

ResQNet: A Crowdsourced Disaster Preparedness and Real-Time SOS Mesh Network Using Wi-Fi Direct MANET, Delay Tolerant Networking, and Artificial Intelligence

Hitha Sundar P S
Department of Information
Science and Engineering
AMC Engineering College,
Bengaluru, Karnataka, India
Visvesvaraya
Technological University

Huma Anjum
Department of Information
Science and Engineering
AMC Engineering College,
Bengaluru, Karnataka, India
Visvesvaraya Technological
University

Karan Vivek
Department of Information
Science and Engineering
AMC Engineering College,
Bengaluru, Karnataka, India
Visvesvaraya Technological
University

Roshan Kumar B K
Department of Information
Science and Engineering
AMC Engineering College,
Bengaluru, Karnataka, India
Visvesvaraya Technological
University

Divyashree
Department of Information
Science and Engineering
AMC Engineering College,
Bengaluru, Karnataka, India
Visvesvaraya Technological
University

Abstract

Natural disasters such as earthquakes, floods, cyclones, landslides, and wildfires frequently damage cellular towers and internet infrastructure, leaving victims unable to contact rescue teams and depriving responders of real-time situational information. This paper presents ResQNet, a crowdsourced, infrastructure-independent disaster communication system that uses Wi-Fi Direct-based Mobile Ad Hoc Networking (MANET) and Delay Tolerant Networking (DTN) to relay SOS alerts between nearby smartphones without relying on cellular or internet connectivity. Each smartphone in the network functions simultaneously as a sender, relay, and store-and-forward node: SOS packets, tagged with a unique identifier, timestamp, GPS coordinates, hop count, and time-to-live (TTL), are propagated hop by hop until they reach an internet-enabled gateway device or a rescue authority. Duplicate packet suppression and TTL-controlled flooding are used to keep the network efficient under heavy load. ResQNet further incorporates artificial intelligence for emergency severity classification, location-based clustering of victims, accessibility analysis, and hotspot identification, together with voice-triggered SOS activation for hands-free operation. A web-based dashboard visualizes live SOS alerts, victim clusters, and priority levels on a map to support rescue coordination. Simulation-based evaluation of packet delivery ratio, latency, and message loss indicates that the proposed architecture can sustain reliable emergency communication even when conventional networks fail, offering a scalable and low-cost foundation for future disaster-response systems.

Keywords—*Disaster Communication, Mobile Ad Hoc Network (MANET), Wi-Fi Direct, Delay Tolerant Networking (DTN), Mesh Networking, SOS System, Emergency Response, Artificial Intelligence, Geospatial Analysis.*

I. INTRODUCTION

Natural disasters such as earthquakes, floods, cyclones, landslides, and wildfires cause severe damage to human life, infrastructure, and communication systems. During such events, mobile networks and internet services frequently fail because communication towers are overloaded or physically destroyed, making it extremely difficult for victims to contact rescue teams. This loss of connectivity delays emergency response and increases the risk to affected populations. Most existing emergency communication systems depend heavily on centralized infrastructure, which becomes unreliable precisely when uninterrupted communication is most needed.

To address this problem, this paper proposes ResQNet, a decentralized emergency communication framework built on Mobile Ad Hoc Network (MANET) concepts and Wi-Fi Direct communication. In ResQNet, smartphones act as independent communication nodes that can generate, receive, store, and forward SOS alerts without relying on internet connectivity. When a victim raises an emergency alert, the resulting SOS packet is transferred across nearby devices through multi-hop mesh communication until it reaches a rescue authority or an internet-enabled gateway node. Delay Tolerant Networking (DTN) is used to temporarily store packets and forward them automatically once connectivity becomes available.

The framework further enhances disaster response through intelligent, location-aware features. GPS coordinates are attached to every SOS packet for accurate victim tracking, voice-triggered SOS activation allows victims to request help without manually operating the device, and live map-based visualization helps rescue teams identify affected regions. Together, these features improve situational awareness and support faster, more coordinated rescue operations while reducing dependency on traditional network infrastructure.

II. RELATED WORK

A review of recent research on disaster communication reveals a shift from purely infrastructure-dependent systems toward decentralized, AI-assisted, and delay-tolerant architectures. Savale et al. proposed a disaster management ecosystem that combines crowdsourced data, DBSCAN clustering, BLIP-based image captioning, and NLP-based categorization within a cloud architecture to improve situational awareness, though the approach assumes continuous cloud connectivity [1]. Okuda et al. designed a wireless mesh network for disaster notification using Low Power Wide Area (LPWA) technology together with the Plumtree routing algorithm, achieving low latency and efficient dissemination under node failure, but without an accompanying mobile SOS application layer [2].

Gupta et al. presented an AI-enabled Delay Tolerant Networking framework that applies store-and-forward routing and intelligent packet prioritization for emergency communication, improving reliability and reducing packet loss, though the work stops short of a full mobile deployment [3]. Shah et al. proposed a store-carry-forward DTN routing protocol for opportunistic networks that improves delivery in disconnected, highly dynamic environments [4], while Estivil and Radenkovic evaluated opportunistic Wi-Fi and Bluetooth-based DTN routing protocols for underground and disaster emergency-alert dissemination, analyzing delivery efficiency, latency, and routing overhead [5].

Carreras-Coch et al. surveyed communication technologies for emergency situations, emphasizing mesh networking, peer-to-peer communication, and fault-tolerant architectures for rapid deployment after infrastructure failure [6]. Pokhrel and Choi examined decentralized mesh networking for post-disaster communication using peer-to-peer forwarding and multi-hop routing, highlighting improved scalability and reduced dependency on cellular infrastructure [7]. Ashraf et al. introduced WiMesh, a low-cost Wi-Fi mesh solution for disaster communication in regions with limited infrastructure, demonstrating effective multi-hop communication with low power consumption [8].

Collectively, this literature demonstrates the value of MANET, DTN, wireless mesh networking, and AI-based analysis for disaster response, but also reveals recurring gaps: most systems address either offline communication or intelligent prioritization in isolation, few integrate voice-triggered activation or AI-based geospatial risk analysis, and mobile, end-to-end deployments that combine store-and-forward mesh communication with a rescue-facing real-time dashboard remain limited. These gaps motivate the design of ResQNet, which unifies Wi-Fi Direct MANET communication, DTN-based store-and-forward routing, AI-driven emergency

prioritization, and real-time rescue-dashboard monitoring within a single deployable system.

III. PROPOSED METHODOLOGY

ResQNet follows a seven-stage methodology spanning SOS generation, peer discovery, multi-hop relay, delay-tolerant storage, duplicate/priority handling, AI-based analysis, and cloud synchronization.

A. SOS Generation and Emergency Trigger

The victim triggers an emergency alert either manually through a large, easily accessible "SEND SOS" button or automatically through voice- and sensor-based distress detection, which listens for distress keywords and monitors accelerometer/gyroscope data for falls or sudden impacts. On activation, the FusedLocationProvider service fetches the current GPS coordinates, which are combined with a unique message ID, timestamp, hop count, and time-to-live (TTL) value to form an SOS packet.

B. Local Storage and Peer Discovery

The generated SOS packet is immediately persisted in a local Room database so that it is never lost, even if no peers are currently reachable. A background service continuously scans for nearby devices using Wi-Fi Direct peer discovery; if no peers are found, the service waits and retries periodically, which is the core behavior that enables Delay Tolerant Networking under sparse connectivity.

C. Multi-Hop Mesh Relay

Once peers are discovered, unsent packets are retrieved from local storage and relayed via Wi-Fi Direct to nearby devices. Each receiving device stores the packet, displays the SOS alert locally, and forwards it to the next peer, allowing the message to propagate across a chain of relay nodes. Hop-count increments and TTL decrements at every relay bound the propagation radius and prevent uncontrolled flooding of the network.

D. Duplicate Suppression and Priority Handling

Every relay node checks incoming packets against previously seen message IDs to discard duplicates before re-forwarding, which prevents redundant transmissions and reduces network congestion. Packets are also tagged with a priority level so that life-threatening alerts are relayed ahead of routine updates whenever a node must choose which packets to forward first.

E. Gateway Detection and Cloud Synchronization

Whenever a node in the mesh detects an active internet connection, it acts as a gateway: pending SOS packets are uploaded to a FastAPI-based backend and stored centrally, while nodes without connectivity continue to propagate the

packet through the mesh only. This hybrid design allows the system to degrade gracefully from full internet connectivity down to a purely offline mesh, and to resynchronize automatically as soon as any device regains connectivity.

F. AI-Based Emergency Prioritization and Geospatial Analysis

On the backend, AI models classify the severity of each incoming emergency, using message content, location context, and historical patterns to assign a priority level. Location-based clustering groups nearby victims to identify high-priority zones, while accessibility analysis considers nearby roads, water bodies, and building density to estimate how easily rescue teams can reach a given location. Where available, satellite imagery and environmental data enrich this analysis with disaster-context information such as flood extent or fire spread.

G. Rescue Dashboard and Monitoring

A web-based dashboard displays live SOS alerts, priority levels, victim clusters, and packet-delivery status on an interactive map, using Leaflet.js for visualization. Rescue authorities can filter and search active alerts, view estimated rescue routes, and update the status of each alert, closing the loop between the field-level mesh network and the command-center decision-making process.

IV. SYSTEM ARCHITECTURE

The ResQNet architecture is organized into four layers: an offline Mesh Network Layer, a Gateway and Cloud Backend, a Rescue Dashboard Layer, and a supporting Communication Protocol / Packet Structure layer that ties them together.

In the Mesh Network Layer, the victim's device generates the SOS packet and relays it via Wi-Fi Direct to peer devices, each of which applies store-and-forward logic, TTL control, hop-count updates, duplicate-packet suppression, and priority handling before relaying the packet onward. When a relay node with internet access — the Gateway Device — is reached, it validates and de-duplicates incoming packets and uploads them to a Cloud Backend, where FastAPI-based backend processing stores the data and triggers an AI/ML engine that classifies emergency severity, prioritizes alerts, and recommends actions. The Rescue Dashboard Layer, intended for rescue teams and authorities, consumes this processed data to present a live SOS alert feed, map visualization with priority and severity indicators, alert-detail and status views, advanced filters and search, and exportable reports.

The system supports two primary actors: the Victim/User, who can send SOS alerts, share live location, send emergency updates, and receive acknowledgements; and the Rescue Team/Authority, who can view SOS alerts on the dashboard, filter and prioritize alerts using AI-based classification, view

victim locations on the map, and dispatch rescue teams. Nearby devices additionally act as Relay Nodes, receiving, storing, and forwarding SOS messages under DTN rules regardless of whether their own user is directly involved in the emergency.

Each SOS packet follows a lightweight JSON structure containing a message ID, source/device ID, timestamp, latitude and longitude, a free-text message, a priority level, a hop count, and a TTL value. This compact structure keeps transmission overhead low, which is important given the constrained bandwidth of Wi-Fi Direct links formed opportunistically between battery-limited smartphones during a disaster.

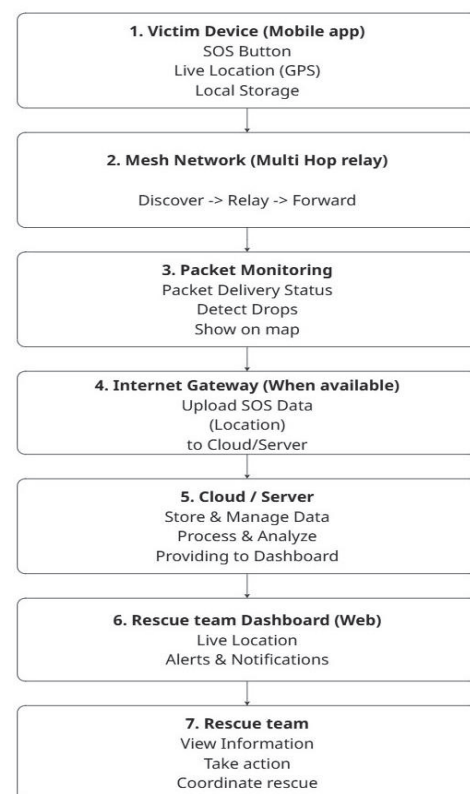


Figure 1: Simplified ResQNet system architecture.

V. IMPLEMENTATION

The ResQNet prototype targets Android smartphones for the field-facing mesh application and a Python-based backend for cloud processing and analytics. The mobile application is developed in Kotlin and is responsible for SOS packet creation, GPS acquisition through Google Location Services, Wi-Fi Direct peer discovery and relay, and local persistence of pending and sent packets using a Room database. The backend is implemented using FastAPI and exposes REST endpoints for SOS ingestion, voice-triggered emergency requests, disaster-

risk prediction, rescue-team detection, and safe-zone recommendation; SQLite is used for structured storage of victim information, SOS alerts, and emergency logs.

AI components are built around PyTorch, OpenCV, and NumPy: convolutional neural network models are intended to classify disaster severity from user-submitted images (floods, fires, earthquake damage, and infrastructure damage), while lightweight classification and clustering models drive emergency-priority scoring and victim-hotspot identification. Development uses Visual Studio Code and Android Studio, with Postman for API testing and GitHub for version control.

Table 1: Technologies used in the ResQNet prototype

Component	Technology
Mobile Application	Kotlin, Android SDK
Location Services	Google FusedLocationProvider (GPS)
Offline Mesh Communication	Wi-Fi Direct, BLE (MANET)
Local Storage	Room Database (SQLite)
Backend API	Python, FastAPI
Server Database	SQLite
AI / ML Libraries	PyTorch, OpenCV, NumPy
Dashboard	Web dashboard with Leaflet.js maps
API Testing	Postman, Swagger UI
Version Control	GitHub

Testing and validation activities to date have covered unit-level checks of SOS-button behavior, GPS retrieval, packet formatting, Wi-Fi Direct peer discovery, and duplicate-packet detection; integration checks between mesh communication, local storage, and the FastAPI backend; system-level validation of end-to-end SOS transmission, multi-hop forwarding, and offline operation; performance measurement of packet delivery ratio, latency, and packet drop rate; and user-acceptance testing focused on the ease of SOS activation and dashboard readability under simulated disaster conditions.

VI. RESULTS

The ResQNet system was developed and tested to evaluate the effectiveness of decentralized emergency communication during simulated disaster situations. Individual modules — SOS generation, GPS tracking, Wi-Fi Direct mesh communication, DTN-based store-and-forward relay, dashboard monitoring, and AI-assisted prioritization — were each validated successfully, and the resulting workflow demonstrates that a smartphone-only, infrastructure-independent system can sustain SOS delivery in the absence of cellular or internet connectivity.

System-level testing showed that SOS alerts could be transmitted and relayed reliably across multiple hops, that mesh

networking maintained continuous message delivery during simulated connectivity failures, and that the dashboard and backend services handled multiple concurrent emergency requests without instability. Performance testing across packet delivery ratio, transmission latency, packet-drop rate, node-discovery time, and resource utilization indicated stable behavior under varying node density and intermittent connectivity, while user-acceptance testing confirmed that victims could trigger and track SOS alerts with minimal guidance and that rescue-team evaluators found the map-based dashboard clear enough to support quick prioritization decisions. As development continues toward a full field deployment, subsequent work will report quantitative benchmarks for packet delivery ratio, end-to-end latency, and AI classification accuracy under larger, more heterogeneous node populations.

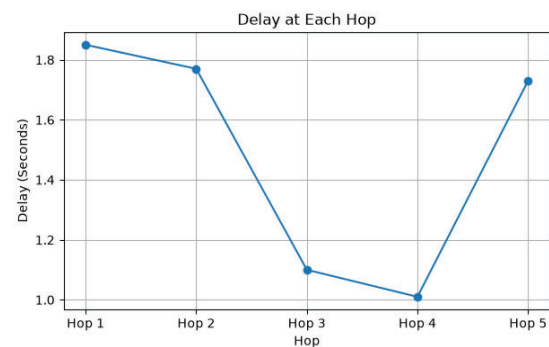


Figure 2: Simulated per-hop delay across the five-hop relay path.

VII. APPLICATIONS

ResQNet is applicable to flood, earthquake, wildfire, landslide, cyclone, and building-collapse response scenarios in which cellular infrastructure is damaged or overloaded. It is well suited to community-level disaster-preparedness deployments, government and NGO-run rescue-coordination centers, and smart-city emergency-management infrastructures. Its modular design also allows extension to satellite-assisted communication in remote areas, integration with drone-based aerial surveillance for hotspot verification, and incorporation into broader civil-defense and search-and-rescue training programs.

VIII. CONCLUSION

This paper presented ResQNet, a crowdsourced disaster-preparedness and real-time SOS mesh network that combines Wi-Fi Direct-based MANET communication, Delay Tolerant Networking, and AI-driven emergency analysis to restore emergency communication when conventional infrastructure fails. By allowing every smartphone to act simultaneously as a sender, relay, and store-and-forward node, and by attaching location, priority, and control metadata to every SOS packet,

ResQNet enables multi-hop delivery of emergency alerts to rescue authorities even in fully disconnected environments. The addition of AI-based severity classification, geospatial clustering, accessibility analysis, and a real-time rescue dashboard extends the system beyond raw message delivery into intelligent decision support for rescue coordination. Overall, ResQNet demonstrates a practical, low-cost, and infrastructure-independent approach to improving disaster communication reliability, reducing rescue response time, and enhancing victim safety.

IX. FUTURE WORK

Planned extensions include completing and benchmarking the AI-based disaster-image classification models, incorporating satellite and drone-based imagery for real-time damage assessment, extending mesh communication with Bluetooth Low Energy and LoRa for longer-range low-power relay, adding multilingual and accessibility-focused interfaces, and conducting large-scale field trials to obtain quantitative packet-delivery, latency, and classification-accuracy benchmarks across realistic disaster scenarios. Furthermore, future development will focus on enhancing the reliability and efficiency of ResQNet through adaptive routing and intelligent network resource management. A context-aware forwarding mechanism could dynamically select relay nodes based on residual battery capacity, link stability, estimated connectivity duration, hop count, and disaster severity, while prioritizing emergency packets over non-critical traffic. The integration of adaptive Time-to-Live (TTL) management, duplicate-packet suppression, and congestion-aware scheduling could reduce redundant transmissions and improve message delivery under intermittent connectivity. Additionally, end-to-end packet identifiers, delivery acknowledgements, and persistent transmission logs could support reliable recovery after network interruptions and facilitate the evaluation of forwarding performance. Future experiments will investigate these mechanisms using metrics such as packet delivery ratio, end-to-end latency, routing overhead, energy consumption, and recovery time under varying node densities and connectivity conditions.

ACKNOWLEDGMENT

The authors sincerely thank the Department of Information Science and Engineering, AMC Engineering College, Bengaluru, for providing the necessary facilities and academic support for this research. The authors express their gratitude to their project guide, Divyashree, for her valuable guidance, constructive feedback, and continuous support throughout the development of ResQNet.

REFERENCES

- [1] V. Savale, S. Patil, A. Raut, M. Phadke, R. Sundkar, and R. Dongre, "Disaster Management Ecosystem Using Crowdsourced Data, Deep Learning and NLP," in Proc. IEEE Int. Conf. Smart Computing and Networking Applications (ICSCNA), 2024.
- [2] T. Okuda, M. Ohnishi, and R. Banno, "A Study of Disaster Notification Wireless Mesh Network Applying Plumtree," in Proc. IEEE TENCON Conference, 2024.
- [3] S. K. Gupta, A. Singh, and P. Sharma, "AI-Enabled Delay Tolerant Network Framework for Emergency Communication in Disaster Scenarios," in Proc. IEEE Int. Conf. Advances in Computing, Communication and Applied Informatics (ACCAI), 2024.
- [4] I. A. Shah, A. Mushtaq, and A. Yadav, "A Novel Delay Tolerant Routing Protocol for Opportunistic Networks," in Proc. IEEE Int. Conf. Intelligent and Innovative Technologies in Computing, Electrical and Electronics (I2CT), 2024.
- [5] B. Estivil and M. Radenkovic, "SubwayMeshDTN: Exploring Opportunistic Delay Tolerant Routing Protocols for Emergency Alert Dissemination," 2024.
- [6] A. Carreras-Coch, J. Navarro, C. Sans, and A. Zaballos, "Communication Technologies in Emergency Situations," *Electronics*, vol. 11, no. 7, 2022.
- [7] R. Gupta and N. Bansal, "Priority-Based Packet Routing in Mobile Ad Hoc Networks for Emergency Applications," *Wireless Personal Communications*, vol. 124, no. 3, pp. 2431–2449, 2022.
- [8] H. Lee and S. Choi, "Efficient Duplicate Packet Suppression Technique for MANET-Based Emergency Communication," *IEEE Communications Letters*, vol. 25, no. 8, pp. 2578–2582, 2021.
- [9] M. Asif, S. Ahmad, and A. Khan, "Disaster Management Using Wireless Sensor and Ad Hoc Communication Networks: A Review," *IEEE Access*, vol. 9, pp. 123245–123263, 2021.
- [10] K. Fall, "A Delay-Tolerant Network Architecture for Challenged Internets," *IEEE Communications Magazine*, vol. 51, no. 1, pp. 36–40, 2013.