

Design and Evaluation of a Hybrid Wireless Sensor Network Combining Cluster-Tree and Mesh Routing

Nicholas Palmer

Dept. of Electrical and Computer Engineering

San Diego State University

San Diego, CA, USA

npalmer2267@sdsu.edu

Abstract—Wireless sensor networks (WSNs) are self-organizing networks of sensor nodes deployed to monitor the environment. These networks are typically battery-powered with limited energy capacity and processing capabilities, requiring energy-efficient communication protocols and resilience to network failures. To address these challenges, we propose a hybrid WSN that combines cluster-tree and mesh routing to enhance network performance. We also implement router-bridge nodes to improve energy efficiency between cluster heads. We evaluate our hybrid WSN using a Python simulation tool and compare it with existing WSNs.

Index Terms—Wireless Sensor Network, AODV, Cluster tree, Mesh routing, Bridge nodes

I. INTRODUCTION

Wireless sensor networks (WSNs) are used in a variety of applications, such as environmental monitoring, industrial automation, healthcare, and increasingly, in consumer devices. The necessity for ease of deployment and lower cost hardware have made WSNs popular in recent years, but they also pose significant challenges in terms of self-organization, energy efficiency, and robustness.

A. Self-Organization

Self-organization is a crucial advantage of WSNs, allowing deployments in remote or inaccessible locations, making it impractical to pre-configure infrastructure. Self-organization allows nodes to form a network topology and select routing paths dynamically based on the network conditions and the environment. Building an optimal topology presents many challenges given the dynamic nature of the environment, limited computational resources, and energy resources of most low-cost hardware. Many networks use one of two topologies: a hierarchical cluster-tree or many-to-many mesh topology.

• Cluster-Tree

- Hierarchical network with a single root node and multiple cluster-heads branching out to form a tree structure.
- Cluster-heads are responsible for routing data to and from their cluster, only they must maintain routing information for members of their cluster and track child networks below them.
- Centralized routing allows for greater energy efficiency as the routing can be optimized for the entire network.

- However, cluster-tree topologies can be fragile, as the failure of a cluster-head can isolate an entire cluster from the network causing a cascading failure if not handled properly.

• Mesh

- Peer-to-peer network with no central nodes or defined hierarchy.
- Each node is responsible for routing data to and from its neighbors, maintaining routing information for all its N-hop neighbors.
- Decentralized routing allows for greater robustness to node and link failures as the routing can adapt to changes in the network topology.
- However, mesh routing can lead to increased overhead and energy use as traditionally, each node must maintain routing information for all neighbors it needs to route to.

As we can see, both topologies have their advantages and disadvantages: cluster-tree topologies are more energy efficient but can be fragile, while mesh topologies are more robust but can be inefficient. These tradeoffs motivate the need for a hybrid approach that combines the strengths of both topologies to create a more robust and energy efficient network.

B. Hybrid WSNs

To address these tradeoffs, hybrid WSNs that incorporate both cluster-tree and mesh routing have been proposed. In these approaches, nodes are organized into a cluster-tree topology, but they can also use mesh links to route data between clusters. This allows for greater robustness to node failures, as nodes can use mesh links to route bypass failed cluster-heads. Additionally, the cluster-tree structure allows for more efficient routing within vertically within the tree structure of the network, further reducing the need for every member node to maintain routing information for all its cluster neighbors.

C. Contributions

In this paper, we present the design and evaluation of a hybrid WSN that combines cluster-tree and mesh routing. We first describe the design of the hybrid topology formation and routing protocol, including the use of router-bridge nodes to improve energy efficiency between cluster heads. We then implement the protocol via a Python simulation and evaluate

its performance in terms of average join time, connectivity, packet delivery, and energy use, under node and link losses. Finally, we compare the performance of our hybrid WSN with existing WSNs.

II. RELATED WORK

We briefly review prior works on cluster-tree and mesh routing protocols in WSNs, as well as existing hybrid approaches.

A. Cluster-Tree Protocols

B. Mesh Routing Protocols

C. Hybrid Approaches

D. Summary

III. SYSTEM MODEL AND ASSUMPTIONS

We describe our system model and assumptions used in our design and evaluation.

A. Network Model

We consider a network consisting of N sensor nodes deployed uniformly at random in a defined deployment area. The network designates one root node that initiates network formation and acts as a gateway, and the remaining nodes self-organize into a cluster-tree topology. We assume static node placement throughout the simulation and assume an ideal MAC layer with IEEE 802.15.4 characteristics operating at 2.4 GHz (i.e., collisions are not modeled), we do not model channels or interference for simplicity. To capture link-layer unreliability, we apply a uniform packet loss probability $P_{\text{loss}} \in [0, 1.0]$ to all transmissions, varied across experiments to evaluate protocol performance under various conditions.

B. Traffic Model

To evaluate protocol performance, we consider both many-to-one and many-to-many traffic patterns. In the many-to-one pattern, each node generates periodic data packets destined for the root node in the network. In the many-to-many pattern, each node generates periodic data packets destined for a randomly selected node in the network. Once a node has joined the network and is in the REGISTERED, CLUSTER_HEAD, or ROUTER state, it begins generating traffic at regular intervals. The packet generation interval is dynamically configurable for each simulation run, resulting in a variable data rate per node.

C. Energy Model

Each node is initialized with a finite energy (battery capacity in mAh) and consumes energy due to radio activity during transmit (TX) modes and receive (RX) mode. We model a Texas Instruments CC2420 IEEE 802.15.4 radio operating at 2.4 GHz with data rate $R = 250$ kbps and supply voltage $V = 3.0$ V.

1) *Initial Energy*: Battery capacity is configured per simulation run for all nodes in mAh and converted to energy as

$$E_{\text{init,mJ}} = C_{\text{mAh}} \cdot V \cdot 3600.$$

In our experiments, we test C_{mAh} across the range 0.25–1.0 mAh at $V = 3.0$ V, corresponding to initial energies from $E_{\text{init}} = 2700$ mJ (2.7 J) to $E_{\text{init}} = 10,800$ mJ (10.8 J). These values are lower than typical sensor node batteries to accelerate simulation time while preserving relative consumption behavior.

2) *Packet Size*: Let $B(p)$ denote the over-the-air size in bytes of packet p . This includes PHY overhead (6 B: preamble, SFD, PHR), MAC overhead (8 B), a fixed 13 B common packet header, and a byte count corresponding to the payload type:

$$B(p) = 6 (\text{PHY}) + 8 (\text{MAC}) + 13 (\text{header}) + \text{payload_bytes}(p).$$

3) *TX and RX Energy*: For a given TX power level ℓ , the CC2420 draws a current $I_{\text{TX}}(\ell)$, which we precompute into per-byte TX energies.

$$e_{\text{TX}}(\ell) \in \{0.82, 0.95, 1.34, 1.67\} \mu\text{J/byte}$$

These correspond to output power settings with $I_{\text{TX}} \in \{8.5, 9.9, 14.0, 17.4\}$ mA. We include a fixed per-packet overhead $E_{\text{start}} = 10 \mu\text{J}$ to model fixed radio energy consumption overhead.

The RX current $I_{\text{RX}} \approx 18.8$ mA yields a per-byte receive cost of $e_{\text{RX}} \approx 1.8 \mu\text{J/byte}$. Thus, the energy to transmit and receive a packet p is modeled as

$$E_{\text{TX}}(p, \ell) = E_{\text{start}} + B(p) e_{\text{TX}}(\ell), \quad E_{\text{RX}}(p) = B(p) e_{\text{RX}}.$$

The simulator tracks energy per node in mAh. Nodes whose energy reaches zero are disabled and cease all TX and RX activity.

D. Scope and Simplifications

Since our focus is on evaluating protocol-level behavior rather than hardware energy profiling, we make the following simplifications:

- We omit MCU processing, idle listening, and sleep-state power consumption, only TX and RX radio activity is modeled.
- We assume an mostly ideal MAC layer without collisions and link unreliability is modeled via a uniform packet loss probability.
- We do not model physical transmission range beyond a simple distance-based connectivity model based on transmission range and TX power level.
- No node mobility is modeled beyond sudden node failure. All nodes are static throughout each simulation run.

These simplifications allow us to evaluate the energy use and routing efficiency of the protocol itself, independent of external conditions that would vary across, implementation, hardware, and deployment.

IV. HYBRID CLUSTER-TREE + MESH NETWORK DESIGN

The proposed protocol features: a lightweight, self-organizing **cluster-tree** that guarantees connectivity in the network, and a **mesh** that can bypass the tree when a route exists via the N-hop neighbor table. The backbone provides a structured hierarchy that ensures connectivity; while the mesh opportunistically reduces hop count, latency, and energy use when routing permits.

A. Design Overview

Cluster-Tree: Nodes form clusters, each managed by a Cluster Head (CH) with a unique `NET_ID` (Network ID). Each cluster member (CM) receives a unique `NODE_ID` within that `NET_ID`. CHs connect upward toward the Root Node (RN), which allocates `NET_ID`s to CHs. This forms a tree of clusters that can route traffic between any two clusters using hierarchical parent/child relationships.

Mesh: Every node maintains an N-hop neighbor table populated by HEARTBEAT overhearing and neighbor table sharing. If a destination (or its parent CH) is reachable via the neighbor table, packets are forwarded directly (peer-to-peer) instead of traversing the tree (except for some packet types which must traverse the tree to build the hierarchy).

Dynamic Addressing: Every dynamic address is the combination of a `NET_ID` and `NODE_ID`. A dynamic address (`NET_ID.NODE_ID`) resolves to a single node at any moment, but a single physical node may be addressed by multiple dynamic addresses (e.g., acting as a CM in one cluster 1.3, while simultaneously acting as a CH for another 2.254). A globally unique 8 byte UID (EUI-64) is used to validate physical identity across the network and is used for address assignment and network registration.

A high-level figure illustrating the architecture (tree backbone with cross-layer mesh links) is given in Figure 1.

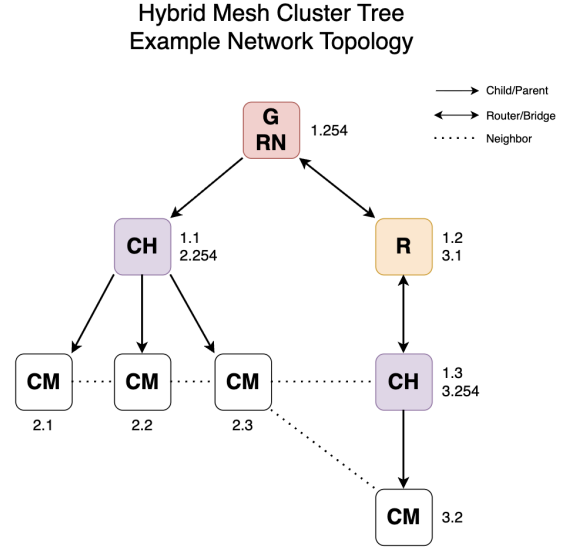


Fig. 1: A high-level architecture of the ForestNet Hybrid Cluster Tree Mesh Network

B. Roles, Capabilities, and Role Transitions

The proposed design separates what a node is currently fulfilling (attributes) from what it is capable of doing (capabilities). Broadcasting both of these fields enables local decisions such as role transfers and promotions without requiring the root node.

Roles:

- **CM:** Ordinary member of a cluster; may also act as CH/R/RN/GW if capable.
- **CH:** Manages one cluster address space; assigns `NODE_ID`s and maintains a Member Table.
- **R:** Bridges clusters when topology suggests a dedicated bridge is preferable to redundant CH chains.
- **RN:** Allocates new `NET_ID`s and acts as a CH for its local cluster (1.254 reserved).
- **GW:** Provides external network connectivity; may coincide with RN or be separate.

Promotion triggers:

- 1) **CH promotion (CM → CH):** If an unregistered node attempts to join through a CM and no reachable CH can serve it, the CM requests a `NET_ID` allocation from the RN (`NETID_REQ/NETID_RESP`) and becomes a CH upon success.
- 2) **Router formation (CM/CH → R):** If a node primarily relays between clusters (e.g., it sits between two CHs and has no local members), it may assume an R role to reduce unnecessary cluster depth.

C. Cluster Formation and Join Workflow

At the network layer, a node can discover a network, join it, and receive an address lease with no pre-configuration.

Discovery: An unregistered node broadcasts `PROBE` messages. Responders send `HEARTBEAT` messages describing

role flags, capabilities, and a `path_cost`. The joining node populates its neighbor table with the received HEARTBEAT messages and selects a candidate CH (or CM that can promote) using lowest local (distance cost) cost or network cost (path cost).

Join (CH available):

- 1) Node sends JOIN_REQ (unicast) to selected CH, including UID and capabilities.
- 2) CH assigns a NODE_ID, creates a Member Table entry, and responds with JOIN_ACK (broadcast) containing targetUid, assigned clusterAddr, and address validity time.
- 3) New CM confirms via ACK (matching seqNo); CH clears renew state.

Join (promotion required): If no CH is reachable directly, the CM triggers promotion: it requests a NET_ID from the RN, becomes CH, and then completes the join.

D. Routing Rules

Every forwarded packet carries a DST address and a NH (Next Hop) address. Forwarders rewrite NH based on local tables, while preserving the DST. This allows routing rules to be applied hop-by-hop if the packet permits.

- **Mesh:** For data packets, if DST or its CH (NET_ID.254) exists in the Neighbor Table, forward to the recorded next hop. Control packets (NETID_REQ, NETID_RESP, JOIN_ACK, ACK) bypass mesh routing to preserve the tree hierarchy.
- **Router:** A Router (R) bridges clusters by forwarding packets destined for its downstream network downstream, or packets destined for its parent to its upstream CH.
- **Tree:** CHs deliver locally if `target_net == my_net` (one of our members), or forward via the Child Net Table if the DST is a child network below the CH. If no route is found, packets are forwarded upstream to the parent CH.

Rationale: Mesh routing reduces hop count and energy cost by exploiting local connectivity. The tree provides guaranteed connectivity via a hierarchy: most nodes need only a parent entry and Neighbor Table, and only CHs need to maintain Member and Child Net Tables saving memory.

E. Self-Healing Protocol

The protocol continuously checks for stale nodes and changes in all tables to adapt to node failures and improve mobility.

Neighbor Table: Updated on any RX (including overheard packets). Entries expire after a configurable window to prevent stale routing.

Member Table (CH): Uses address leases. If a CM fails to send a HEARTBEAT for a configurable duration, the CH removes the entry and frees the NODE_ID. This recycles the NODE_ID for a new CM to join.

Child Net Table (CH): Populated by observing NETID_REQ/NETID_RESP traversals. Entries transition PENDING → VALID on promotion, becoming STALE if

TABLE I: Simulation Parameters

Parameter	Value
Network size (N)	50 nodes
Deployment area	650×650 m
Initial energy (E_0)	0.25 mAh
Traffic interval (T_s)	1.0 s
Packet loss probability (P_{loss})	0.01

unheard. Stale entries are pruned at configurable intervals if they have not been heard from in a configurable duration.

Failure handling examples:

- **Lost parent CH:** CM times out neighbor/parent; re-enters discovery and starts (PROBE) and rejoins another node.
- **Lost CH with active members:** Members lose parent; they rejoin alternate CHs rebuilding clusters outward from surviving nodes.

F. Energy and Resource Usage

Memory: Ordinary nodes store only the Neighbor Table, bounded by node density and the configured maximum hop count (N-hop). CHs additionally maintain the Member Table (bounded by cluster size) and Child Net Table (bounded by child networks).

Energy: Neighbor learning via passive overhearing of HEARTBEAT messages requires no TX energy. Nodes dynamically adjust TX power based on distance to the next hop, reducing energy consumption and extending network lifetime.

Hop Count: Mesh forwarding reduces hop count in dense networks by enabling horizontal movement. In sparse networks, tree routing guarantees reachability via the tree hierarchy. Without mesh links, packets must traverse upward before descending to the destination cluster.

V. SIMULATION AND EXPERIMENTS

We implement the hybrid protocol in a modified version of WSNLAB, a discrete-event Network Simulator written in Python. The network, traffic, and energy models follow Section III. Unless otherwise noted, experiments use the default parameters in Table I.

A. Metrics

We evaluate the protocol using the following metrics:

- **Join time:** Time from node power-on until successful registration with a CH.
- **Connectivity:** Fraction of nodes with a valid route to the sink over time.
- **Packet delivery ratio (PDR):** Fraction of generated packets successfully delivered.
- **Node lifetime:** Time until a node depletes its energy budget. Nodes with $E_i(t) \leq 0$ are permanently disabled.
- **Network lifetime:** Time until connectivity drops below 80%, capturing when the network is no longer operationally useful.

B. Results

We organize experiments into three categories: (1) scalability, varying network size from 25–300 nodes; (2) reliability, varying packet loss probability P_{loss} ; and (3) energy, varying initial energy E_0 and traffic interval T_s . We also compare tree-only routing against the hybrid tree+mesh configuration.

1) *Scalability*: Fig. 2 shows the average join time as a function of network size. Join time increases from approximately 7.2 s at 25 nodes to 13 s at 300 nodes. Interestingly, a plateau occurs between 150–250 nodes, we suspect this is where denser deployments provide more CH candidates nearby, reducing discovery latency. The increase at 300 nodes suggests packet congestion during formation in very dense networks.

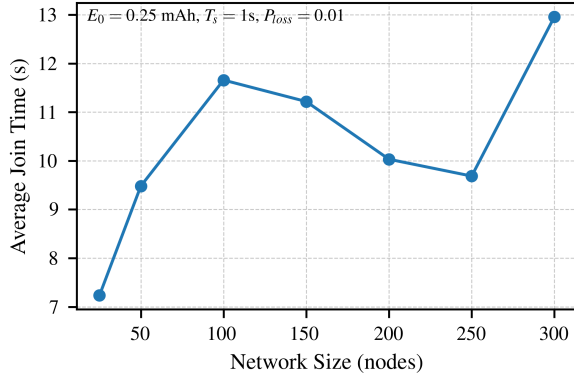


Fig. 2: Average join time versus network size.

2) *Network Lifetime vs. Initial Energy*: Fig. 3 shows a mostly linear relationship between initial energy E_0 and network lifetime. At $T_s = 1$ s, lifetime increases from 650 s to 2850 s as E_0 grows from 0.1 to 0.5 mAh. High traffic ($T_s = 0.1$ s) reduces lifetime by approximately 45% compared to low traffic, as lifetime ranges from 400 s to 1500 s. The widening gap at higher energy levels indicates that traffic load becomes the primary factor when energy budgets are less constrained.

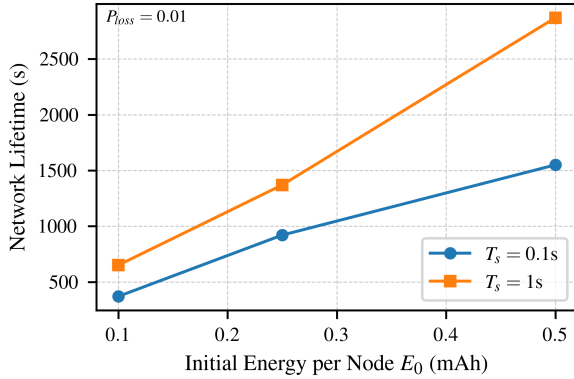


Fig. 3: Network lifetime vs. initial energy E_0 for different traffic intervals.

3) *Reliability Under Packet Loss*: Fig. 4 shows cumulative packet delivery ratios under varying packet loss probabilities. The protocol maintains fairly good delivery rates up to $P_{\text{loss}} = 0.01$, achieving an average of approximately 75% cumulative delivery ratio. At $P_{\text{loss}} = 0.1$, cumulative delivery ratio drops below 60%, as multi-hop packet loss compounds.

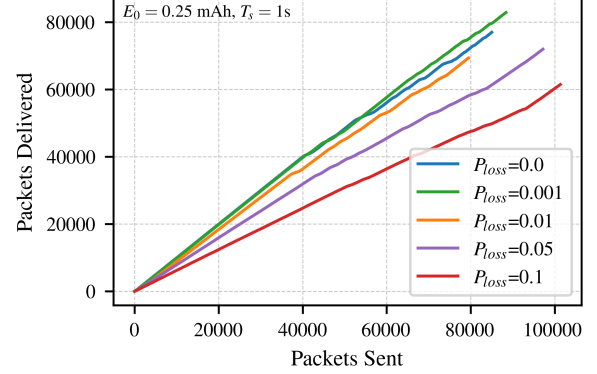


Fig. 4: Cumulative packets delivered vs. sent under different packet loss probabilities.

4) *Tree vs. Mesh: Network Lifetime*: Fig. 5 compares network lifetime between tree-only and hybrid tree+mesh routing. The hybrid approach consistently outperforms tree-only across all 3 energy levels: with minor improvement at $E_0 = 0.1$ mAh (600 s vs. 500 s) and at 0.25 mAh (1400 s vs. 1200 s), and a more substantial improvement at 0.5 mAh (2700 s vs. 2400 s). The average network lifetime increases with energy level, confirming that mesh reduces hop count and energy per packet over the entire lifetime of the network.

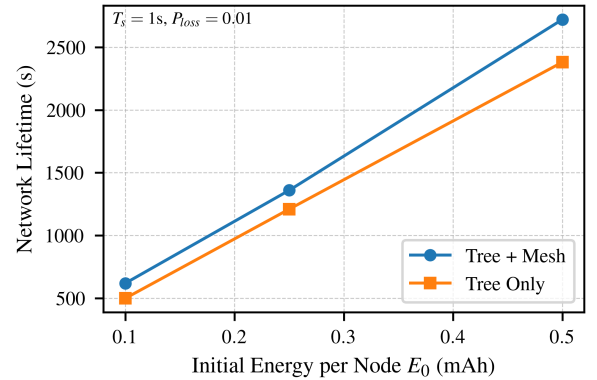


Fig. 5: Network lifetime: tree-only vs. tree+mesh routing.

5) *Tree vs. Mesh: Average Packet Delivery Ratio*: Fig. 6 shows average PDR over time for both routing modes. The hybrid approach achieves a peak PDR of 95% compared to 93% for tree-only. More importantly, tree-only PDR drops sharply to 84% at approximately 1000 s as CH failures disrupt routes, while tree+mesh maintains 88–89% PDR until 3700 s. The PDR advantage demonstrates that mesh links provide

alternative paths when tree links fail, improving network resilience.

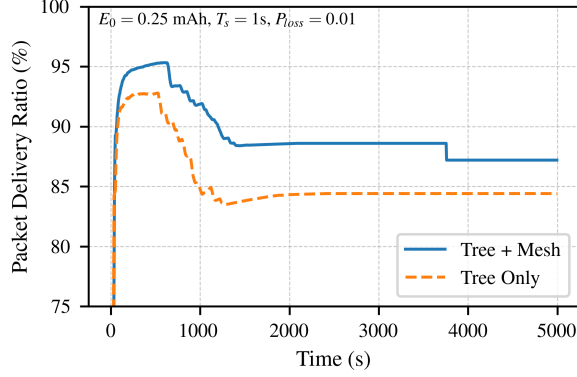


Fig. 6: PDR over time: tree-only vs. tree+mesh routing.

6) *Impact of Traffic Load:* Fig. 7 shows connectivity over time for different traffic intervals. Network lifetime (80% connectivity threshold) varies dramatically: approximately 900 s for $T_s = 0.1$ s, 1300 s for $T_s = 1$ s, 1600 s for $T_s = 5$ s, and 1800 s for $T_s = 10$ s. Notably, all curves exhibit steep “cliff” drops when critical nodes fail—connectivity collapses rapidly rather than degrading gradually. This behavior is not surprising, as critical nodes in the network deplete faster due to forwarding and control overhead along the cluster tree.

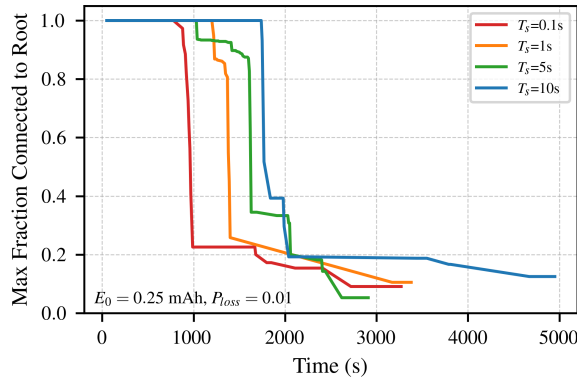


Fig. 7: Connectivity over time for different traffic intervals T_s .

7) *Role-Based Energy Depletion:* Fig. 8 presents the CDF of node lifetimes by role. Routers die earliest with a median lifetime of approximately 1500 s, followed by cluster heads at 2000 s, while leaf nodes survive longest with a median of 2500 s. The spread between roles confirms significant energy use imbalance—critical nodes deplete faster due to forwarding and routing overhead. This explains why the network disconnects while many leaf nodes still have substantial remaining energy. The results suggest that CH rotation or energy aware routing could significantly extend network lifetime.

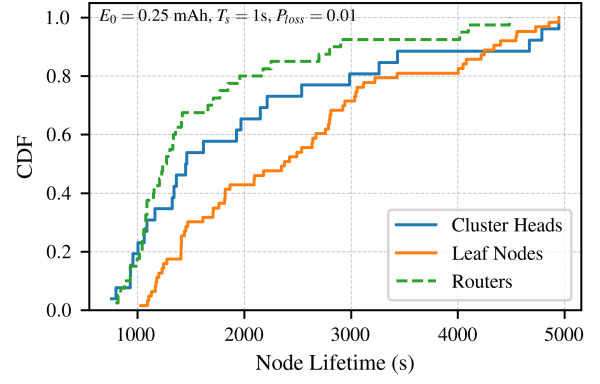


Fig. 8: CDF of node lifetimes by role.

VI. CONCLUSION AND FUTURE WORK

REFERENCES

- [1] W. Heinzelman, A. Chandrakasan, and H. Balakrishnan, “Energy-efficient communication protocol for wireless microsensor networks,” in *Proceedings of the 33rd Annual Hawaii International Conference on System Sciences*, 2000, pp. 10 pp. vol.2-.
- [2] C. Perkins and E. Royer, “Ad-hoc on-demand distance vector routing,” in *Proceedings WMCSA’99. Second IEEE Workshop on Mobile Computing Systems and Applications*, 1999, pp. 90–100.
- [3] D. B. Johnson, D. A. Maltz, J. Broch *et al.*, “Dsr: The dynamic source routing protocol for multi-hop wireless ad hoc networks,” *Ad hoc networking*, vol. 5, no. 1, pp. 139–172, 2001.
- [4] S. Lindsey and C. Raghavendra, “Pegasis: Power-efficient gathering in sensor information systems,” in *Proceedings, IEEE Aerospace Conference*, vol. 3, 2002, pp. 3–3.
- [5] S. C. Ergen, “Zigbee/ieee 802.15. 4 summary,” *UC Berkeley*, September, vol. 10, no. 17, p. 11, 2004.
- [6] O. Younis, M. Krunz, and S. Ramasubramanian, “Node clustering in wireless sensor networks: recent developments and deployment challenges,” *IEEE Network*, vol. 20, no. 3, pp. 20–25, May 2006.
- [7] T. M. Behera, S. K. Mohapatra, U. C. Samal, M. S. Khan, M. Daneshmand, and A. H. Gandomi, “Residual energy-based cluster-head selection in wsns for iot application,” *IEEE Internet of Things Journal*, vol. 6, no. 3, pp. 5132–5139, June 2019.
- [8] M. Abolhasan, T. Wysocki, and E. Dutkiewicz, “A review of routing protocols for mobile ad hoc networks,” *Ad Hoc Networks*, vol. 2, no. 1, pp. 1–22, 2004. [Online]. Available: <https://www.sciencedirect.com/science/article/pii/S157087050300043X>