An Open, Verifiable, and Performant Architecture for Resilient Mobile Communication

Project Progress Update

Supervisor: Dr. Ali Sayyed

Hassaan Anwar (22P-9160)

Muhammad Aais Rabbani (22P-9164)

Department of Computer Science

FAST-NUCES, Peshawar

Date: 26th October 2025

Outline

Project Recap: Introduction & Motivation

Goal: Design an open, protocol-agnostic architecture for resilient mobile communication when infrastructure fails or is censored.

Core Features: A multi-hop local mesh network and an opportunistic "gateway" to bridge offline groups to the internet.

Motivation: Existing centralized systems are fragile. Closed-source alternatives like Bridgefy lack the verifiability and trust needed for critical scenarios.

Competitor Review

App / Project	Strength	Weakness
FireChat	Uses Bluetooth and Wi-Fi P2P for offline messaging through a self-forming mesh network. It gained popularity for working without internet or SIM registration and functions effectively in dense environments.	Relies heavily on user density for stable connectivity. Limited message delivery reliability in sparse areas and no built-in support for network bridging or modern APIs.
Bridgefy	Provides offline messaging via Bluetooth and offers an SDK for developers to embed mesh messaging features in their apps. It is simple and quick to set up.	Bluetooth-only communication restricts scalability and can cause high battery usage during long sessions. Performance degrades with many simultaneous users.
Briar	Offers secure offline and online communication using Bluetooth, Wi-Fi, or the Tor network. Strong focus on privacy with end-to-end encryption and decentralized design.	More suited for privacy-focused users than real-time communication. Syncing messages can be slow, and setup complexity reduces usability for general audiences.
Serval Mesh	Creates a peer-to-peer Wi-Fi mesh network that supports voice, text, and file sharing. It is open-source and useful for community or disaster communication.	Requires Wi-Fi interface configuration, making setup complex for average users. Lacks integration with modern mobile APIs and automatic reconnection mechanisms.

Literature Review

Paper Title	Year	Key Contributions	Limitations / Gaps (Addressed in Our Work)
Dolphin: A Cellular Voice Based Internet Shutdown Resistance System	2023	Presents a system that enables data transmission over voice calls using a reliable, TCP-like protocol with encryption and compression. Demonstrates resilience during internet shutdowns.	Focuses solely on cellular voice channels without integrating with ad hoc or mesh-based communication. Lacks dynamic peer discovery and hybrid gateway support.
A Framework for Multi-Hop Ad Hoc Networking over Wi-Fi Direct with Android Smart Devices	2021	Proposes a framework for enabling multi-hop routing over Wi-Fi Direct on Android devices, achieving internet-free communication between peers.	Offers no mechanism for automatic peer discovery or transition between offline and online modes. Scalability and gateway bridging are not addressed.
Device-to-Device Communications with Wi-Fi Direct: Overview and Experimental Evaluation	2013	Provides a detailed experimental analysis of Wi-Fi Direct's performance, including device discovery time, connection setup delay, and power consumption metrics.	Evaluation limited to controlled single-hop scenarios; no consideration for message persistence, mesh expansion, or integration with cloud backends.

Problem Statement

Scenario: During an infrastructure failure (e.g., natural disaster) on a campus, mobile users are isolated from central services and each other, preventing essential communication.

Technical Challenge: Build a resilient system that respects mobile constraints (battery, churn) and allows for reproducible, auditable performance measurements on a small-scale testbed.

Project Progress To Date

Technology stack

- Kotlin (Android prototype)
- Google Nearby Connections Framework

Proof-of-Concept (PoC) testbed

- Working prototype demonstrating the core P2P link
- Device discovery and connection management

System architecture

- High-level components defined
- Core mesh logic separated from pluggable transport layer
- Designed for protocol-agnostic transport swapping

Milestones achieved

- Literature review complete
- Prototype testbed validated on small devices

Diagrams

- High-Level System Architecture
- Sequence Diagram
- Activity Diagram
- Swimlane Diagram
- Use Case Diagram

Test Cases

ID	Category	Test Description	Expected Result
TC-1	Basic Connection	Establish a peer-to-peer connection between two nearby devices.	Devices successfully connect using Bluetooth/Wi-Fi Direct.
TC-2	Device Discovery and Advertisement	Verify that a device can advertise its presence and discover nearby peers.	Devices are able to detect each other through the discovery and advertisement process.
TC-3	Mesh Messaging	Send message between two nearby nodes.	Message successfully delivered within mesh range.
TC-4	Multi-Hop Delivery	Send message from Node A $\rightarrow$ C through intermediate Node B.	Message reaches Node C via Node B (multi-hop verified).
TC-5	Store-and-Forward	Send message while receiver is offline.	Message stored locally and delivered once receiver reconnects.
TC-6	Gateway Bridging	Send message from offline mesh node to online user through gateway.	Message routed successfully via gateway to remote user.
TC-7	Gateway Outage	Gateway loses internet temporarily.	Messages buffered and synced once connectivity resumes.
TC-8	Security Check	Attempt to intercept transmitted message.	Data appears encrypted and unreadable.
TC-9	Session Persistence	Restart app and resume session.	Previous messages persist; session restored successfully.
TC-10	Fault Tolerance	Random node failure during transmission.	Message rerouted through alternate path if available.

Challenges & Risks

Android Background Process Limitations:

Modern Android versions are extremely aggressive in killing background services to save battery.

This poses a significant risk to maintaining persistent P2P connections and requires careful implementation of foreground services and battery optimization exceptions.

Google Nearby API Dependencies:

The API requires Google Play Services, which may not be available on all devices or in all regions, slightly conflicting with our goal of maximum accessibility. This is an accepted trade-off for the FYP prototype.

Next Steps (Immediate Priorities)

Implement Multi-Peer Mesh Formation:

Extend the current P2P link to support a true M-to-N (many-to-many) cluster topology.

Develop the logic for nodes to discover and connect to multiple peers simultaneously.

Establish Connection Reliability & Stability:

Implement robust handling for unexpected disconnections and graceful peer reconnection.

Ensure the mesh remains stable as devices enter and leave the network (churn).

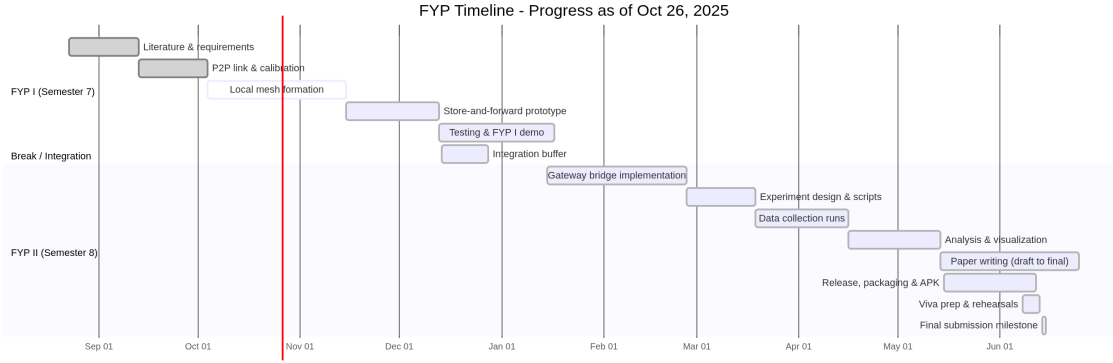
Conduct Foundational Performance Benchmarking:

Design and execute experiments to measure key baseline metrics: connection range, latency, and throughput in realistic scenarios (e.g., with obstacles).

Updated Timeline (Gantt)

FYP1: Prototype P2P link, local mesh, calibration and baseline tests.

FYP2: Gateway bridging, store-and-forward, experiments, analysis and paper.



Team Work

Equal contribution across Requirements → Design → Implementation → Testing.

Workflow: sequential handoff — one member leads a phase while the other supports; integration and testing run concurrently when needed.

Roles:

Hassaan Anwar (22P-9160) — architecture, P2P link, core APK.

Muhammad Aais Rabbani (22P-9164) — gateway/integration, experiment harness, data analysis.

Contributions are tracked via concise git commits and short log notes for traceability.

Thank You!

We welcome your questions and feedback.

Supervisor: Dr. Ali Sayyed

Hassaan Anwar (22P-9160) • Muhammad Aais Rabbani (22P-9164)