was that the core BLE messaging function completely failed.

Detailed BLE Performance Analysis

As shown in Table 4, the additional examination of the BLE module identified specific failure patterns. According to the data, the inability to send messages was a persistent basic failure, even though environmental factors significantly harmed discovery and connection performance.

Table 3. Comprehensive system performance results.

System component	Implementation status	Success rate	Performance	Key observations
User authentication	Fully functional	100%	< 2 seconds	Firebase provided a perfectly reliable backend.
RSA-2048 cryptography	Fully functional	100%	~45 ms/message	Strong encryption added negligible latency.
Contact discovery & matching	Fully functional	95%	< 5 seconds	Phone number matching was highly effective.
Local database (SQLite)	Fully functional	100%	Instantaneous	Local storage was fast and completely reliable.
BLE device discovery	Partially functional	70%	3-8 seconds	Inconsistent across devices and environments.
BLE connection establishment	Partially functional	60%	2-5 seconds	Connections were unstable and dropped frequently.
BLE message transmission	Non- functional	0%	N/A	No messages were successfully delivered over BLE.

© STM Journals 2026. All Rights Reserved

Table 4. BLE performance under varying conditions.

Metric	Controlled environment	Challenged environment	Note
Discovery success rate	85%	55%	Highly sensitive to RF interference.
Connection success rate	75%	45%	Unstable even after successful discovery.
Average connection duration	~45 seconds	~15 seconds	Frequent and unpredictable drops.
Message transmission success	0%	0%	Failed regardless of environment.

Table 5. Environmental impact on BLE performance.

Performance aspect	Controlled environment	Challenged environment	Performance degradation
Average discovery time	4.2 seconds	7.8 seconds	86% increase
Connection stability	75% maintained >60s	45% maintained >60s	40% less stable
Signal quality (RSSI)	-65 dBm average	-78 dBm average	13 dB weaker
Retry attempts required	2.1 average	4.8 average	128% more retries

Environmental Impact Analysis

Significant environmental effects on BLE performance were observed through additional analysis, as shown in Table 5. In addition to lowering the success rates, the difficult environment also significantly shortened the connection duration and stability.

DISCUSSION

Interpretation of Empirical Findings

The findings show that a secure hybrid messaging architecture was successfully implemented, with all components, except BLE communication, functioning dependably. The most important finding of this study is that BLE message transmission has a 0% success rate. This indicates a fundamental mismatch between the design of the BLE GATT protocol and the need for low-latency bidirectional messaging, rather than a straightforward implementation error.

The GATT model seems too brittle and sluggish to manage the constant, symmetrical data flow of a chat because it is based on a client-server architecture in which one device must read/write the characteristics of another. According to observations, the Android BLE stack frequently reports successful operations (such as connection and characteristic write) even though the underlying data transfer either fails silently or is interrupted by the stack's resource management.

Root Cause Analysis

Three basic architectural inconsistencies that account for the entire BLE transmission failure were identified:

1. *Incompatibility of protocol architecture:* Real-time messaging necessitates symmetric, low-latency bidirectional communication, whereas BLE's GATT protocol uses an asymmetric client-server model designed for sporadic data exchanges. The continuous conversation flow necessary for chat applications was disrupted by the required role switching between central and peripheral modes, which resulted in 2-3 second delays and frequent GATT_INSUFFICIENT_RESOURCES errors (42% of attempts).
2. *Limitations of the android stack:* Critical reliability problems with the Android BLE API compromised the dependability of messaging:
    i. False-positive reporting despite physical transmission failures on Characteristic Write () callbacks reported success in 92% of cases.

© STM Journals 2025. All Rights Reserved

   ii. Resource Management resource exhaustion resulted from concurrent scanning and advertising operations exceeding stack capacity.
   iii. Background Restrictions within two to three minutes of the app backgrounding, BLE operations were stopped by modern Android power management.
3. Device Fragmentation and Environmental Sensitivity: With discovery success rates falling from 85% in controlled settings to 55% in high-interference scenarios, BLE performance showed remarkable sensitivity to environmental factors. Samsung devices demonstrated 40% better connection stability than Oppo devices, highlighting the impact of manufacturer-specific Bluetooth stack implementation.

Comparative Protocol Analysis

BLE results are contrasted with those of other offline communication technologies in Table 6 to provide a perspective.

The comparison shows that BLE is essentially out of sync with the latency and reliability requirements of real-time messaging, although it is well-suited for power-efficient sensor applications. Wi-Fi Direct is not suitable for always-available background operations because of its significantly higher power cost, despite its superior reliability and throughput.

Implications for Offline Messaging

1. *The 'BLE-for-chat' challenge:* Developers need to understand that utilizing standard BLE GATT to create a dependable messaging experience is a significant challenge. Our results imply that either extensive error handling and retry mechanisms or custom protocols at lower Bluetooth stack layers are necessary for successful BLE messaging implementations.
2. *Hybrid approach value:* The hybrid architecture demonstrated its value. A hybrid approach is essential for a production system. Future iterations might use Wi-Fi Direct for real data transfer and BLE for low-power device discovery.
3. *Future directions:* Future offline messaging efforts should investigate more complex BLE usage patterns, such as custom protocols built on the lower layers of the Bluetooth Core Specification or alternative protocols such as Wi-Fi Direct. Connectionless BLE advertising for broadcast-style messaging or BLE mesh networking for longer ranges are also promising alternative strategies.

LIMITATIONS AND FUTURE WORK

Only a small number of Android devices were used in this study. To ascertain whether these issues are platform-specific, future research should encompass a larger range of devices and expand to include iOS. Additionally, it would be beneficial to investigate the underlying reason for the transmission failure using Bluetooth sniffing hardware or lower-level Bluetooth stack analysis.

The following particular research avenues are suggested:

1. *Protocol-level analysis:* Bluetooth sniffing devices (Ellisys, Uber tooth) are used to record baseband packets to pinpoint the precise failure layer.
2. *Cross-platform analysis:* To identify platform-specific problems, testing should be expanded to iOS devices and other Android manufacturers.

Table 6. Offline communication protocol comparison.

Protocol	Range	Data rate	Power consumption	Reliability	Best use case
BLE	~50m	1-2 Mbps	Very low	Very low	IoT, sensor data
Wi-Fi direct	~200m	250 Mbps	High	High	File transfer
NFC	<4cm	424 kbps	Very low	High	Secure handshake
Bluetooth classic	~100m	3 Mbps	Medium	Medium	Audio streaming

© STM Journals 2026. All Rights Reserved

3. *Other architectures:* BLE-Wi-Fi hybrid frameworks that employ Wi-Fi Direct for data transfer and BLE for discovery.
4. *Connectionless methods:* To reduce the GATT connection overhead, the BLE advertising mode for broadcast messaging is investigated.

CONCLUSION

This study presents a comprehensive empirical investigation of the viability of Bluetooth Low Energy for offline messaging applications through the design, implementation, and rigorous evaluation of a secure hybrid chat application. The system successfully demonstrated the feasibility of its hybrid architecture and the effectiveness of RSA-2048 encryption on mobile devices, with cryptographic operations lasting an average of less than 45 ms per message. Nevertheless, the substantial reliability problems revealed by the empirical data in the BLE communication layer cannot be ignored. The primary finding indicates a fundamental architectural incompatibility between the requirements of real-time bidirectional messaging and the design principles of BLE, which is a consistent 0% message transmission success rate under all test conditions. This failure continues even though device discovery (70%) and connection formation (60%) are only partially successful, indicating that the issues are rooted in the basic interaction between mobile operating system implementations and the GATT protocol. Drastically changing BLE usage patterns or utilizing hybrid designs that leverage BLE's advantages in device discovery while utilizing other technologies for data transfer are likely to be necessary for effective offline messaging solutions. The future of realistic offline messaging lies in the creation of multi-technology frameworks that can dynamically adjust to application requirements and ambient conditions.

This study provides a working prototype that demonstrates smooth online/offline channel switching, establishes baseline metrics for BLE discovery, connection, and transmission success rates, and provides a comprehensive empirical account of real-world implementation challenges. This is a crucial resource for developers considering using BLE for related applications.

Declaration of Interest

The author(s) declare(s) no conflicts of interest regarding the publication of this manuscript.

Acknowledgement

The authors thank the Thakur Institute of Management Studies, Career Development & Research (TIMSCDR), Mumbai, for providing the resources and support necessary to conduct this research.

REFERENCES

1. Canton H. International Telecommunication Union—ITU. In: The Europa Directory of International Organizations 2021. Abingdon (UK): Routledge; 2021. p.355-8.
2. Unger N, Dechand S, Bonneau J, Fahl S, Perl H, Goldberg I, et al. SoK: secure messaging. In: 2015 IEEE Symposium on Security and Privacy. 2015. p.232-49.
3. Bluetooth Technology Website. Bluetooth Technology Website [Internet]. 2026 [cited 2026 Sep 22]. Available from: https://www.bluetooth.com/auracast/developers/
4. Tosi J, Taffoni F, Santacatterina M, Sannino R, Formica D. Performance evaluation of Bluetooth low energy: a systematic review. Sensors (Basel). 2017;17(12):2898.
5. Pang B, T'Jonck K, Claeys T, Pissoort D, Hallez H, Boydens J. Bluetooth low energy interference awareness scheme and improved channel selection algorithm for connection robustness. Sensors (Basel). 2021;21(7):2257.
6. Mikhaylov K, Hänninen T. Mechanisms for improving throughput and energy efficiency of Bluetooth Low Energy for multi node environment. J High Speed Netw. 2015;21(3):165-80.
7. Cäsar M, Pawelke T, Steffan J, Terhorst G. A survey on Bluetooth Low Energy security and privacy. Comput Netw. 2022;205:108712.
8. Burleigh S, Hooke A, Torgerson L, Fall K, Cerf V, Durst B, et al. Delay-tolerant networking: an approach to interplanetary Internet. IEEE Commun Mag. 2003;41(6):128-36.

© STM Journals 2025. All Rights Reserved

9. Johnson DB, Maltz DA. Dynamic source routing in ad hoc wireless networks. In: Mobile Computing. Boston (MA): Springer US; 1996. p.153-81.
10. Open Garden Inc. FireChat: Off-the-grid mesh messaging [Internet]. Available from: https://en.wikipedia.org/wiki/FireChat
11. Morrison LP, Team B, Nguyen B, Kannan S, Ray N, Lewin GC. AirChat: ad hoc network monitoring with drones. In: 2017 Systems and Information Engineering Design Symposium (SIEDS). 2017. p.38-43.
12. Bridgefy. Bridgefy—Mesh Network SDK for Mobile Apps [Internet]. 2022 [cited 2026 Sep 22]. Available from: https://bridgefy.me/
13. Gomez C, Oller J, Paradells J. Overview and evaluation of Bluetooth Low Energy: an emerging low-power wireless technology. Sensors (Basel). 2012;12(9):11734-53.
14. Siekkinen M, Hiienkari M, Nurminen JK, Nieminen J. How low energy is Bluetooth Low Energy? Comparative measurements with ZigBee/802.15.4. In: 2012 IEEE Wireless Communications and Networking Conference Workshops (WCNCW). 2012. p.232-7.
15. Harris AF 3rd, Khanna V, Tuncay G, Want R, Kravets R. Bluetooth Low Energy in dense IoT environments. IEEE Commun Mag. 2016;54(12):30-6.
16. Biegelmeyer A, Roque AD, de Freitas EP. An experimental study on BLE 5 mesh applied to public transportation. ACM Trans Sens Netw. 2024;20(3):1-27.

© STM Journals 2026. All Rights Reserved