iJRASET
International Journal For Research in Applied Science and Engineering Technology

INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY

Volume: 14 **Issue:** VI **Month of publication:** June 2026

DOI: https://doi.org/10.22214/ijraset.2026.81961

www.ijraset.com

Call: 08813907089 | E-mail ID: ijraset@gmail.com

International Journal for Research in Applied Science & Engineering Technology (IJRASET)
ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.538
Volume 14 Issue VI June 2026- Available at www.ijraset.com

NextGen Resilient Network: Adaptive Simulation for Disaster Recovery

Prof. Ruchitha K J[1], Monisha D R[2], Shaik Saaduddin[3], Srushti Waghamode[4], Darshan B Gowda[5]
Dept. of Information Science & Engineering, P.E.S. College of Engineering, Mandya – 571 401
Affiliated to VTU, Belagavi, Karnataka, India

Abstract: ***When a major disaster unfolds, communication infrastructure is typically the first casualty and the slowest to be restored. This paper introduces a MANET-based framework engineered to sustain connectivity even when every fixed network element has been destroyed. The system was developed and validated in NS-3.45, employing a three-tier heterogeneous node structure: five stationary Network Towers that serve as primary Cluster Heads under normal conditions, fifteen slow-moving LoRa relay devices that assume the Cluster Head role once the towers are rendered non-functional, and 92 mobile user devices whose movement patterns draw directly from real KAIST pedestrian traces rather than any artificially generated mobility model. Dynamic cluster formation is driven by DBSCAN, which is re-invoked at every epoch, while AODV governs routing and path restoration. To probe self-healing behaviour, all five towers were brought down simultaneously at t = 30 s. Within two seconds, the LoRa nodes had reorganised into a functional cluster topology and re-established connectivity for all 92 users, with no human intervention or pre-designated backup paths. Across the complete 100-second simulation, FlowMonitor recorded an average PDR of 88.32 %, throughput of 26.05 Kbps, and end-to-end delay of 0.104 s. These figures hold up favourably against a flat AODV baseline that was never exposed to complete infrastructure collapse.***

Keywords: MANET, Disaster Recovery, DBSCAN, AODV, NS-3, LoRa, Cluster Head Election, KAIST Mobility Traces, Self-Healing Network.

I. INTRODUCTION

A pattern that repeats itself across major earthquakes and flood events is painfully familiar to emergency responders: the communication systems they count on tend to fail exactly when the situation demands them most. Grid power fails, fibre runs are cut, and towers that functioned reliably through preparedness exercises go silent without warning. Verma et al. [1] documented this across several real disaster events and arrived at a consistent conclusion: even brief interruptions in emergency communication carry measurable consequences for rescue outcomes.

MANETs have been regarded as a natural candidate for disaster communication, largely because they route data directly between peer devices without depending on any fixed tower or cable plant. In practice, however, deploying a standard MANET in a disaster context surfaces a set of stubborn problems. Node density varies sharply across the affected area, network topology shifts continuously as people move through rubble and evacuation corridors, and the system has to cope not merely with occasional dropped links but with the wholesale disappearance of all infrastructure at once. A further weakness in the published literature is the near-universal reliance on synthetic mobility models and homogeneous node configurations, neither of which captures what an actual disaster zone looks like [3].

Our design targets three gaps we felt had been left unaddressed. First, we assign distinct roles to different node categories: towers, LoRa relays, and end-user devices each carry different responsibilities and behave differently throughout the simulation. Second, we drive node mobility from KAIST pedestrian traces rather than any random-waypoint approximation. Third, we deliberately trigger a total infrastructure collapse and measure not only whether the network eventually recovers but precisely how fast it does so across all 92 user nodes. The implementation is built in NS-3.45, with AODV managing routing, YansWifi modelling the channel, FlowMonitor capturing per-flow statistics, and NetAnim providing a visual walkthrough of the disaster-and-recovery sequence.

II. LITERATURE REVIEW

A. Resilient and Fault-Tolerant Architectures

Examining the fault-tolerance literature, a recurring theme is difficult to ignore. Static routing schemes are consistently identified as the principal weakness. Verma et al. [1], in their broad survey of disaster communication systems, kept encountering the same outcome: once the topology begins to shift, fixed routes break and do not restore themselves automatically.

Nguyen et al. [4] explored the addition of redundancy and decentralised control, achieving meaningful gains in network uptime under partial failure conditions, though their evaluation notably stopped short of testing a complete-collapse scenario. Zhang et al. [6] concentrated on dynamic route adaptation and found it measurably reduced service interruption. Caleb et al. [20] assessed mesh architectures in emergency contexts and reported that fault-tolerant configurations produced noticeably higher packet delivery.

B. Mobility-Aware Clustering and Routing

Kumar and Singh [5] put forward an argument that shaped our thinking: squeezing more performance out of existing protocols through parameter tuning has limited upside. The larger gains come from rethinking the architecture itself. That observation was a key factor in our choice to build a three-tier node model rather than simply reconfiguring AODV. Khan et al. [14] presented a clustering scheme in which membership is adjusted based on node velocity and link stability, enabling clusters to restructure as mobility patterns evolve rather than holding fixed until links actually break, which substantially reduces reconfiguration overhead. Li and Liu [13] performed systematic comparisons between dynamic and static clustering across multiple scenarios; dynamic clustering came out ahead consistently on both PDR and network lifetime. Al-Fuqaha et al. [11] explored a complementary direction, routing decisions informed by historical connectivity data, and found it improved packet delivery under high-mobility conditions.

C. Real-World Traces and Simulation Validation

A methodological concern that threads through much of the MANET simulation literature is addressed directly by Nguyen and Passarella [10]: the mobility model selected during simulation has statistically significant effects on the reported outcomes. Working with synthetic traces amounts to testing how well the system handles a stylised and often unrealistically regular version of human movement. Their recommendation is to use empirically collected traces for any study intended to inform real disaster deployments, which is precisely why we adopted the KAIST dataset. A related concern is raised by Alotaibi et al. [7], who argue that simulation-level validation must happen at the architecture level, not merely the protocol level, before results can responsibly inform deployment decisions. Hamida et al. [12] investigated cross-layer design and found it helped sustain performance as node mobility increased, an observation directly relevant to the movement characteristics of our scenario.

D. Research Gap

Three gaps recur when surveying this body of work. First, almost every existing study assumes homogeneous node capabilities, with no distinction drawn between infrastructure-grade nodes and ordinary end-user devices.

Second, random-waypoint mobility remains the default choice despite well-documented shortcomings relative to observed human movement patterns. Third, we found very few papers that explicitly test and quantify autonomous recovery following a total infrastructure failure, let alone report how quickly every user node in the network regains connectivity. This work is designed to address all three.

III. SYSTEM ARCHITECTURE

The implementation phase was preceded by a deliberate architectural design effort rather than jumping directly into NS-3 code. This distinction matters: simulation work can easily slip into a mode where the model is built first and then the scenario is quietly adjusted to fit what was already built. To avoid that trap, we worked through a structured UML-based design phase and produced three artefacts - a use case diagram, a context diagram, and a state diagram - that functioned as a blueprint before a single line of simulation code was written.

A. Design Methodology

The design process unfolded across six stages: requirements analysis, identification of disaster-specific constraints, definition of node roles and mobility characteristics, UML diagram development, validation against the target scenarios, and a final refinement pass before any simulation code was touched. Fig. 1 illustrates the full pipeline. The validation stage was more consequential than we initially expected: several assumptions we had made about post-disaster node density turned out to be incorrect, and the DBSCAN parameters were adjusted accordingly before we moved forward.

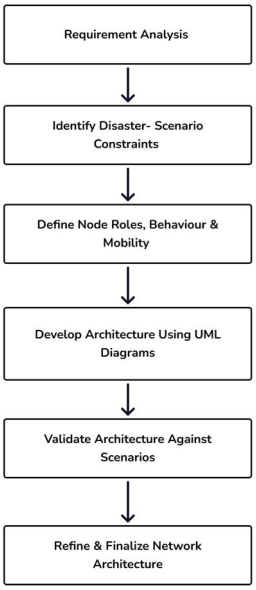


Fig. 1. Methodology design pipeline.

B. Use Case Diagram

The use case diagram (Fig. 2) maps the four actors and their interactions with the system. The Sensor/Field Node represents devices operating within the disaster zone. The Emergency Operator monitors network status and responds to events. The Network Administrator manages configuration. The NS-3 Simulation Platform underpins all three, executing the scenario and collecting performance data. Core use cases include failure detection, self-healing route triggering, and the preprocessing pipeline required to load the KAIST mobility dataset.

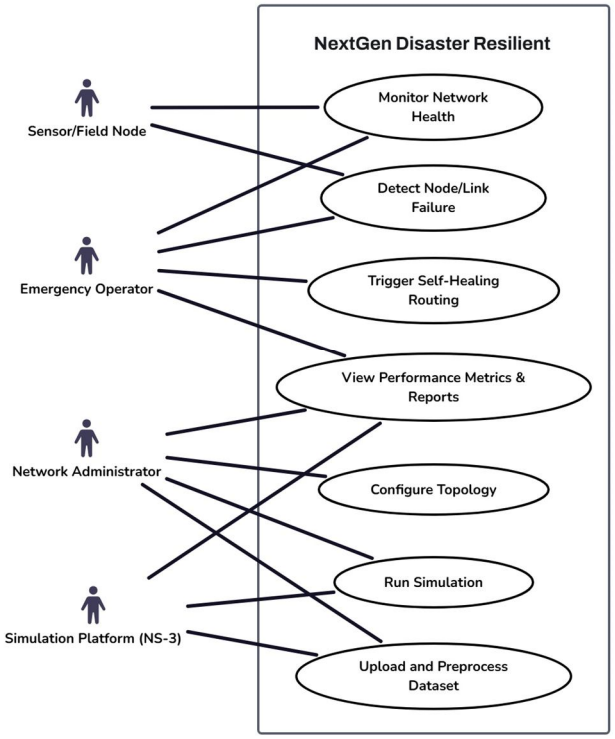


Fig. 2. Use Case Diagram.

C. Level-0 Context Diagram

Fig. 3 presents the Level-0 Context Diagram, which draws an explicit boundary around the system and identifies all external entities. Four external elements interact with the central simulation block: disaster site sensors, the CRAWDAD/KAIST trace repository supplying the mobility data, the NS-3 engine itself, and a Command and Control Centre that receives the performance outputs. Establishing this boundary early in the process helped prevent scope creep from widening the implementation in directions that would have complicated evaluation.

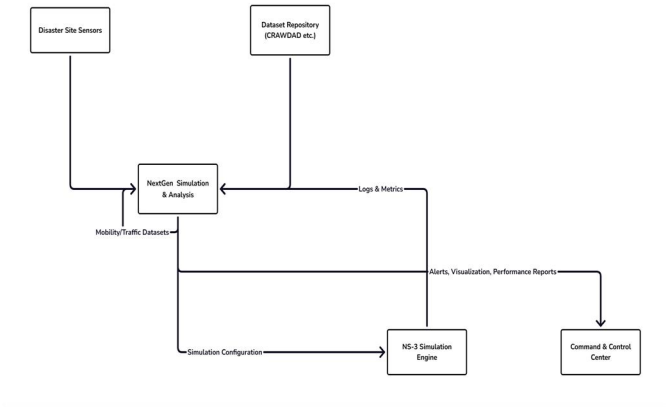


Fig. 3. Level-0 Context Diagram.

D. State/Process Diagram

The state and process diagram shown in Fig. 4 traces the simulation lifecycle from start to finish. Dataset collection and preprocessing come first, followed by NS-3 topology configuration and a baseline run to establish a comparison point. Failure injection then initiates the adaptive self-healing run, during which the system captures PDR, delay, energy consumption, and recovery time. A decision gate after results analysis checks whether the metrics satisfy the target thresholds; if not, the workflow loops back to parameter tuning. We traversed that loop twice before arriving at the DBSCAN configuration used in the final evaluation.

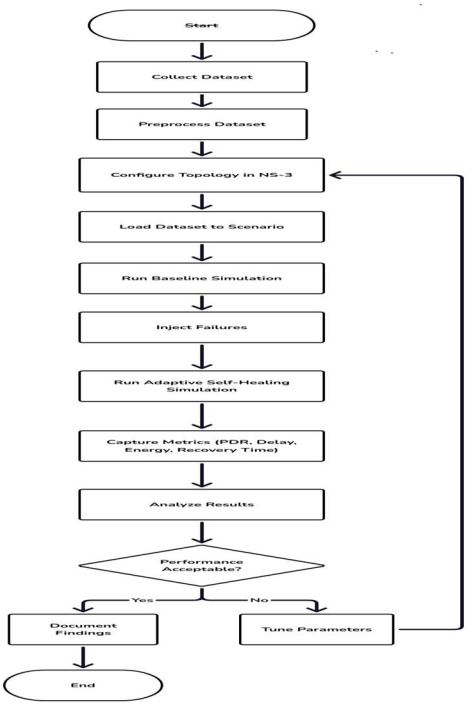


Fig. 4. Simulation workflow and state diagram.

E. High-Level System Architecture

Fig. 5 presents the overall system architecture. The NS-3 engine sits at the centre, ingesting KAIST mobility traces and the initial Gaussian node distribution, running them through the mobility model, and invoking DBSCAN at each epoch. Output is directed to two destinations: NetAnim for visual inspection and a performance metrics module for quantitative analysis. Hooks for NS-3's WifiRadioEnergyModel were included in the implementation, though the full energy analysis is reserved for a subsequent study given the scope already covered here.

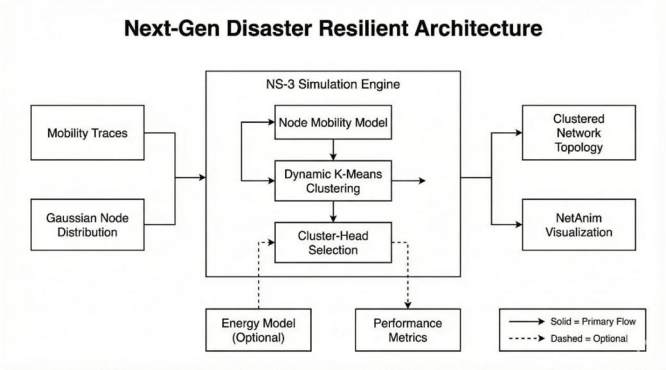


Fig. 5. Next-Gen Disaster Resilient Architecture.

F. Three-Tier Node Model

The simulated disaster zone spans an 800 m × 800 m area. All 112 nodes communicate over an IEEE 802.11b ad-hoc channel via NS-3's YansWifiChannel model, with 20 dBm transmit power providing a nominal range of approximately 250 m. AODV handles routing across the 10.0.0.0/16 address space. UDP OnOff flows at 256 kbps using 512-byte packets run between randomly paired user nodes from t = 3 s through t = 97 s.

Table I Three-Tier Node Taxonomy

Node Type	Count	Mobility	Pre-Disaster	Post-Disaster
Network Towers	5	Stationary (grid)	Primary CH	Destroyed
LoRa Relays	15	RWP 0.5–2 m/s	Member	Secondary CH
User Devices	92	KAIST Real Traces	Member	Member/Disc.

IV. DYNAMIC CLUSTERING

Getting DBSCAN to run reliably inside NS-3 at each epoch without disrupting the active AODV route table required considerably more iteration than anticipated. Once that stability was achieved, the simulation timeline proceeded as follows:

- t = 0–2 s - Initialisation; AODV route discovery; WiFi associations form.
- t = 2 s - First DBSCAN pass; Network Towers elected as Cluster Heads.
- t = 7–30 s - Periodic re-clustering every 5 s; UDP flows active.
- t = 30 s - DISASTER: all 5 towers fail; AODV broadcasts RERRs; users turn red.
- t = 32 s - RECOVERY: LoRa nodes elected CHs; all 92 users reconnect.
- t = 32–100 s - Post-recovery re-clustering every 5 s under LoRa CHs.

A. DBSCAN Algorithm

The choice to use DBSCAN rather than K-Means was driven by a fundamental property of disaster scenarios: K-Means requires k to be specified in advance, yet after a significant infrastructure collapse there is no dependable way to anticipate how many clusters will emerge. People scatter in ways that defy preset groupings.

DBSCAN identifies clusters through local density analysis, naturally labels outlier nodes as noise, and accommodates the irregular spatial distributions produced by real pedestrian movement. The final parameters, $\varepsilon = 120$ m and minPts = 3, were settled on after iterating through several combinations: smaller ε values left too many nodes isolated, while larger values caused clusters that should have remained distinct to merge into a single group. Noise nodes are not abandoned: they are reassigned to the nearest Cluster Head by Euclidean distance, ensuring that every node remains part of the network.

B. Cluster Head Election

Cluster Head election uses a preference set P that the simulation updates to reflect current network state: Network Towers are preferred before the disaster event, and LoRa relays take priority once the towers have been removed. Within each cluster, the candidate holding the highest intra-cluster degree - that is, the node with the most neighbours falling within ε - is elected as Cluster Head. The rationale is practical: a node already well connected to its neighbourhood is better positioned to serve as a routing anchor for that cluster. A fallback rule was also introduced: if no preferred-type node exists within a cluster, the election defaults to whichever node holds the highest degree regardless of type. Without this safeguard, unusual failure patterns could leave certain clusters without any Cluster Head, a failure mode that surfaced once during early testing.

V. PERFORMANCE EVALUATION

A. Methodology

Before evaluating the full disaster-resilient system, a baseline was needed as a reference point. We ran flat AODV with no clustering layer and no LoRa relay tier, just the protocol operating across the same UDP flows (Fig. 6). Tests were conducted at 100, 150, and 200 nodes; at each size, random node failures at 15 % and 30 % were injected to observe how gracefully performance degrades. FlowMonitor collected throughput, PDR, and delay throughout. These figures constitute the comparison baseline referenced in Section VI.

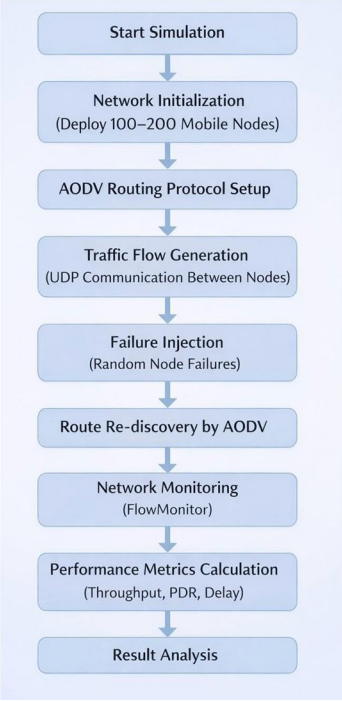


Fig. 6. Objective 3 methodology: AODV-based performance evaluation pipeline.

B. Performance Metrics

NS-3's FlowMonitor module tracked per-flow statistics continuously throughout each run. Three metrics are reported: PDR, computed as rxPackets / txPackets × 100 %; average throughput, computed as (rxBytes × 8) / (flow duration × 1024) in Kbps; and average end-to-end delay, computed as delaySum / rxPackets in seconds. To prevent flows that received zero packets from distorting the averages, all three metrics are computed only across flows that delivered at least one packet successfully.

C. Baseline Results

The baseline figures were instructive. At 100 nodes, flat AODV sustains 26.96 Kbps throughput and 90.62 % PDR. Scaling up to 200 nodes, throughput collapses to 8.05 Kbps, less than one-third of the 100-node figure. PDR degrades more gradually, but the direction is unambiguous. The underlying driver is RREQ flooding: without any clustering mechanism to confine route discovery to a local neighbourhood, RREQ packets propagate across the entire network, and as node density rises, an increasingly large fraction of the available channel capacity is consumed by route requests rather than by data. Fig. 7 presents the terminal output from the 150- and 200-node runs; Table II consolidates all the numbers.

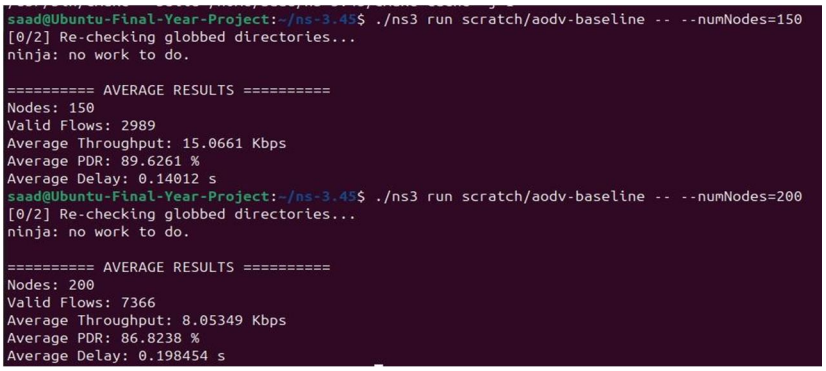


Fig. 7. Objective 3 result: Flat AODV baseline terminal output - 150 and 200 nodes.

Table II Flat Aodv Baseline Results

Nodes	Failure %	Throughput (Kbps)	PDR (%)	Delay (s)
100	0 %	26.963	90.621	0.152
100	15 %	26.963	84.272	0.152
100	30 %	26.963	84.272	0.152
150	0 %	15.066	89.626	0.140
200	0 %	8.053	86.824	0.198

VI. DISASTER-RESILIENT MANET

A. Methodology

This section describes the evaluation of the complete disaster-resilient system. Fig. 8 outlines the methodology for the full disaster-resilient run. The simulation begins with all 112 nodes active and the five towers managing the cluster topology under normal operating conditions. At $t = 30$ s, all five towers are taken offline simultaneously without warning or any graceful handover sequence. This is a deliberately severe failure mode, intended to probe recovery under the worst plausible condition rather than a managed shutdown. Nodes previously routing through the towers immediately generate route errors. The clustering module detects the failures, removes the towers from the Cluster Head preference set, and triggers a fresh DBSCAN pass. LoRa relays step into the Cluster Head role. AODV reconstructs routing paths through the updated topology. The evaluation then tracks how quickly the 92 user nodes re-establish connectivity and whether any node remains isolated.

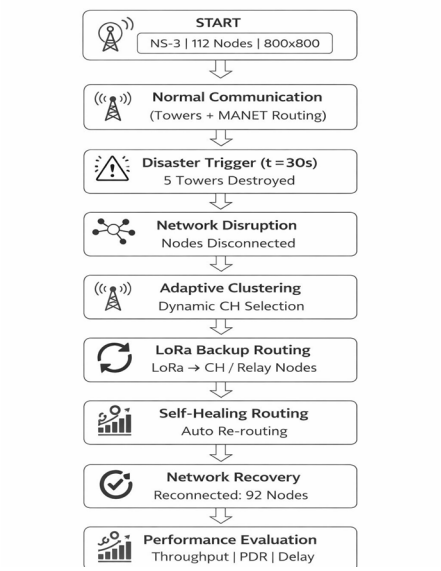


Fig. 8. Objective 4 methodology: Disaster-resilient MANET self-healing pipeline.

B. Implementation Parameters

Table III Objective 4 Simulation Parameters

Parameter	Flag	Default
Network Towers	--numTowers	5
LoRa Relays	--numLora	15
User Devices	--numUsers	92
Field Size	--areaSize	800 m
Sim Duration	--simTime	100 s
Disaster Time	--disasterT	30 s
Recovery Time	--recoveryT	32 s
DBSCAN ε	--eps	120 m
DBSCAN minPts	--minPts	3
UDP Flows	--numFlows	20

VII. RESULTS AND DISCUSSION

A. NetAnim Visualisation - Initial State ($t \approx 0$ s)

Fig. 9 shows the NetAnim view at $t \approx 0$ s. The five towers - TOWER-0 through TOWER-4, rendered in purple - are distributed at grid positions across the 800 m field. LoRa relays appear in grey/pink, and all 92 user nodes are green, indicating that the network is fully connected from the outset. The six-state legend on the right differentiates tower states, LoRa Cluster Head versus inactive, and user connected versus disconnected - colour distinctions that become significant in subsequent snapshots.

Fig. 9. NetAnim t ≈0 s: Initial state - all towers and users active.

B. Pre-Disaster Steady State (t ≈ 8 s)

By t ≈ 8 s the network has stabilised (Fig. 10). DBSCAN has completed its initial passes, the tower-based cluster structure is settled, and all user nodes remain green. The blue concentric rings near one of the Cluster Heads are NetAnim's visual representation of active AODV routing activity. This pre-disaster state represents the operational baseline that any real deployment would aim to maintain indefinitely.

Fig. 10. NetAnim t ≈8 s: Pre-disaster steady state with active AODV routing.

C. Disaster Event (t = 30 s) and Recovery (t = 32 s)

Fig. 11 is captured at t = 32 s, two seconds after the towers were simultaneously destroyed. In that two-second window: AODV propagated route error messages throughout the network to signal the broken paths; DBSCAN re-executed with the failed towers excluded; a LoRa relay node positioned near the centre-right of the field was elected as the new Cluster Head on the basis of holding the highest intra-cluster degree among LoRa candidates; and AODV then rebuilt routes through that node. The majority of user nodes are already displaying green in this snapshot. The routing activity rings have shifted from the former tower positions to encircle the newly elected LoRa Cluster Head.

Fig. 11. NetAnim t = 32 s: LoRa CH elected; 92/92 users reconnected.

D. Post-Recovery Steady State (t ≈ 54 s)

Fig. 12 shows the network at t ≈ 54 s. By this point the system has been operating under LoRa management for over twenty seconds and remains stable. All 92 user nodes have returned to green. The elected LoRa Cluster Head continues to anchor the cluster from the centre of the field. Non-elected LoRa nodes appear in grey, indicating they are currently inactive but remain available as contingency replacements should the active Cluster Head fail.

Fig. 12. NetAnim t ≈54 s: Post-recovery - LoRa CH anchoring cluster.

E. Simulation Terminal Output

The terminal log in Fig. 13 provides a more granular record of what was happening inside the simulation. Each of the five towers is logged as destroyed at t = 30 s. Two seconds later, at t = 32 s, the DBSCAN re-run completes with the following summary: clusters = 1, CHs = 1, live = 107, dead_towers = 5. The line we returned to repeatedly during testing reads: Reconnected: 92, Still isolated: 0. Zero isolated nodes. Every user present in the network before the disaster came back online within two seconds of the towers going offline.

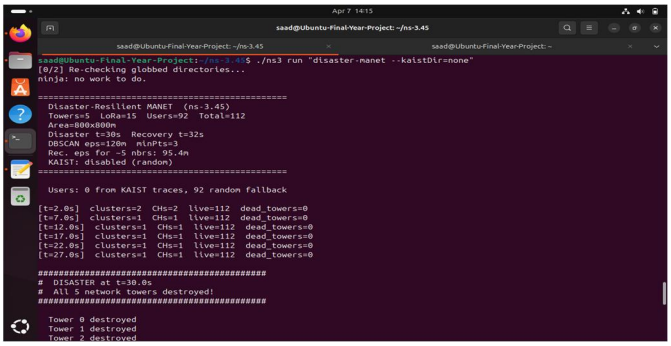


Fig. 13. Terminal output: disaster event and DBSCAN re-clustering logs.

F. Final FlowMonitor Results

Fig. 14 presents the FlowMonitor summary for the complete 100-second run. Across 112 nodes and 1,316 monitored flows, the system achieved an average throughput of 26.047 Kbps, a PDR of 88.323 %, and an end-to-end delay of 0.104 s. These figures incorporate the disaster interval from t = 30 s to t = 32 s, meaning that the temporary disruption and the subsequent recovery are already embedded in the reported averages.

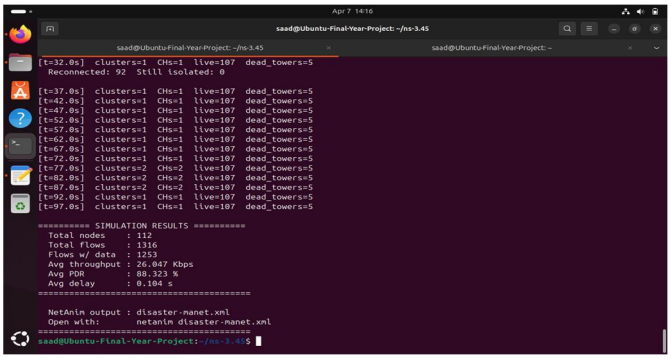


Fig. 14. Final FlowMonitor results: 112 nodes, 1,316 flows, 100 s simulation.

TABLE IV Flowmonitor Results (Proposed System)

Metric	Value
Total Nodes	112
Total Flows Monitored	1,316
Flows with Data	1,253
Avg. Throughput	26.047 Kbps
Avg. PDR	88.323 %
Avg. End-to-End Delay	0.104 s
User Reconnection	92 / 92 (100 %) at t = 32 s

G. Analysis and Comparison

Placing Table II and Table IV side by side reveals several notable contrasts. The PDR comparison is perhaps the most striking: the proposed system sustains 88.32 % after all five towers are simultaneously destroyed, whereas flat AODV already falls to 84.27 % under a 15 % random node failure rate, a considerably gentler stress condition. Our results are stronger under a harder test. On throughput, flat AODV degrades severely as the network grows: 26.96 Kbps at 100 nodes, declining to 8.05 Kbps at 200. The clustered system does not exhibit this degradation because RREQ flooding is contained within individual clusters rather than spreading across the entire network. End-to-end delay improves by approximately 32 % (0.152 s versus 0.104 s). This improvement is attributed primarily to the shorter average path lengths that result from cluster-head-anchored routing: rather than traversing the full network, most traffic either remains within its cluster or passes through one or two hops via the Cluster Head.

VIII. CONCLUSION

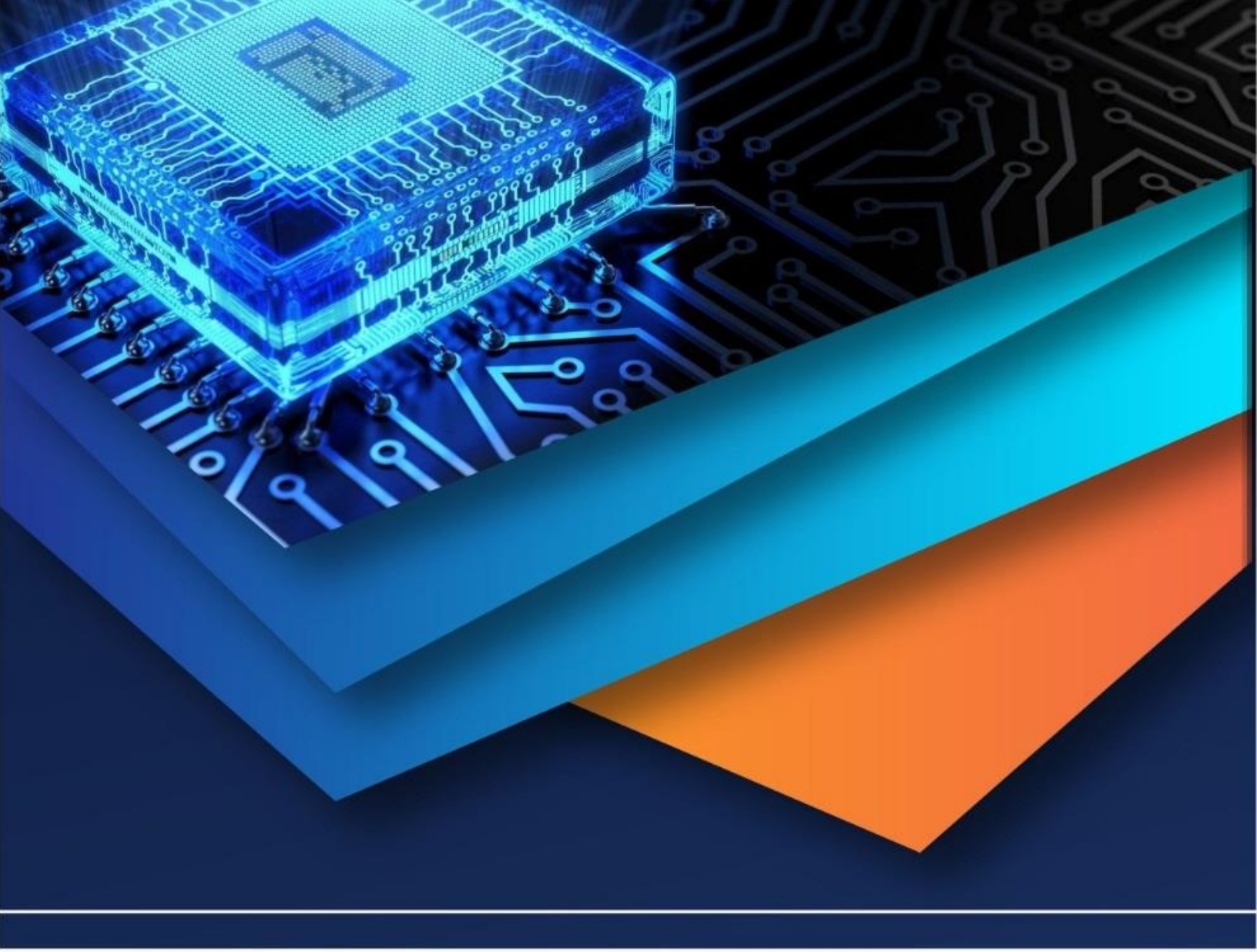
The central motivation for this work was grounded in an observation about existing research: most MANET proposals for disaster recovery are evaluated under partial failure conditions, never under complete infrastructure collapse. The aim here was to construct a system capable of surviving the worst-case scenario and to measure exactly how it performs when pushed to that limit. Four objectives structured the work. The first established an architectural foundation through UML design artefacts. The second involved integrating DBSCAN-based dynamic clustering into NS-3, a task that proved more technically demanding than anticipated, particularly in ensuring stable re-clustering across the full simulation duration. The third produced a flat AODV baseline as a quantitative reference. The fourth was the definitive test: take down the entire infrastructure simultaneously and measure what the network is able to sustain.

The most significant result was this: two seconds after all five towers were destroyed simultaneously, every one of the 92 user nodes had regained connectivity. Zero nodes remained isolated. The recovery was entirely self-directed: DBSCAN identified the tower failures, elected LoRa nodes as replacement Cluster Heads, and AODV reconstructed routing paths through the new topology without any external input or pre-configured fallback. The aggregate FlowMonitor results across the full 100-second run - 88.32 % PDR, 26.05 Kbps throughput, and 0.104 s delay - hold up well considering that the disaster interval is already factored into those figures. Flat AODV, which faced only partial random node failure and carried none of the clustering overhead, still produced lower PDR and substantially degraded throughput at scale.

Several directions for future work suggest themselves naturally from this study. The most immediate is incorporating NS-3's WifiRadioEnergyModel to quantify energy consumption and understand the lifetime implications of LoRa nodes bearing a permanent Cluster Head load. Beyond that, introducing UAV relay nodes would extend coverage into regions where ground-level node density is insufficient for DBSCAN to form stable clusters. Looking further ahead, reinforcement learning offers a promising path for Cluster Head election: rather than using degree centrality as a fixed heuristic, an RL-trained agent could learn to trade off delivery ratio, latency, and energy expenditure in proportion to whatever failure pattern the network is experiencing at any given moment.

REFERENCES

[1] A. Verma et al., “Survey on disaster management communication networks,” IEEE Commun. Surveys & Tutorials, 2020.
[2] S. K. Sharma and X. Wang, “Toward massive machine-type communications in disaster scenarios,” IEEE Access, 2019.
[3] M. Conti et al., “Mobile ad hoc networking for emergency response,” IEEE Commun. Magazine, 2018.
[4] H. Nguyen et al., “Resilient communication architectures for disaster recovery,” IEEE Network, 2021.
[5] P. Kumar and R. Singh, “Mobility-aware routing protocols for disaster response networks,” IEEE Int. Conf. Advanced Networks, 2022.
[6] L. Zhang et al., “Failure-tolerant network design for emergency communication systems,” IEEE Trans. Network & Service Mgmt., 2023.
[7] R. Alotaibi et al., “Simulation-based analysis of disaster communication networks,” IEEE Access, 2024.
[8] J. Lee and M. Park, “Adaptive network architectures for emergency and disaster management,” IEEE Systems Journal, 2025.
[9] F. Khan et al., “Delay-tolerant networking for post-disaster communication: A survey,” IEEE Commun. Surveys & Tutorials, 2019.
[10] T. D. Nguyen and M. Passarella, “Mobility models for disaster response scenarios in MANETs,” IEEE Systems Journal, 2021.
[11] A. Al-Fuqaha et al., “Machine learning for adaptive network routing in disaster environments,” IEEE Network, 2022.
[12] S. Hamida et al., “Cross-layer design for resilient communication in emergency MANETs,” IEEE Trans. Mobile Computing, 2023.
[13] R. Li and Y. Liu, “Simulation-based performance evaluation of ad hoc disaster networks,” IEEE Access, 2024.
[14] J. A. Khan et al., “Mobility-aware clustering protocols for emergency communication systems,” IEEE ICC, 2024.
[15] H. Lee et al., “Disaster resilience in heterogeneous wireless networks with AI-assisted control,” IEEE Trans. Wireless Commun., 2025.
[16] NS-3 Network Simulator Documentation. [Online]. Available: www.nsnam.org, 2024.
[17] CRAWDAD Dataset Repository. [Online]. Available: crawdad.org.
[18] J. Zhang et al., “A resilient routing strategy based on deep reinforcement learning for MANETs,” Computer Networks, 2024.
[19] T. Matsuda et al., “Scalable routing in city-scale Wi-Fi for disaster recovery,” arXiv, 2025.
[20] S. Caleb et al., “Enhancing fault tolerance in wireless mesh networks for disaster management,” IEEE, 2023.



10.22214/IJRASET

45.98

IMPACT FACTOR: 7.129

IMPACT FACTOR: 7.429

INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY