The Root-Net Architecture: Engineering Resilient Decentralized Communication Systems for Humanitarian Crisis Response

The Crisis of Connectivity: Infrastructure Fragility in Extreme Environments

The modern telecommunications landscape is characterized by a high degree of centralization, which creates inherent vulnerabilities during large-scale natural disasters. Traditional cellular networks, classified as Ring 3 infrastructure, operate on a hierarchical model where mobile devices act as passive endpoints dependent on a functional Base Transceiver Station (BTS) and a stable backhaul connection.[1] When catastrophic events such as the 7.8 magnitude earthquake in southern Türkiye and northern Syria occur, as witnessed in February 2023, this entire hierarchy is susceptible to immediate collapse.[3] The physical destruction of towers, combined with the severing of fiber-optic backhauls and the widespread loss of electricity, renders billions of dollars of infrastructure useless at the exact moment it is most required.[2]

Statistics from the 2023 Kahramanmaraş earthquakes provide a sobering quantitative baseline for the failure of centralized systems. Over 53,537 deaths were confirmed in Türkiye alone, with nearly 14 million people affected across 11 provinces.[3] A critical factor in the high casualty rate was the "digital darkness" that descended upon the region. In the Hatay province, communication was effectively severed for the first three days—the "Golden Window" during which the probability of rescue is highest.[3] Infrastructure reports indicated that the loss of power made it impossible to refuel the generators powering surviving radio towers, while the massive surge in local traffic caused a "digital tragedy of the commons," where the few functional nodes were saturated beyond capacity.[1]

The Root-Net framework is proposed as a direct architectural response to this fragility. By shifting the communication paradigm from a star topology to a decentralized peer-to-peer (P2P) mesh, Root-Net transforms every individual smartphone into a "resilient root" capable of carrying, storing, and forwarding packets.[8] This architecture does not seek to replace the cellular grid but to provide a robust, ad-hoc fallback layer that functions at the packet level, utilizing the hardware already present in the hands of survivors and responders.[11]

The Impact of Network Failure on Survival Metrics

The availability of communication is not merely a logistical convenience; it is a primary determinant of survival probability. Disaster medicine recognizes the "Golden Period"—the first 72 hours following an incident—as the timeframe during which human physiology can endure without external resources if extracted or stabilized.[6]

Time Post-Disaster	**Survival Rate**	**Operational Reality**
First 24 Hours	90%	Local self-organized rescue is primary.
24 - 48 Hours	50% - 60%	Aid distribution and triage begin.
48 - 72 Hours	20% - 30%	Critical window for survival without water/food.
After 72 Hours	< 10%	Recovery replaces rescue as the priority.
[13]		

In the absence of functional communication networks, the coordination of rescue efforts becomes fragmented. During the 2023 Maui fires and the subsequent 2025 Istanbul coastal earthquake, survivors remained without outside assistance for over 48 hours, relying solely on local coordination that was hampered by the lack of a common digital operating picture.[15] Root-Net addresses this by enabling a localized mesh that allows for the dissemination of high-priority SOS messages and the sharing of GPS coordinates even when every cellular tower in a 50-mile radius is offline.[8]

Root-Net Conceptual Framework: Packet-Level Resilience

The core innovation of Root-Net lies in its implementation of networking at the packet level, rather than relying on session-based connectivity. It utilizes a Delay-Tolerant Networking (DTN) architecture, which assumes that a continuous end-to-end path from the source to the destination may never exist.[9] Instead of the standard TCP/IP "connect-then-send" model, Root-Net uses a "Store-Carry-Forward" paradigm.[9]

Hardware Transport Layer Analysis

Root-Net leverages the dual-transport capabilities of modern Android and iOS devices:

Bluetooth Low Energy (BLE) and Wi-Fi Direct (P2P). These technologies operate on different physical layer principles, offering a trade-off between range, energy efficiency, and bandwidth.[21]

Feature	Bluetooth Low Energy (BLE)	Wi-Fi Direct (P2P)
Max Range	100 meters (approx. 330 ft)	Up to 200 meters
Bandwidth	Low (approx. 1-2 Mbps)	High (up to 250 Mbps)
Power Impact	Minimal; suitable for background	Significant; requires energy management
Security	AES-128 (standard)	WPA2/WPA3 (standard)
Primary Use	Peer discovery and tiny packet sync	Large file/media/voice transfer
21		

The technical challenge for Root-Net is the seamless orchestration of these two transports. Bluetooth is used as the "persistent heartbeat" of the mesh, constantly scanning for nearby peers and exchanging routing tables or small text messages.[11] When a larger payload is detected (such as a photo of a collapsed building or a voice note), the system triggers a Wi-Fi Direct "Group Owner" negotiation, momentarily increasing power consumption to achieve high-speed transfer before returning to a low-power state.[11]

Routing and Graph Theoretical Foundations

In a mesh network, every device is a node, and every connection is an edge. Root-Net treats the entire disaster zone as a dynamic graph $G = (V, E)$, where the set of vertices V consists of mobile nodes and the set of edges E consists of transient peer-to-peer links.[27] Because nodes are mobile, the graph is constantly evolving, requiring graph algorithms that prioritize pathfinding over path maintenance.[9]

Root-Net implements a hybrid routing strategy that combines flooding-based and prediction-based categories:

1. **Epidemic Routing**: This protocol aggressively replicates packets to every newly encountered node that does not possess a copy.[18] It serves as the upper bound on delivery performance but is extremely resource-hungry.[30]
2. **PRoPHET (Probabilistic Routing Protocol using History of Encounters and Transitivity)**: This protocol uses a delivery predictability metric $P(a, b) \in $ to decide whether to forward a packet. If node A frequently encounters node B, and node B frequently encounters node C, node A assumes it has a non-negligible chance of delivering a message to C via B.[18]

The update rule for PRoPHET is governed by three primary equations:

- **Encounters**: Whenever two nodes meet, their predictability increases:

$$P(a,b)_{new} = P(a,b)_{old} + (1 - P(a,b)_{old}) \times P_{init}$$

- **Aging**: If nodes do not meet for a duration k, the predictability decays:

$$P(a,b)_{new} = P(a,b)_{old} \times \gamma^k$$

- **Transitivity**: Nodes exchange predictability vectors to estimate multi-hop paths:

$$P(a,c)_{new} = P(a,c)_{old} + (1 - P(a,c)_{old}) \times P(a,b) \times P(b,c) \times \beta$$

[31]

By utilizing these algorithms, Root-Net ensures that an SOS message "hops" from a trapped victim's phone (zero connectivity) to a passing volunteer's phone, and eventually to a device with a satellite link or a surviving cellular connection (Ring 3), thereby closing the communication loop.[8]

Security and Cryptography: Hardening the Resilient Root

In the context of civil unrest or large-scale humanitarian crises, security is not a secondary feature but a prerequisite for trust. The failure of existing mesh applications like Bridgefy highlights the catastrophic consequences of improper security implementation.[33]

Vulnerability Landscape: Lessons from Bridgefy

Academic security analyses of Bridgefy between 2020 and 2024 revealed that simply adopting industry-standard protocols like the Signal Protocol (libsignal) is insufficient if the integration is flawed.[33]

Vulnerability Type	Mechanism	Potential Impact
TOCTOU Attack	Side-stepping Signal guarantees by exploiting time-of-check issues.	Total loss of message confidentiality.
User Tracking	Exposure of Bluetooth MAC addresses/identifiers.	Production of social graphs by state actors.
Zip Bombs	Sending maliciously compressed packets (Decompression Bombs).	Network-wide denial of service (DoS).
Impersonation	Lack of authentication in broadcast channels.	Spread of misinformation/fake rescue alerts.
33		

Root-Net mitigates these risks through a "Privacy-First" decentralized architecture. Unlike centralized apps that require a one-time internet login (which creates a metadata trail), Root-Net employs local QR code-based handshakes for contact verification, similar to the Briar project.[22] This ensures that man-in-the-middle (MitM) attacks are physically impossible during the initial trust exchange.[22]

Cryptographic Hardening Strategies

Root-Net implements several layers of encryption to protect the user's "resilient root":

- **End-to-End Encryption (E2EE)**: Utilizing the Double Ratchet Algorithm (Signal Protocol) for private 1:1 messages, ensuring forward secrecy and self-healing properties.[24]
- **Noise Protocol Framework**: For secure handshakes and transport encryption across BLE and Wi-Fi Direct links.[38]
- **Anonymity and Pseudonymity**: No phone numbers or real-world identifiers are used. Users are identified by transient cryptographic public keys that can be rotated periodically to prevent tracking.[29]
- **Resource Validation**: To prevent "zip bomb" attacks, Root-Net implements a rate-limiting Energy Manager and strict packet size validation at the native layer before processing

payloads in the Dart VM.[26]

UX/UI Design for Extreme Stress: The Front-End Challenge

Designing for disaster response is fundamentally different from standard mobile app design. In high-stress scenarios, cognitive function is impaired, working memory shrinks, and the "Golden Period" demands instantaneous action.[41]

The IDEA Model of Crisis Communication

Root-Net adheres to the IDEA model (Internalization, Distribution, Explanation, Action) to ensure instructional risk communication is effective.[44]

- **Internalization**: The UI uses vivid, high-contrast color-coding (vivid red for live threats, yellow for caution) to indicate urgency without requiring the user to read long blocks of text.[41]
- **Distribution**: Ensuring the message reaches the right audience via multi-channel mesh propagation.[41]
- **Explanation**: Providing clear, jargon-free context (e.g., "Fire is approaching" instead of "Consider evacuating").[42]
- **Action**: Dominated by a single, prominent primary button (e.g., "Broadcast SOS") to minimize friction.[42]

Technical Roadmap: The "One Night" Front-End MVP

To achieve a functional front-end in a single night for a presentation, the focus must shift from native protocol debugging to high-fidelity simulation and core visualization. Native P2P permissions on Android (API 33+) are complex and time-consuming to configure.[21]

Night 1 Strategy: Simulation over Protocol Implementation

1. **State Management (BLoC/Riverpod)**: Build a robust local state that handles a list of "Virtual Peers." Use a Timer to simulate the discovery of nodes to demonstrate the mesh growth.[46]
2. **Mesh Visualization (CustomPainter)**: Instead of a standard list view, use Flutter's CustomPainter to draw a real-time graph of the mesh. Use animated lines to show "Message Hopping" between nodes A, B, and C. This provides a "wow factor" in a PowerPoint presentation that a simple chat screen lacks.[48]
3. **UI Kit Integration**: Drop in production-ready chat components from kits like CometChat or Stream to handle the messaging interface quickly, allowing development time to be spent on the unique mesh-specific features.[50]
4. **Mocked Data Injection**: Create a "Scenario Mode" that injects real data points from the 2023 Turkey earthquake (e.g., specific Hatay collapse locations) to demonstrate how the

mesh would provide localized maps offline.[4]

Feature	presentation MVP (Night 1)	production Implementation
Networking	Mocked NearbyService streams	Full Wi-Fi Direct/BLE protocol stack
Routing	Simulated hop logic via Timer	PRoPHET/Epidemic C++ implementation
Discovery	Auto-generating "fake" peers	Native WifiP2pManager discovery
Map	Static image with mesh overlay	Offline downloaded Map Tiles (SQLite)
[17]		

Infrastructure and Social Impact: Scaling the Mesh

The true value of Root-Net is realized when it achieves critical mass. Like a "game of digital telephone," the mesh network grows exponentially stronger with each additional user.[8]

Multi-Hazard Resilience and Government Integration

While Root-Net is a grassroots tool, its integration into formal disaster management frameworks is vital for long-term viability. The Kahramanmaraş earthquakes demonstrated that government agencies (AFAD) and NGOs (AHBAP) need standardized protocols to bridge the gap between their specialized systems and the public's mobile devices.[3]

Root-Net provides a "Unified Communication Layer":

- **Responders**: Use ruggedized radios (e.g., Hytera DMR) as long-range gateways.[12]
- **Civilians**: Use Root-Net on standard smartphones for local neighborhood coordination.[6]
- **Infrastructure**: Surviving terrestrial nodes (IXPs) act as high-bandwidth bridges for cross-regional data.[54]

Economic and Humanitarian Outcomes

From 1990 to 2018, natural disasters affected over 5.89 billion people, costing US $2.95 trillion.[55]

Studies on the application of Information and Communication Technologies (ICTs) in relief efforts suggest that while many technologies are deployed, very few focus on the "participatory fairness" of survivors.[6] Root-Net's "SOS Hub" protocol, which rotates the burden of acting as a network relay based on battery charge, ensures that even those with 5% battery can remain connected to the life-saving mesh for the duration of the 72-hour golden period.[6]

Final Synthesis: The Root-Net Thesis

The Root-Net project represents a paradigm shift from vulnerable, centralized connectivity to a "resilient root" of peer-driven communication. By leveraging the packet-level capabilities of Flutter and the underlying physics of Wi-Fi Direct and BLE, Root-Net addresses the fundamental failure of modern telecommunications during natural disasters. The architecture is built on the dual pillars of graph-theoretical routing efficiency (PRoPHET) and rigorous cryptographic security (Noise/libsignal).

For the developer, the path forward involves a rapid transition from simulation to field-tested protocol. While a functional front-end can be built in a single night to demonstrate the concept to stakeholders, the long-term goal is the creation of a delay-tolerant infrastructure that treats every mobile device not just as a tool for consumption, but as a vital node in a global, off-grid safety net. In the rubble of cities like Hatay or the flooded streets of the future, Root-Net ensures that when the towers fall, the people remain connected, organized, and visible to those who can save them.[3]

Works cited

1. Impact of Communication Failures in Crisis & Solutions | BCC - Base Camp Connect, accessed on February 26, 2026, https://basecampconnect.com/bcc-the-impact-of-communication-failures-in-crisis-situations-solutions/
2. Emergency Telecommunications Failure (TL0211) - UNDRR, accessed on February 26, 2026, https://www.undrr.org/understanding-disaster-risk/terminology/hips/tl0211
3. Communication and Coordination Networks in the 2023 Kahramanmaras Earthquakes, accessed on February 26, 2026, https://hazards.colorado.edu/quick-response-report/communication-and-coordination-networks-in-the-2023-kahramanmaras-earthquakes
4. 2023 Turkey–Syria earthquakes - Wikipedia, accessed on February 26, 2026, https://en.wikipedia.org/wiki/2023_Turkey%E2%80%93Syria_earthquakes
5. Disaster Communication | Loma Linda University, accessed on February 26, 2026, https://llu.edu/campus-spiritual-life/emergency/disaster-communication
6. Saving lives with a fair emergency communication system, accessed on February 26, 2026, https://www.tudelft.nl/en/stories/articles/saving-lives-with-a-fair-emergency-communication-system

7. Crisis Communication Suffers During Natural Disasters - Public Policy Institute of California, accessed on February 26, 2026, https://www.ppic.org/blog/crisis-communication-suffers-during-natural-disasters/
8. Bridgefy: The Offline Messaging App Revolutionizing Crisis Communication Worldwide, accessed on February 26, 2026, https://www.alchemistaccelerator.com/blog/bridgefy-the-offline-messaging-app-revolutionizing-crisis-communication-worldwide
9. Delay-tolerant networking - Wikipedia, accessed on February 26, 2026, https://en.wikipedia.org/wiki/Delay-tolerant_networking
10. Routing strategies in Delay Tolerant Networks: A Survey - SciSpace, accessed on February 26, 2026, https://scispace.com/pdf/routing-strategies-in-delay-tolerant-networks-a-survey-1vpl5h00sz.pdf
11. Decentralized P2P Messaging with Bluetooth Authentication and ..., accessed on February 26, 2026, https://substack.com/home/post/p-173414962
12. Emergency Communications Vital to Natural Disaster Preparedness - Hytera Blogs, accessed on February 26, 2026, https://www.hytera.com/en/connect/blog/emergency-communications-vital-to-natural-disaster-preparedness
13. Statistics of Rescue Time and Survival Rate [5] - ResearchGate, accessed on February 26, 2026, https://www.researchgate.net/figure/Statistics-of-Rescue-Time-and-Survival-Rate-5_fig2_259657048
14. Statistics of Rescue Time and Survival Rate [5] | Download Scientific Diagram - ResearchGate, accessed on February 26, 2026, https://www.researchgate.net/figure/Statistics-of-Rescue-Time-and-Survival-Rate-5_fig2_259674878
15. Why the First 72 Hours Decide Survival – Stay Ready Outdoors, accessed on February 26, 2026, https://stayreadyoutdoors.com/blogs/guides-and-education/average-rescue-aid-timelines-in-survival-scenarios
16. Mobile networks collapse after İstanbul earthquake, echoing failures seen in 2023 disaster, accessed on February 26, 2026, https://turkishminute.com/2025/04/23/mobile-networks-collapse-after-istanbul-earthquake-echoing-failures-seen-in-2023-disaster/
17. Open-source Flutter client for MeshCore LoRa mesh networking devices - GitHub, accessed on February 26, 2026, https://github.com/zjs81/meshcore-open
18. Delay Tolerant Network Routing as a Machine Learning Classification Problem, accessed on February 26, 2026, https://ntrs.nasa.gov/api/citations/20190000830/downloads/20190000830.pdf
19. Enhanced Message Replication Technique for DTN Routing Protocols - Semantic Scholar, accessed on February 26, 2026, https://pdfs.semanticscholar.org/fd49/f5e08d6557f11c8865e407e6b68e5239ebbc.pdf

20. Evaluation of DTN Routing Protocols in Post Disaster Scenario - SciSpace, accessed on February 26, 2026, https://scispace.com/pdf/evaluation-of-dtn-routing-protocols-in-post-disaster-4kfhz7r8tv.pdf
21. Create P2P connections with Wi-Fi Direct - Android Developers, accessed on February 26, 2026, https://developer.android.com/develop/connectivity/wifi/wifi-direct
22. Briar Offline Mesh: When Internet Shutdowns Cut 85M Off | byteiota, accessed on February 26, 2026, https://byteiota.com/briar-offline-mesh-when-internet-shutdowns-cut-85m-off/
23. ugo-studio/flutter_p2p_connection: A plugin for creating wifi direct p2p connections. - GitHub, accessed on February 26, 2026, https://github.com/ugo-studio/flutter_p2p_connection
24. Bridgefy – Mesh Network SDK for Mobile Apps, accessed on February 26, 2026, https://bridgefy.me/
25. wifi_direct_plugin | Flutter package - Pub.dev, accessed on February 26, 2026, https://pub.dev/packages/wifi_direct_plugin
26. Speew: Building a 100% Offline P2P Mesh Network with Flutter (Using Wi-Fi Direct/BT Mesh for Zero-Internet Comm) Hey Flutter devs! - Reddit, accessed on February 26, 2026, https://www.reddit.com/r/FlutterDev/comments/1pnw1ru/speew_building_a_100_offline_p2p_mesh_network/
27. Bridgefy: The Most Popular Offline Mesh Messaging App, accessed on February 26, 2026, https://dscnextconference.com/bridgefy-most-popular-offline-mesh-messaging-app/
28. Delay Tolerant Network and the Algorithms of DTN Routing - ResearchGate, accessed on February 26, 2026, https://www.researchgate.net/publication/331748089_Delay_Tolerant_Network_and_the_Algorithms_of_DTN_Routing
29. AOSSIE-Org/OpenPeerChat-flutter - GitHub, accessed on February 26, 2026, https://github.com/AOSSIE-Org/OpenPeerChat-flutter
30. Delay-Tolerant Networking for Tsunami Evacuation on the Small Island of Hachijojima - arXiv, accessed on February 26, 2026, https://www.arxiv.org/pdf/2601.00109
31. Routing in delay-tolerant networking - Wikipedia, accessed on February 26, 2026, https://en.wikipedia.org/wiki/Routing_in_delay-tolerant_networking
32. (PDF) Performance Comparison of Epidemic, PRoPHET, Spray and Wait, Binary Spray and Wait, and PRoPHETv2 - Academia.edu, accessed on February 26, 2026, https://www.academia.edu/9001708/Performance_Comparison_of_Epidemic_PRoPHET_Spray_and_Wait_Binary_Spray_and_Wait_and_PRoPHETv2
33. Breaking Bridgefy, again: Adopting libsignal is not enough - USENIX, accessed on February 26, 2026, https://www.usenix.org/conference/usenixsecurity22/presentation/albrecht
34. Breaking Bridgefy, again, accessed on February 26, 2026,

https://eikendev.github.io/breaking-bridgefy-again/
35. Breaking Bridgefy, again - ETH Zürich, accessed on February 26, 2026, https://ethz.ch/content/dam/ethz/special-interest/infk/inst-infsec/appliedcrypto/education/theses/breaking-bridgefy-again.pdf
36. Mesh Messaging in Large-scale Protests: Breaking Bridgefy (Abridged version), accessed on February 26, 2026, https://martinralbrecht.wordpress.com/wp-content/uploads/2020/08/bridgefy-abridged.pdf
37. Survey of Mesh Networking Messengers - TUM, accessed on February 26, 2026, https://www.net.in.tum.de/fileadmin/TUM/NET/NET-2021-05-1/NET-2021-05-1_01.pdf
38. permissionlesstech/bitchat: bluetooth mesh chat, IRC vibes - GitHub, accessed on February 26, 2026, https://github.com/permissionlesstech/bitchat
39. chatapp-flutter · GitHub Topics, accessed on February 26, 2026, https://github.com/topics/chatapp-flutter
40. How it works - Briar, accessed on February 26, 2026, https://briarproject.org/how-it-works/
41. Top UX design tips for disaster management applications - Intersec, accessed on February 26, 2026, https://intersec.com/blog/top-ux-design-tips-for-disaster-management-applications
42. UX Design for Crisis Situations: Lessons from the Los Angeles Wildfires - UXmatters, accessed on February 26, 2026, https://www.uxmatters.com/mt/archives/2025/03/ux-design-for-crisis-situations-lessons-from-the-los-angeles-wildfires.php
43. Designing for Urgency: What 911 Emergency Apps Reveal About Fast UX - Medium, accessed on February 26, 2026, https://medium.com/@blessingokpala/designing-for-urgency-what-911-emergency-apps-reveal-about-fast-ux-e698f8f9d726
44. The crisis response to the Türkiye Syria earthquakes in 2023 - SSRN, accessed on February 26, 2026, https://papers.ssrn.com/sol3/Delivery.cfm/68764937-1f12-4a08-8d3e-9f91f3120b3c-MECA.pdf?abstractid=5248556&mirid=1
45. nearby_connections | Flutter package - Pub.dev, accessed on February 26, 2026, https://pub.dev/packages/nearby_connections
46. Advanced Flutter UI: How to Build a Chat App with Custom Message Bubbles - Maxim Gorin, accessed on February 26, 2026, https://maxim-gorin.medium.com/advanced-flutter-ui-how-to-build-a-chat-app-with-custom-message-bubbles-4f90282b8be0
47. What's the best state management library for flutter in 2024? | Manchester Digital, accessed on February 26, 2026, https://www.manchesterdigital.com/post/foresight-mobile/whats-the-best-state-management-library-for-flutter-in-2024
48. Flutter Canvas: From Basics to Interactive Drawing and Animated Equalizer - Maxim Gorin, accessed on February 26, 2026,

https://maxim-gorin.medium.com/flutter-canvas-from-basics-to-interactive-drawing-and-animated-equalizer-106b3716487c
49. Animating Canvas in Flutter | Darshan's Blog, accessed on February 26, 2026, https://blog.darshanrander.com/posts/tech/animating-canvas-in-flutter/
50. Add In-App Chat to Flutter Apps with CometChat UI Kit & SDK, accessed on February 26, 2026, https://www.cometchat.com/build/expo-chat
51. Add In-App Chat to Flutter Apps with CometChat UI Kit & SDK, accessed on February 26, 2026, https://www.cometchat.com/build/flutter-chat
52. Theming - Flutter (v5) Chat Messaging Docs - GetStream.io, accessed on February 26, 2026, https://getstream.io/chat/docs/sdk/flutter/v5/customization/stream_chat_and_theming/
53. Türkiye & Syria Earthquakes 2023 | NASA Applied Sciences, accessed on February 26, 2026, https://appliedsciences.nasa.gov/what-we-do/disasters/disasters-activations/turkiye-syria-earthquakes-2023
54. A Data-Driven Approach to Disaster Resilience in Communication Networks - TU Delft Research Portal, accessed on February 26, 2026, https://research.tudelft.nl/en/publications/a-data-driven-approach-to-disaster-resilience-in-communication-ne/
55. Real World Effectiveness of Information and Communication Technologies in Disaster Relief: A Systematic Review - PMC, accessed on February 26, 2026, https://pmc.ncbi.nlm.nih.gov/articles/PMC7719660/