International Journal of General Engineering and Technology (IJGET)
ISSN (P): 2278–9928; ISSN (E): 2278–9936
Vol. 15, Issue 1, Jan – Jun 2026; 15–26

International Academy of Science, Engineering and Technology
Connecting Researchers; Nurturing Innovations

AIRLINK: A DECENTRALIZED, SELF-HEALING MESH NETWORKING FRAMEWORK FOR RESILIENT MOBILE COMMUNICATION

Ketan Anand[1], Saumya Shukla[2], Harsh Pal[3], Stuti Tiwari[4] & Deepika Verma[5]

[1]Associate Professor, Axis Institute of Technology and Management, Kanpur, Uttar Pradesh, India

[2,3,4,5]Axis Institute of Technology and Management, Kanpur, Uttar Pradesh, India

ABSTRACT

In an increasingly connected world, the fragility of centralized communication infrastructures remains a critical vulnerability. Natural disasters, infrastructure failures, and censorship often render traditional cellular and internet-based messaging systems inoperative. This research presents AirLink, a novel decentralized mesh networking application built on the Flutter framework, designed to provide resilient, peer-to-peer communication without reliance on centralized servers or backhaul connectivity. AirLink leverages heterogeneous radio interfaces, including Bluetooth Low Energy (BLE) and Wi-Fi Direct, to form a dynamic, self-healing mesh topology. Unlike traditional mobile ad-hoc networks (MANETs), AirLink introduces an Adaptive Intelligence (AI) Layer that optimizes discovery frequencies based on battery health and motion telemetry. Furthermore, the system implements a Gossip-based Reputation Framework to mitigate the impact of malicious or unreliable nodes, ensuring high-integrity routing in trustless environments. The technical architecture utilizes a modified Dijkstra's algorithm for multi-hop routing, combined with a robust Store-and-Forward mechanism that handles intermittent connectivity. Our evaluations demonstrate that AirLink achieves significant improvements in mesh stability and message delivery rates compared to standard flooding-based protocols, while maintaining a sustainable battery profile for long-term emergency deployments.

KEYWORDS: *Mesh Networking, Decentralized Systems, Gossip Protocols, P2P Communication, Mobile Resilience, Trust & Reputation, Store-and-Forward*

Article History

Received: *19 Apr 2026* | ***Revised:*** *20 Apr 2026* | ***Accepted:*** *22 Apr 2026*

INTRODUCTION

Motivation

The paradigm of modern communication is built upon the assumption of pervasive, high-bandwidth infrastructure. From fiber-optic backbones to 5G cellular arrays, contemporary society relies on a centralized topology where data is routed through controlled, managed gateways. However, this centralization introduces a single point of failure increasingly exposed during catastrophic events. Natural disasters such as earthquakes, hurricanes, and floods frequently devastate physical telecommunications infrastructure, leaving survivors and first responders in communication blackouts.

Beyond physical destruction, the sociopolitical landscape has highlighted the need for decentralized communication as a tool for digital sovereignty. In regions experiencing political instability or civil unrest, centralized internet access is often weaponized through selective shutdowns or pervasive surveillance. A decentralized, peer-to-peer (P2P) network that operates independently of the global internet offers a last-mile solution for maintaining the flow of information, preserving both privacy and freedom of association.

Problem Statement

Existing mobile mesh networking solutions often face a trilemma of challenges: Resilience vs. Battery Life vs. Scalability.

- **Resilience:** Many current apps rely on simple flooding protocols. While robust, flooding quickly saturates shared bandwidth, leading to high collision rates and message loss in dense environments.
- **Battery Life:** Keeping Wi-Fi and Bluetooth radios active for discovery is power-intensive. A mesh networking app that drains the battery in hours is impractical for emergency use.
- **Scalability:** Maintaining an accurate routing table in a dynamic mesh where nodes constantly move is computationally expensive. As the network grows, the overhead of heartbeat signals and topology updates can exceed actual data throughput.

Research Objectives

This research aims to bridge the gap between theoretical MANET protocols and practical mobile implementation. Specifically, the objectives are:

- To design a multi-layered mesh architecture that abstracts the complexities of heterogeneous radio protocols.
- To implement an Adaptive Discovery Mechanism that modulates radio activity based on node metadata.
- To develop a Weighted Multi-Hop Routing Algorithm that prioritizes backbone nodes (nodes with high power and stability).
- To integrate a Reputation-Based Trust System that allows the network to self-regulate without central authority.

Contributions

The primary contributions of this work include:

- **AirLink Framework:** A complete, open-source implementation of a mesh networking stack for Flutter.
- **Gossip Reputation Protocol:** A novel method for propagating peer quality scores across a mesh to optimize routing decisions.
- **Stateful Reconnection Manager:** A system designed to handle frequent link disruptions in mobile P2P environments, ensuring message integrity through store-and-forward buffers.
- **Empirical Performance Analysis:** A comprehensive evaluation of trade-offs between mesh density, message latency, and power consumption.

BACKGROUND & RELATED WORK

Evolution of Mesh Networking

The conceptual architecture of mesh networking traces its origins back to the packet radio experiments of the 1970s. The DARPA-funded PRNET (Packet Radio Network) was designed as a decentralized communication system capable of operating in the presence of physical infrastructure destruction [1]. Unlike the hierarchical telephone networks of the time, PRNET treated every radio node as a potential router, establishing the cooperative relay paradigm.

Throughout the 1980s and 90s, this evolved through the work of amateur radio operators (utilizing the AX.25 protocol) and academic researchers [6]. The transition to mobile ad-hoc networks (MANETs) was accelerated by the miniaturization of Wi-Fi and Bluetooth chipsets. However, as the number of nodes increased, the problem of Broadcast Storms — where network maintenance overhead consumes the entirety of available bandwidth — became the primary technical hurdle [1]. AirLink positions itself as the next iteration, moving from static routing to an Adaptive, Telemetry-Aware routing model.

Taxonomy of MANETs

Theoretical research into MANETs has historically divided routing protocols into three categories:

- **Proactive (Table-Driven):** Protocols such as OLSR and DSDV require every node to maintain a complete routing table for every other node, updated through periodic broadcasts [14]. This ensures near-zero latency but generates high background traffic causing constant table churn in mobile settings.
- **Reactive (On-Demand):** Protocols like AODV and DSR only establish a path when data needs to be sent [2], [15]. While battery-efficient, the initial Time-to-First-Byte can be high, often taking several seconds in sparse networks.
- **Hybrid Approaches:** AirLink adopts a Gossip-State Hybrid, maintaining a proactive local view of 1-hop and 2-hop neighbours while utilizing reactive, opportunistic Store-and-Forward for long-distance multi-hop targets.

Gossip Protocols and Information Dissemination

Gossip-based dissemination mimics the mathematical models of biological disease spread to ensure reliable data synchronization in large-scale distributed systems. Reference [13] identifies two primary gossip mechanisms: Anti-Entropy (nodes reconcile entire data state) and Rumour Mongering (nodes share new information with k random neighbours until the rumour is "old") [13]. AirLink utilizes a Bounded Rumour Monger approach for its Reputation and Topology updates [3]. By limiting the fan-out and injecting a decay factor, the mesh achieves 99.9% consistency in O(log N) time without saturating the local spectrum.

Existing Decentralized Messaging Solutions

- **Bridgefy:** Uses BLE for proximity-based messaging [16]. Its reliance on a proprietary, closed-source SDK prevents independent security audits. Routing performance in sparse environments is highly variable.
- **Briar:** A gold standard for security and privacy, utilizing Tor and sync-puzzles. However, its Security-First architecture results in extreme battery consumption and background throttling on modern mobile OSes.

- **FireChat (Legacy):** Used during the 2014 Hong Kong protests. Now discontinued, leaving a void for a robust, open-source alternative bridging Proximity Chat and True Multi-Hop Mesh.

AirLink distinguishes itself by introducing the Adaptive Intelligence Layer, which proactively manages radio states and routing weights based on real-world device telemetry — a dynamic capability fundamentally absent in static protocol implementations.

SYSTEM ARCHITECTURE

Flutter-Based Service-Oriented Design

The architectural philosophy of AirLink is rooted in decoupling and modularity. AirLink achieves isolation through a Service-Oriented Architecture (SOA) implemented within the Flutter ecosystem, utilizing the Provider pattern for dependency injection and lifecycle management. The application is structured into four discrete layers:

- **Consumer Layer (UI):** Built using Flutter's declarative widget tree. Communicates with services solely through Streams and Change Notifiers.
- **State Management Layer (Providers):** Acts as the bridge between UI and Services. The Chat Provider handles Optimistic UI updates for message sending.
- **Service Layer (Core Logic):** Each service (Discovery Service, Messaging Service, Reputation Service) is a standalone singleton responsible for algorithmic lifting such as gossip merging and background maintenance.
- **Platform Layer (Native Interop):** Interfaces with Android and iOS hardware via Method Channels. The Nearby Connections plug in provides raw P2P primitives.

The Mesh Networking Stack

AirLink's networking stack is visualized as a custom OSI model tailored for P2P:

- **Layer 1 – Physical Protocol (Nearby Connections):** Abstracts Wi-Fi Direct and Bluetooth complexity, handling the handshake and radio negotiation.
- **Layer 2 – Link Layer (Discovery Service):** Manages Scanning and Advertising cycles, endpoint authentication, and Payload Transfer Update events.
- **Layer 3 – Network Layer (Routing Isolate):** Dijkstra's algorithm is offloaded to a background Flutter Isolate to prevent UI jank. The isolate returns a Next-Hop Cache to the Messaging Service.
- **Layer 4 – Transport Layer (Messaging Service):** Handles fragmentation, reassembly, acknowledgment, and the Store-and-Forward logic.

Data Models and Persistence Strategy

AirLink uses SQLite (sqflite) for high-performance, structured data storage. The schema is optimized for rapid lookups and atomic updates:

- **Peer Table:** Stores granular metadata including is_backbone, battery_level, last_rssi, and reputation_score.

- **Message Table:** Each record includes a delivery_status enum (Pending, Sent, Delivered, Read), hop_count, and expires_at field for self-destructing messages.
- **Identity Table:** Stores public keys and registration IDs required for the Signal Protocol's E2EE.

DISCOVERY & CONNECTIVITY MANAGEMENT

Multi-Radio Strategy

AirLink utilizes the Google Nearby Connections API with the Strategy.P2P_CLUSTER configuration — a complex M-to-N topology where every device simultaneously acts as a Discovery Source and a Discovery Target [5]. This multi-radio approach employs two distinct radio tiers:

- **Tier 1 – Control Plane (BLE):** Bluetooth Low Energy acts as the always-on discovery beacon, exchanging Endpoint Discovery Packets (EDPs) containing UUID, battery status, and backbone capability.
- **Tier 2 – Data Plane (Wi-Fi Direct):** Once a handshake is completed via BLE, nodes negotiate a High Bandwidth Upgrade to Wi-Fi Direct for PTT audio and image transfers.

The Adaptive Discovery Manager (ADM)

The ADM governs the scheduling of radios to solve the Energy-Connectivity Trilemma. The duty cycle D is defined as:

$$D = T_{active} / (T_{active} + T_{sleep})$$

The ADM adjusts D based on three environmental variables:

- **Neighbour Density (ρ):** If $\rho > 10$ neighbors, D is reduced to minimize interference and battery drain.
- **Energy Reserves (E):** If battery falls below 20%, D is throttled to Survival Mode ($D \approx 0.05$).
- **Mobility Index (M):** If a node is Stationary, D is lowered; if Vehicular, D is increased to maximize the Contact Window.

Discovery Jitter and Collision Avoidance

AirLink implements Discovery Jitter, a randomized offset applied to every discovery cycle:

$$T_{interval} = T_{base} + random(-\delta, \delta)$$

Where δ is the jitter factor (typically 15% of T_base). This ensures discovery windows of neighboring nodes eventually overlap, achieving a Discovery Success Rate 40% higher than static-interval protocols in high-density environments [6].

Resilience in the Background: The Persistence Bridge

Modern mobile OSes impose aggressive Doze modes that kill background sockets. AirLink bypasses these restrictions using:

- **Foreground Service & Sticky Notifications:** Signals to the OS that mesh networking is a Vital Utility, preventing process caching.

- **Strategic Wakelocks:** Acquires PARTIAL_WAKE_LOCK only during active PTT transmissions or multi-hop relaying.
- **Job Scheduler & Work Manager:** For non-time-critical maintenance tasks, AirLink schedules periodic Maintenance Windows batched with other system tasks.

MESSAGING & ROUTING PROTOCOL

Store-and-Forward (SaF)

AirLink implements a robust Store-and-Forward (SaF) mechanism that treats every node as a temporary custodian of data, ensuring Eventual Consistency across fragmented partitions. The SaF logic is governed by the Custodial Acceptance Principle:

- **Persistent Buffering:** Messages are stored as Blob objects in SQLite, ensuring they survive application crashes or battery death.
- **Opportunistic Forwarding:** Every time Discovery Service identifies a new peer, it triggers a buffer_audit checking if the new peer provides a path to the target.
- **Congestion Control:** An LRU pruning strategy deletes oldest messages when total buffer size exceeds 100MB.

Dijkstra's Multi-Variable Path Optimization

AirLink utilizes a custom implementation of Dijkstra's Algorithm that calculates the Link Cost (C_link) using a weighted linear combination:

$$C\_link = \alpha(1/RSSI) + \beta(1 - Reputation) + \gamma(Power\ Cost)$$

Where RSSI penalizes weak links, Reputation penalizes nodes with a history of packet drops, and Power Cost heavily penalizes nodes with battery < 15% [2]. Computation is performed in a Flutter Isolate to preserve UI responsiveness.

Broadcast Storm Suppression

AirLink suppresses broadcast storms using a Sequence-Acknowledge (SeqAck) framework:

- Every broadcast is tagged with a unique (OriginUUID, SequenceID).
- Each node maintains a SeenCache (HashSet) of recently received sequence IDs.
- If the ID is present, the packet is silently dropped. If not, it is re-broadcast exactly once.
- **Time-Window Pruning:** The cache is cleared of IDs older than 1 hour to prevent memory bloat.

Self-Healing and Proactive Rerouting

If a node detects a neighbor's RSSI dropping below a critical threshold (e.g., -90 dBm), it triggers a Pre-emptive Route Audit without waiting for the link to fail. If a better path is found, the Messaging Service updates its cache before the current link snaps, providing a seamless Make-Before-Break user experience.

TRUST & REPUTATION SYSTEM

Anatomy of a Reputation Score

The Reputation Service assigns every peer a score from 0.0 to 100.0, derived from three empirical weighting vectors:

- **Handshake Success (W_hs):** A ratio of successful connection attempts to total attempts. Nodes that advertise but fail to respond (the Ghosting problem) quickly lose points.
- **Relay Fidelity (W_rf):** The most critical vector. When a node acts as a relay, the origin expects an End-to-End Acknowledgment (E2E-ACK). Dropping packets marks nodes as Black-holes or Grey-holes.
- **Uptime Stability (W_us):** Evaluates link flapping. A stable -80 dBm link for 10 minutes is more valuable than one oscillating every 30 seconds.

Gossip-Based Trust Propagation

AirLink implements Reputation Gossiping with a Bayesian-Inspired Merge formula:

$$R_{local}(C) = (1-\lambda) \cdot R_{direct}(C) + \lambda \cdot (R_{gossip}(C) \cdot R_{trust}(B) / 100)$$

Where $\lambda = 0.3$ (the Openness Factor) ensures direct experience always carries more weight than hearsay, preventing collusion-based reputation whitewashing [10].

Reputation Decay and Forgiveness Window

AirLink implements Exponential Decay so that peer reputation drifts toward a Neutral Baseline (50.0) during periods of no interaction:

$$R(t) = R_{initial} \cdot e^{(-kt)} + 50(1 - e^{(-kt)})$$

This Forgiveness mechanism prevents permanent blacklisting for transient issues, allowing nodes to rejoin the mesh and prove their reliability anew.

Sybil Attack Mitigation

AirLink counters Sybil Attacks through Cryptographic Identity Anchoring [4]:

- **Key-to-UUID Binding:** A node's UUID is a hash of its Signal Identity Public Key, breaking the Trust Chain when changed.
- **Proof-of-Relay:** Reputation boosts are only granted upon successful E2E-ACK return. Attacker nodes must actually relay successfully to gain trust.
- **OOB Verification:** Physically verified nodes (QR scan) receive a Trust Anchor Bit, with their gossip weighted 3x more heavily.

ADAPTIVE INTELLIGENCE & PEER AI

Motion-Aware Networking Strategy

AirLink treats Motion Telemetry as a primary input for networking decisions. Using the MotionService, the application monitors the device's accelerometer and gyroscope to classify mobility state:

- **Stationary:** The node is at rest. These are the most valuable nodes for routing stability.
- **Pedestrian:** Low-speed movement. Links are relatively stable but require frequent heartbeats.
- **Cyclist/Vehicular:** High-speed movement. Treated as Ephemeral Relays, used only for small, time-critical gossip packets.

By tagging every link in the Dijkstra graph with a Mobility Penalty, AirLink naturally guides long-term data streams toward stationary anchor nodes, significantly reducing the Route Break Rate in crowded urban environments [9].

Predictive Link Drop Analysis

The Peer AI Service applies Least-Squares Linear Regression over a sliding RSSI window to compute the RSSI Gradient (G):

$$G = \Sigma(t_i - \bar{t})(R_i - \bar{R}) / \Sigma(t_i - \bar{t})^2$$

The Time-to-Floor (T_floor) — estimated seconds until the signal hits the −95 dBm radio floor — is then computed as:

$$T\_floor = (-95 - R\_current) / G$$

If T_floor < 10 seconds, the AI dispatches a Pre-emptive Drop Alert to the Messaging Service, which immediately initiates Shadow Route discovery for a Seamless Handoff invisible to the user.

Swarm-Based Backbone Election

A node promotes itself to Backbone status if its local state satisfies:

$$Cost\_Backbone = f(E, M, C)$$

Where E (Energy) > 80% or charging, M (Mobility) = Stationary for > 300s, and C (Connectivity) > 3 stable neighbors. Backbone nodes serve as primary Store-and-Forward hubs, scan every 5 seconds, and synthesize gossip — reducing total gossip transactions by up to 60%.

SECURITY & CRYPTOGRAPHIC IDENTITY

Implementing the Signal Protocol

The application implements the Signal Protocol (Double Ratchet Algorithm), providing End-to-End Encryption (E2EE) for all user-level data [4]. The Signal Protocol Service manages:

- **X3DH (Extended Triple Diffie-Hellman):** The initial handshake. In AirLink, nodes Gossip their Pre-Keys to immediate neighbors. A four-part Diffie-Hellman exchange establishes a shared secret without requiring both parties to be online simultaneously.
- **The Double Ratchet:** Each message is encrypted using a unique Message Key derived from a KDF Chain that ratchets forward with every new packet, providing Forward Secrecy and Break-in Recovery.

Identity Anchoring and OOB Verification

To mitigate Man-in-the-Middle (MitM) attacks, AirLink introduces physical trust through QR Code Verification:

- Node A generates a hash of its Identity Public Key.
- Node B scans the QR and compares the hash against the key received during discovery.
- If hashes match, the Signal Protocol Service signs the peer as Verified in the local SQ Lite database.

Metadata Privacy

- **UUID Salting and Hashing:** Nodes advertise a Salted SHA-256 Hash rotating every 24 hours, preventing tracking by external observers [7].
- **Traffic Padding:** All small control packets are padded with random bytes to reach a uniform MTU, preventing size-based analysis.
- **Heartbeat Obfuscation:** Periodic Dummy Pulses are sent even during idle periods, masking the exact moment a user-initiated message is sent.

Resistance to Eclipse Attacks

AirLink's Reputation-Aware Routing naturally counters Eclipse Attacks. Because victim nodes detect that surrounding malicious nodes have low Relay Fidelity (failing to deliver messages to the wider mesh), they prioritize high-capacity backbone nodes discovered before the eclipse, effectively routing through cracks in the attacker's wall.

RESULTS & EVALUATION

Hop-Count Latency and Throughput

The table below presents median performance metrics for a standard 256-byte payload across hop distances.

Table 1: HOP-Count Latency and Throughput

Hops	L_med (ms)	L_p95 (ms)	PDR (%)	Throughput (Kbps)
1 (Direct)	145	380	99.9%	1,200
2 Hops	320	810	98.5%	450
3 Hops	680	1,550	95.2%	180
4 Hops	1,150	2,800	89.7%	75
5 Hops	1,850	4,200	81.4%	25

The data reveals a non-linear latency increase per hop, primarily induced by Store-and-Forward processing overhead. Despite latency hitting nearly 2 seconds at 5 hops, the PDR remains remarkably high (>81%). AirLink's asynchronous buffering ensures that even if links sever entirely, the message is merely delayed, never destroyed — validating the core design goal of extreme resilience over sheer speed.

B. Energy Efficiency: Validating the ADM

Table 2: Energy Efficiency Comparison

Discovery Mode	Avg Time-to-Discovery (s)	8-Hr Battery Drain (%)	Est. Standby (Days)
High-Freq (5s/5s)	3.2	28.5%	1.1
Standard (5s/30s)	16.5	11.2%	2.9
AirLink ADM	8.4 (active) / 45.0 (idle)	4.6%	7.2

The ADM achieved a battery drain profile of approximately 0.5% per hour — allowing AirLink to remain persistently active in the background for over a week, a mandatory requirement for an emergency communication application.

Broadcast Storm Suppression Efficacy

In the Dense Core virtual simulation (200 nodes), a single Global SOS Broadcast generated starkly contrasting results. Without SeqAck Suppression, the network collapsed immediately — over 1.2 million redundant packets generated within 4 seconds, causing OOM crashes on 85% of virtual nodes [3]. With AirLink SeqAck Suppression, the SOS broadcast penetrated 100% of nodes, generating only 398 total network packets, maintaining network throughput at 98% capacity.

Reputation System Integrity under Byzantine Conditions

In a 100-node simulation with 20% Byzantine Actors (100% packet drop Black-holes):

- **T=0 min:** All Byzantine nodes had reputation 50.0. Network PDR: 48%.
- **T=5 min:** Missing E2E-ACKs lowered W_rf of attackers. Network PDR improved to 65%.
- **T=15 min:** Reputation Gossiping synthesized local observations. Byzantine node reputation plummeted to < 15.0.
- **T=20 min:** Dijkstra Isolates universally rejected attacker paths. Network PDR stabilized at 96%.

This test conclusively proves that AirLink's decentralized, Bayesian-merged gossip protocols can rapidly and organically route around coordinated malicious interference without central moderation.

CHALLENGES & LIMITATIONS

Hardware Diversity and the OEM Implementation Gap

The primary engineering challenge is Android hardware fragmentation. Certain vendors implement aggressive kernel-level power-saving that fails to correctly release Bluetooth 5.0 sockets after failed discovery, causing Socket Leakage that deadlocks the radio. Additionally, Wi-Fi Direct Asymmetry between flagship and budget devices — where legacy chipsets cannot return a Handshake ACK within the 2000 ms timeout — required adaptive timeout adjustments that introduced overall mesh slowdowns.

The Physics of Density: 2.4 GHz Spectrum Congestion

The Hidden Terminal Problem arises when Nodes A and C are both in range of Node B but out of range of each other, causing simultaneous collisions. In ultra-dense configurations, the collision rate exceeded 40%. Human Body Attenuation was also a significant discovered limitation: a user holding their phone close to their chest can attenuate signal by up to 20 dB — a 100x reduction. The AI Layer addresses this using gyroscope data to adjust Dijkstra link costs based on detected posture.

Regulatory and Legal Constraints

While use of ISM bands is globally unlicensed, routing encrypted data across borders or within certain jurisdictions poses legal challenges. Some nations have strict regulations regarding E2EE where keys are not held by a telecommunications provider. AirLink strictly enforces the Signal Protocol to protect human rights in crisis zones, but the dark nature of multi-hop encrypted P2P traffic faces significant political hurdles for mainstream, app-store-distributed adoption.

CONCLUSION & FUTURE DIRECTIONS

Summary of Contributions

This research presents AirLink, a comprehensive framework proving that resilient, decentralized communication is achievable on standard consumer-grade mobile hardware without root access or custom firmware. Core contributions include:

- **Adaptive Discovery Manager (ADM):** Solving the energy-responsiveness trilemma with continuous mesh background operation under 0.5% battery drain per hour.
- **Reputation-Aware Dijkstra Routing:** A novel implementation of graph pathfinding factoring in historical reliability rather than transient RSSI values.
- **Sequence-Acknowledge (SeqAck) Protocol:** Effectively eliminating the Broadcast Storm problem in dense mobile topologies, enabling high-delivery-ratio flooding without network collapse.

Future Work: The Mesh-of-Meshes Architecture

The immediate limitation of AirLink is geographic constraint. The next evolution focuses on Long-Range (LoRa) and Low-Earth Orbit (LEO) Satellite Backhauls [8]. In a Mesh-of-Meshes architecture, local AirLink clusters would autonomously elect a Super-Relay node equipped with aftermarket LoRa hardware or a satellite gateway, bridging isolated clusters tens or thousands of kilometers apart to create a genuinely planetary-scale decentralized network [12].

Final Remarks

The centralization of digital communication has optimized for convenience at the absolute expense of resilience. As natural disasters intensify and digital censorship expands, the fragility of client-server architectures is continuously exposed. AirLink is not merely a technical proof-of-concept; it is a step toward a more democratic, anti-fragile internet — one where the network is derived intrinsically from the people who use it, and where the indispensable right to communicate can never be severed by a single point of failure.

REFERENCES

1. *Akyildiz, I. F., Wang, X., & Wang, W. (2005). "Wireless mesh networks: a survey." Computer Networks, 47(4), 445-487.*
2. *Perkins, C. E., & Royer, E. M. (1999). "Ad-hoc on-demand distance vector routing." Proceedings WMCSA '99.*
3. *Vogels, W., Van Renesse, R., & Birman, K. (2003). "The power of epidemics: robust communication for large-scale distributed systems." ACM SIGCOMM CCR, 33(1), 131-135.*
4. *Cohn-Gordon, K., et al. (2017). "A Formal Security Analysis of the Signal Messaging Protocol." Journal of Cryptology, 33(4), 1914-1983.*
5. *Google Developers (2025). Nearby Connections API Overview. https://developers.google.com/nearby/connections/overview*
6. *Karn, P. (1990). "MACA-A new channel access method for packet radio." ARRL/CRRL Computer Networking Conf., 134-140.*

7. *Zimmermann, P. R. (1995). "The Official PGP User's Guide." MIT Press.*
8. *Baggio, A. (2005). "Wireless sensor networks in precision agriculture." ACM REALWSN Workshop.*
9. *Jardosh, A., et al. (2003). "Towards realistic mobility models for mobile ad hoc networks." MobiCom '03, 217-229.*
10. *Lekkas, D. (2003). "Establishing and managing trust in peer-to-peer networks." IEEE Network, 17(5), 14-19.*
11. *Gubbi, J., et al. (2013). "Internet of Things (IoT): A vision, architectural elements, and future directions." Future Generation Computer Systems, 29(7), 1645-1660.*
12. *AirLink Open Source Repository (2026). Decentralized Mobile Mesh Framework. https://github.com/AirLink-mesh/core*
13. *Demers, A., et al. (1987). Epidemic algorithms for replicated database maintenance. PODC, 1-12.*
14. *Clausen, T., & Jacquet, P. (2003). Optimized link state routing protocol (OLSR). IETF RFC 3626.*
15. *Johnson, D. B., & Maltz, D. A. (1996). Dynamic source routing in ad hoc wireless networks. Mobile Computing, 153-181.*
16. *Bluetooth SIG (2021). Bluetooth Core Specification v5.3. https://www.bluetooth.com/specifications/specs/core-specification/*