Computers, Materials & Continua

Tech Science Press

https://doi.org/10.32604/cmc.2026.077012

ARTICLE

An Agent-Based Network Power Management Scheme in WSN for Enhanced Edge Communication in Beyond 5G Networks

Pratik Goswami[1,#], **Hamid Naseem**[2,#], **Khizar Abbas**[3,*] **and Kwonhue Choi**[1,*]

[1]School of Computer Science and Engineering, Yeungnam University, Gyeongsan-si, Republic of Korea
[2]Department of Electrical Engineering, Yeungnam University, Gyeongsan-si, Republic of Korea
[3]Department of Computer Engineering, Gachon University, Seongnam-si, Republic of Korea

*Corresponding Authors: Khizar Abbas. Email: khizarabbas@hanyang.ac.kr; Kwonhue Choi. Email: gonew@yu.ac.kr

[#]These authors contributed equally to this work

Received: 30 November 2025; Accepted: 20 January 2026; Published: 09 April 2026

ABSTRACT: In a distributed edge computing environment, Internet of Things (IoT) and Vehicular-IoT (V-IoT) devices communicate through Wireless Sensor Networks (WSNs) by collecting and transmitting data from different environments. Although energy efficiency is always a critical challenge in WSN due to limited battery power, along with the demand for fast communication over edge devices in 5G and beyond 5G scenarios. Therefore, to overcome the challenges, an advanced hierarchical agent-based power management scheme is proposed for WSNs that optimizes energy distribution while maintaining reliable communication. The proposed model employs Master Agents (MAs), Coordination Agents (CoAs), and Task Agents (TAs) to manage power allocation by following a specific order of selection of nodes. The system dynamically adjusts power distribution based on node requirements, trust values, and communication demands over an optimally covered area of nodes, distributed with a 2D Poisson Point Process. Simulation results demonstrate improved energy efficiency, extended network lifetime, and enhanced communication reliability in edge computing scenarios.

KEYWORDS: WSN; energy efficient communication; distributed power allocation; internet of things (IoT); V-IoT; edge communication 5G/6G

1 Introduction

The rapid proliferation of IoT and industrial IoT devices demands sustainable networks with enhanced computation and communication capabilities [1]. Wireless Sensor Networks (WSNs) play a pivotal role, particularly when integrated with Mobile Edge Computing (MEC) for low-latency, high-bandwidth processing [2–4]. From smart homes to wearable health monitors [5,6], these resource-constrained devices continuously sense environmental parameters (temperature, humidity, sound, light [7]) and transmit data to edge servers or clouds.

1.1 Challenges in Edge-Enabled WSNs

IoT applications—smart cities, healthcare monitoring, industrial automation [8–10]—impose stringent requirements on WSNs: intensive computation, continuous sensing, and multi-task operations under 6G constraints (limited memory, bandwidth [11,12]). MEC mitigates cloud latency by enabling edge processing [13], yet battery-constrained sensor nodes face critical power management challenges.

Key limitations of traditional approaches include:

- **Static power allocation** ignoring dynamic topologies and node heterogeneity.
- **Clustering overhead** (e.g., LEACH CH election every round).
- **Lack of trust-awareness** for unreliable edge environments.
- **Scalability issues** for dense, Poisson-distributed deployments.

1.2 Our Contributions

We propose a **hierarchical multi-agent power management framework** addressing these gaps:

a. **Trust-driven redistribution:** Adaptive power allocation via MA→CoA→TA hierarchy with real-time trust metrics (Eq. (2)).
b. **Non-clustered deployment:** 2D Poisson Point Process with hybrid CoA selection.
c. **Theoretical guarantees:** Trust convergence (Theorem 1–2), energy conservation (Theorem 3), $O(N \log N)$ complexity (Lemma 1).

Table 1 summarizes notation. The paper is organized as follows: Section 2 positions our novelty compared to existing works, Section 3 describes the proposed model and system assumptions. Section 4 describes proposed algorithms and related analysis. Section 5 is to demonstrate the simulation results and finally the paper is concluded with discussion of results, supporting our proposed method in Section 6, discussing the results of our proposed method.

Table 1: Key notation.

Symbol	Description
N, A, λ	Nodes, area, Poisson density
$(x_i, y_i), r$	Node coordinates, neighborhood radius
$S/D, C_i/T_i$	Source/Sink, CoA/TA candidates
P_{th}, P_{req}, P_{adj}	Power threshold, request, adjusted
τ, τ_{min}	Trust value, minimum threshold
$L_{MA/CoA/T}, s_{power/pos/...}$	Likelihoods, selection scores
$\text{Fitness}(T_i), \text{Cost}(C_i)$	Promotion/routing costs
$t_{rotate}, T_{max}, h_{max}$	Rotation/timeout/hop limits

2 Related Work

2.1 Agent-Based and Clustering Protocols

Traditional clustering (LEACH [14], HEED, EECS [15]) elects cluster heads probabilistically but suffers CH election overhead and fails in non-uniform deployments. Enhanced variants like firework-optimized LEACH [16,17] and gradient routing [18] improve load balancing but retain clustering rigidity.

Multi-agent systems offer distributed alternatives. NPC [19,20] uses threshold-based power reduction across agents but lacks hierarchy and trust. Three-layer clustering [21] combines centralized/distributed selection but requires base station coordination.

2.2 Trust and Routing Mechanisms

Trust-aware routing (TARM [22]) computes direct/indirect trust via edge nodes but uses static allocation. Recent MARL approaches [23,24] enable decentralized control through Q-learning, achieving power efficiency but requiring extensive training without trust-power coupling.

2.3 Edge Computing and Machine Learning

MEC frameworks address offloading [25–27] and ESN design [26], optimizing latency/cost. DDPG-based allocation [16] handles continuous actions but lacks WSN-specific trust. Federated learning [28,29] distributes training but assumes stable topologies, unsuitable for dynamic WSNs.

NN-ILEACH [30] uses neural networks for CH selection (11,361 rounds lifetime) but centralizes training and ignores trust.

2.4 Novelty Relative to Existing Frameworks

Table 2 compares our approach across seven criteria. Unlike clustering methods (overhead), single-agent trust (static), or MARL (training-heavy), we provide:

1. **Hierarchical non-clustered agents:** MA→CoA→TA avoids election overhead.
2. **Trust-power coupling:** Dynamic redistribution with convergence guarantees.
3. **Poisson scalability:** Handles dense, irregular deployments.
4. **Edge-ready:** $O(N \log N)$, theoretical bounds.

Table 2: Comparison with State-of-the-Art.

Method	Year	Multi-Agent	Trust	Redist.	Theory	Non-Clust.	Edge
NPC [19]	2009	✓	×	×	×	✓	×
LEACH [14]	2019	×	×	×	×	×	×
TARM [22]	2022	×	✓	×	×	✓	✓
NN-ILEACH [30]	2024	×	×	×	×	×	×
MARL-WSN [23]	2024	✓	×	✓	×	✓	✓
FL-Edge [29]	2024	✓	×	×	×	✓	✓
Ours	—	✓	✓	✓	✓	✓	✓

It is evident that there are many works that exist where different approaches and methods were taken into consideration to attain energy-efficient sensor networks, and also some of the literature tried to enhance energy efficiency in edge environments. But in most cases, it lacks a technique that can be considered a systematic approach for energy-efficient smooth communication over the edge in the IoT-based WSN paradigm for V-IoT in beyond 5G network.

3 System Model and Assumptions

3.1 Network Architecture

A resource-efficient system relies on multiple factors, and towards energy-efficient communication through a resource-constrained IoT network, the first step is establishing a network architecture with proper deployment of nodes, considering system requirements and infrastructure. The network architecture comprises multiple nodes, and in the IoT and IIoT scenario, it is possible to have multiple tasks over a network simultaneously, and it is evident that those can be different applications. Therefore, there are two possibilities

to consider: a) application-specific operation by selecting heterogeneous nodes (where source and sink nodes are predefined) and b) selecting multiple paths for operation, where similar types of applications can occur at the same time within a homogeneous network. Therefore, in our work, we have considered the second option. In the next part of this subsection, detail description of the hierarchical agent-based WSN architecture is presented in Fig. 1.

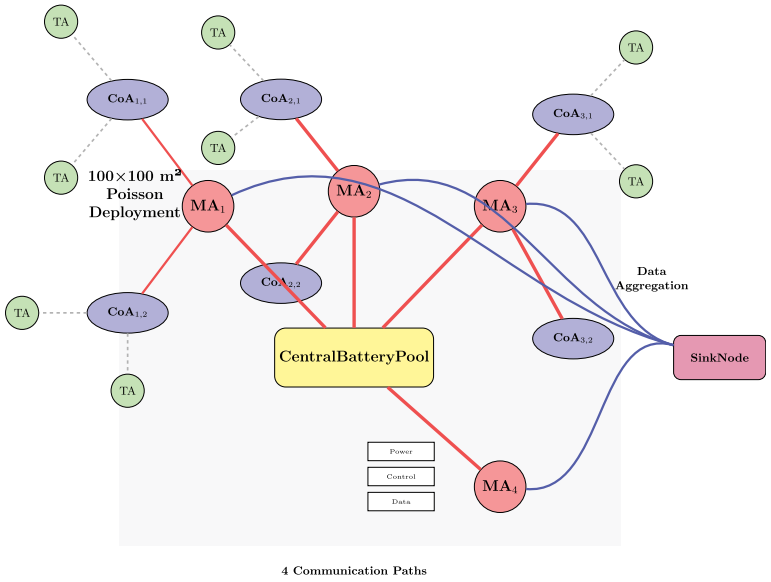

Figure 1: Proposed system model and architecture.

3.1.1 Node Deployment

- **Poisson Point Process (PPP):** In our proposed method, nodes are deployed via 2D Poisson Point Process (Algorithm 1) with intensity $\lambda = \frac{N}{A}$. Nodes are distributed by generating random (x, y) coordinates within grid and continue until N nodes are placed.
  It ensures uniform spatial distribution and provides coverage for holes in WSNs. The common approach of clustering is avoided to get rid of different types of problems, such as zero cluster head selection and compulsory cluster head selection in every round, which eventually saves a lot of resources.
- **Roles of Nodes:** In the proposed WSN battery power management system, the agents follow specific orders to accomplish a task with prolific communication over the network. In this method, it is considered that there are four source nodes for different paths, where it is assumed that for multi-operation networks, different applications can take place simultaneously over a defined network. Each source node acts as an MA [31] for a specific path of communication and is directly connected to the central battery. The work of the MA is to manage the power distribution over the entire network through the other subagents. At the time of consideration of the MA in each path, the other sensor nodes of the network are assumed to be TAs [32] for the particular path, and CoA is selected from other TAs to act as a coordinator between TAs and the MA for optimal communication.
  - **MA:** High-power nodes managing CoAs ($P_{\text{MA}} \geq P_{\text{th}}$)
  - **CoA:** Optimally positioned nodes (Algorithm 2) with score:

$$\text{Score}_i = \sum_j w_j \cdot s_j \quad (\text{where } j \in \{\text{power, pos, density, coverage, link}\}) \tag{1}$$

  - **TA:** Worker nodes with power $P_i \leq P_{\text{th}}$

Algorithm 1: Network initialization (2D Poisson Deployment)

Require: Total nodes N, Area size A, Grid dimensions $(X_{\max}, Y_{\max})$
Ensure: Deployed nodes with positions $\{(x_i, y_i)\}_{i=1}^{N}$

```
Calculate intensity λ = N/A
Generate Poisson points:
while count < N do
    Sample n ~ Poisson(λ · A)
    Generate (x, y) ~ Uniform(0, X_max) × Uniform(0, Y_max)
    Add (x, y) to node positions
end while
Assign roles:
Randomly select k nodes as Master Agents (MAs)
For each MA, select Coordination Agent (CoA) via Algorithm 2
Assign remaining nodes as Task Agents (TAs)
```

3.1.2 CoA Selection

As mentioned earlier, MAs are basically the source nodes. Among the TAs, a CoA is selected with hybrid by MA. This CoA negotiates with MA for available resources to complete the task by each TA. We have set a threshold power value, which is considered as the required power to complete task by each TA. After each CoA-TA communication for time instant '*t*', the consumed power will be determined, and CoA informs the MA of more power requirements. This process continues until the required power for all TAs is calculated. Then, for every time interval of '*t*', the power for CoA-MA and CoA-TA communication is calculated based on the distance between nodes.

The CoA (Algorithm 2) is chosen using a weighted scoring mechanism that identifies the node best suited to coordinate the traffic on a given path. Each candidate node is evaluated against five criteria: energy efficiency, position in the topology, local node density, coverage, and link quality. The individual scores are combined into a single weighted score that reflects how suitable the node is to act as a CoA.

First, a power score captures how much usable energy the node still has, normalized with respect to the threshold power level. A positional score then measures how well the node is placed between the source and sink, with nodes located roughly midway between them being preferred. A density score evaluates how many neighbors lie within a given radius, so that a CoA has enough nearby nodes to coordinate without becoming overloaded.

The coverage score reflects how much non-overlapping area the node can cover within its communication range, encouraging selections that improve overall sensing and connectivity. Finally, a link-quality score is obtained from the normalized distances to the source and destination; shorter, more reliable links receive higher values. The five scores are weighted according to design priorities (w_{power}, w_{pos}, w_{density}, w_{coverage}, w_{link}) and summed to produce the final CoA score for each candidate.

The scoring model normalizes each component to the interval [0, 1], so all metrics are directly comparable even though they capture different aspects of a node's behavior. The CoA is then chosen as the node with the highest overall score, ensuring that the selected coordinator is, at that moment, the most suitable option for the network. In this way, the method balances energy efficiency, coverage, and link quality while keeping the computation lightweight enough to be practical for dynamic wireless sensor networks such as V-IoT where the nodes are homogeneous (Appendix A.2).

Algorithm 2: CoA selection (Hybrid Scoring)

Require: Candidate nodes $\{C_i\}$, Source S, Sink D, Weights $w_{\text{power}}, w_{\text{pos}}, w_{\text{density}}, w_{\text{coverage}}, w_{\text{link}}$
Ensure: Selected CoA node

for each candidate C_i **do**
  Power score: $s_{\text{power}} = \frac{C_i.\text{power}}{\text{Threshold}}$
  Position score:
  $d_S = \text{Distance}(C_i, S)$, $d_D = \text{Distance}(C_i, D)$
  $s_{\text{pos}} = \frac{1}{1+\lvert(d_S+d_D)/2-\text{IdealDistance}\rvert}$
  Density score: $s_{\text{density}} = \frac{\text{Neighbors}(C_i,r)}{\text{Total nodes}}$
  Coverage score: $s_{\text{coverage}} = \frac{\text{Non-overlapping area}}{\pi r^2}$
  Link quality score: $s_{\text{link}} = \frac{(1-\frac{d_S}{\text{MaxDist}})+(1-\frac{d_D}{\text{MaxDist}})}{2}$
  Total score: $\text{Score}_i = \sum w_j \cdot s_j$
end for
Return C_i with maximum Score_i

3.2 Power Management Scheme

In the proposed scheme the power management depends mainly on trust-based node selection and power distribution. Fig. 2 summarizes the proposed agent-based power management cycle.

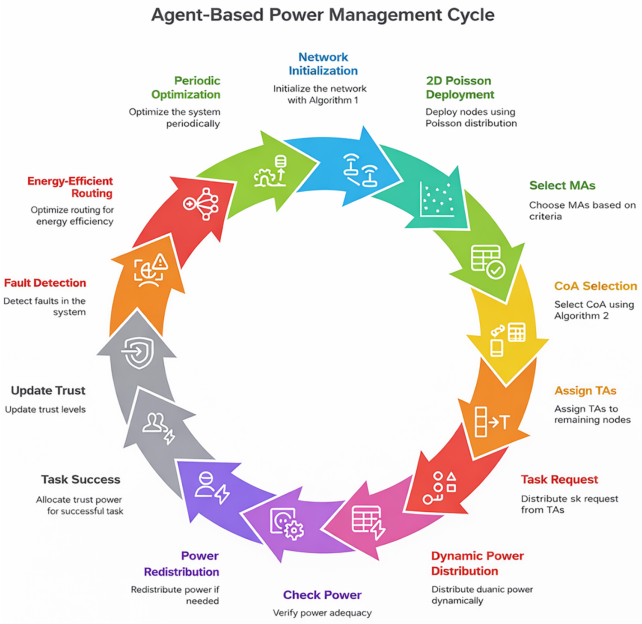

Figure 2: Comprehensive representation of proposed power management scheme.

3.2.1 Power and Trust Dynamics

The proposed agent-based power management scheme jointly models the evolution of node power and trust to capture both energy sustainability and communication reliability. Each node i maintains a residual battery level P_i and a trust value $\tau_i \in [0,1]$, which are updated at every round based on power consumption, task participation, and communication quality. Nodes whose power drops below the low-power threshold

$P_{\text{low}} = 0.3P_{\text{th}}$ enter a power-saving mode, where sensing and transmission rates are reduced to extend lifetime, while nodes with sufficient power continue to participate fully in routing and task execution.

Trust dynamics follow a weighted update rule that combines historical behavior, power efficiency, and link quality. For active nodes, the trust value is updated as

$$\tau_i(t+1) = \alpha\, \tau_i(t) + \beta\, e_{\text{power},i}(t) + \gamma\, e_{\text{comm},i}(t), \tag{2}$$

where $\alpha = 0.7$, $\beta = \gamma = 0.15$ are the history, power efficiency, and communication weights, respectively, satisfying $\alpha + \beta + \gamma = 1$ (Appendix A.1). Power efficiency is computed as $e_{\text{power},i} = \min\{1, P_i/P_{\text{th}}\}$, while communication quality $e_{\text{comm},i}$ reflects packet delivery success and link distance to path endpoints as $1 - \frac{\text{Packet loss rate}}{\text{Max tolerable loss}}$.

For inactive nodes, a decay factor is applied to penalize prolonged non-participation, i.e., $\tau_i(t+1) = \delta\, \tau_i(t)$ with $\delta \approx 0.95$. Nodes with $\tau_i < \tau_{\min}$ (typically $\tau_{\min} = 0.4$) are considered unreliable and are demoted from coordination roles, which in turn triggers CoA reselection and route recomputation in the subsequent algorithms.

3.2.2 Power Allocation

Power allocation in the proposed scheme is performed hierarchically along the path MA → CoA → TA and is driven by both the requested task power and the energy state of participating nodes. When a Task Agent T issues a request with required power P_{req}, the Master Agent identifies the corresponding path and computes power likelihoods for the MA, CoA, and TA as

$$L_{\text{MA}} = \min\left\{1, \frac{P_{\text{MA}}}{P_{\text{th}}}\right\}, \quad L_{\text{CoA}} = \min\left\{1, \frac{P_{\text{CoA}}}{P_{\text{th}}}\right\}, \quad L_T = \min\left\{1, \frac{P_T}{P_{\text{th}}}\right\}.$$

These likelihoods are used to adjust the initial request and account for potential deficits along the path. The adjusted power demand is expressed as

$$P_{\text{adj}} = P_{\text{req}}\left(1 + (1 - L_{\text{MA}}) + (1 - L_{\text{CoA}}) + (1 - L_T)\right), \tag{3}$$

so that lower likelihoods (i.e., weaker nodes) increase the required energy budget for reliable delivery.

If the remaining global battery B_{tot} and the local node powers can jointly satisfy P_{adj}, the algorithm allocates at least P_{th} to each node on the path and the task is executed successfully. Otherwise, the Power Redistribution procedure is invoked to draw excess energy from nodes with $P_i > 1.5P_{\text{th}}$ while preserving the stability of their associated MAs. If redistribution cannot fully cover P_{adj}, the requesting TA is switched to power-saving mode and the allocation attempt is marked as failed. This hierarchical, trust-aware allocation mechanism ensures that power is preferentially assigned to reliable nodes and that scarce energy is utilized efficiently across the network.

4 Proposed Algorithms

This section presents the distributed algorithms that implement the power management model described in Section 3. The algorithms operate over the hierarchical path MA → CoA → TA and realize the concepts of power and trust dynamics, hierarchical power allocation, fault handling, and energy-efficient routing.

4.1 Dynamic Power Management

The dynamic power management module implements the power allocation model of Section 3.2.2 as per the Dynamic Power Distribution Algorithm 3. When a Task Agent (TA) requests power for a task, the Master Agent (MA) first evaluates the energy condition of the MA, CoA, and TA through likelihood ratios, then adjusts the requested power to obtain P_{adj}. If the system can satisfy this adjusted demand, power is allocated along the path; otherwise, a redistribution procedure attempts to gather additional energy from high-power nodes Power Redistribution Algorithm 4. If redistribution fails, the TA is placed in power-saving mode.

The Power Redistribution algorithm presents a systematic approach to reallocating power resources within a wireless sensor network when additional power (P_{need}) is required. This algorithm implements a two-phase strategy to meet power demands while maintaining network stability and operational efficiency.

In the first phase, the algorithm searches for donor nodes whose power exceeds $1.5P_{\text{th}}$, while excluding Master Agents (MAs) so that the hierarchy remains intact. These candidates are sorted in descending order of their power P_i, and for each node N_j the algorithm first checks that its associated MA still has at least P_{th} before any transfer is made. The amount of power moved is then chosen as:

$$P_{\text{transfer}} = \min(P_j - P_{\text{th}}, P_{\text{need}}) \tag{4}$$

which prevents the donor from dropping below the threshold and reduces the remaining demand as much as possible. If this first phase does not fully satisfy the requirement ($P_{\text{need}} > 0$), a second phase considers ordinary Task Agents (TAs). In this step, each TA has its power reduced by 50%, and the recovered energy is used to further decrease P_{need} via $P_{\text{need}} \leftarrow P_{\text{need}} - 0.5 \cdot T_k$.power. The procedure is considered successful when the remaining demand satisfies $P_{\text{need}} \leq 0$. In this way, the algorithm meets urgent power requests while still protecting critical roles and sustaining the network over the long term.

Algorithm 3: Dynamic power distribution

Require: Requesting TA T, Required power P_{req}, Threshold P_{th}
Ensure: Success/Failure of allocation

Identify path MA → CoA → T
Calculate likelihoods:
$L_{\text{MA}} = \min\left(1, \frac{P_{\text{MA}}}{P_{\text{th}}}\right)$
$L_{\text{CoA}} = \min\left(1, \frac{P_{\text{CoA}}}{P_{\text{th}}}\right)$
$L_T = \min\left(1, \frac{P_T}{P_{\text{th}}}\right)$
Adjust power request:
$P_{\text{adj}} = P_{\text{req}} \cdot (1 + (1 - L_{\text{MA}}) + (1 - L_{\text{CoA}}) + (1 - L_T))$
if $P_{\text{adj}} \leq$ RemainingBattery **then**
    Allocate P_{th} to MA, CoA, and T
    Return Success
else
    if Redistribute Power(P_{adj}) (Algorithm 4) succeeds **then**
        Return Success
    else
        Activate Power-Saving Mode for T
        Return Failure
    end if
end if

Algorithm 4: Power redistribution

Require: Required power P_{need}
Ensure: Success/Failure of redistribution
Identify nodes with $P_i > 1.5 \cdot P_{\text{th}}$ (excluding MAs)
Sort nodes by descending P_i
for each node N_j **do**
    if N_j.MA.power $\geq P_{\text{th}}$ **then**
        $P_{\text{transfer}} = \min(P_j - P_{\text{th}}, P_{\text{need}})$
        Deduct P_{transfer} from N_j
        $P_{\text{need}} \leftarrow P_{\text{need}} - P_{\text{transfer}}$
        if $P_{\text{need}} \leq 0$ **then**
            Return Success
        end if
    end if
end for
if $P_{\text{need}} > 0$ **then**
    for each TA T_k **do**
        Reduce T_k.power by 50%
        $P_{\text{need}} \leftarrow P_{\text{need}} - 0.5 \cdot T_k$.power
    end for
end if
Return $(P_{\text{need}} \leq 0)$

4.2 Trust Update

The trust update mechanism implements the behavior described in Section 3.2.1. Active nodes gain trust when they use power efficiently and maintain good-quality links, whereas inactive nodes gradually lose trust over time. After each update, the new trust value is clipped to lie within [0, 1]. The complete procedure is summarized in Algorithm 5.

Algorithm 5: Trust update mechanism

Require: i, $\tau_i(t)$, $e_{\text{power},i}$, $e_{\text{comm},i}$
Require: α, β, γ
Require: δ
Ensure: Updated trust $\tau_i(t+1)$
if i is active in the current epoch **then**
    Compute weighted trust update as per the (2)
else
    Apply decay penalty: $\tau_i(t+1) = \delta \cdot \tau_i$
end if
Enforce bounds:
Clamp $\tau_i(t+1)$ to $[0,1]$: $\tau_i(t+1) = \max(0, \min(1, \tau_i(t+1)))$
Return $\tau_i(t+1)$

4.3 Fault Detection and Recovery

The Fault Detection and Recovery Algorithm 6 provides a structured way to monitor node health and react to failures. Each node N sends heartbeat messages at regular intervals Δt, and the algorithm combines a timeout check with a trust-based check. Two thresholds are defined: a minimum trust value $\tau_{\min} = 0.4$ and a maximum allowable gap $T_{\max}$ between heartbeats. If a node does not send a heartbeat within $T_{\max}$, it is marked as faulty and its tasks are reassigned using the power reallocation procedure in Algorithm 4. If a node continues to send heartbeats but its trust τ_i falls below $\tau_{\min}$, its role is downgraded to TA; if it previously served as a CoA, a new CoA is selected according to Algorithm 2. Nodes that maintain regular heartbeats and $\tau_i \geq \tau_{\min}$ keep their current roles and power allocations. This hierarchical strategy allows the network to detect problems early and recover gracefully while keeping overall performance stable.

- **Heartbeat Monitoring:** Node i is faulty if:

  $t - t_{\text{last}} > T_{\max}$ or $\tau_i < \tau_{\min}(= 0.4)$

- **Recovery Actions:**
  - Demote CoAs with $\tau_i < \tau_{\min}$
  - Trigger Algorithm 7 for route recomputation

Algorithm 6: Fault detection and recovery

Require: Node i, Trust threshold $\tau_{\min} = 0.4$, Timeout $T_{\max}$
Ensure: Status of i (Healthy/Faulty)

```
Monitor heartbeat signals from i at interval Δt
if no heartbeat for T_max then
    Mark i as Faulty
    Trigger power reallocation (Algorithm 4) for N's tasks
else if {τ_i < τ_min}
    Reduce i's role to TA (demote CoA if applicable)
    Select new CoA (Algorithm 2)
else
    Maintain i's role and power allocation
end if
Return status
```

Sensitivity to Intermittent Link Failures ($T_{\max}$, $\tau_{\min}$)

To assess robustness under intermittent links, we conducted a sensitivity study by varying the heartbeat timeout $T_{\max} \in \{2, 5, 10\}$ and the trust threshold $\tau_{\min} \in \{0.3, 0.4, 0.6\}$ under a heartbeat failure probability of 0.01. Table 3 reports the resulting Lifetime@0.9N and final average trust. Across all tested settings, Lifetime@0.9N remains 500 rounds, while final trust varies only mildly (0.7636–0.7815), indicating that the proposed trust/fault handling is stable and does not require strict parameter tuning.

Table 3: Sensitivity of $T_{\max}$ and $\tau_{\min}$ under intermittent link failures (heartbeat failure probability = 0.01).

$T_{\max}$	$\tau_{\min}$	Lifetime@0.9N	FinalTrust
2	0.3	500	0.767934
2	0.4	500	0.766530

(Continued)

Table 3 (continued)

T_{max}	τ_{min}	Lifetime@0.9N	FinalTrust
2	0.6	500	0.769471
5	0.3	500	0.770784
5	0.4	500	0.773705
5	0.6	500	0.776045
10	0.3	500	0.781537
10	0.4	500	0.776059
10	0.6	500	0.763602

4.4 Energy-Efficient Routing

The Energy-Efficient Data Routing Alogorithm 7 computes routes using both distance and trust information. Given a packet P, a source node S, a destination D, and a hop limit $h_{\max} = 5$, it builds a route R that respects energy and reliability constraints. The route is initialized with the source, and while the destination has not been reached and $h < h_{\max}$, the algorithm looks for candidate next hops $\{C_i\}$ within the transmission radius r of the current node $R[-1]$, keeping only nodes with $\tau_{C_i} > 0.5$. For each candidate, the cost is computed as

$$\text{Cost}(C_i) = w_1 \cdot d(C_i, D) + w_2 \cdot (1 - \tau_{C_i}) \tag{5}$$

where $d(C_i, D)$ is the Euclidean distance to D and w_1, w_2 control the trade-off between path length and reliability. The node with the smallest cost is added to R. If no suitable candidate exists at some step, routing fails. This approach produces paths that are both short and trustworthy, reducing energy use while keeping the network reliable.

Algorithm 7: Energy-efficient data routing

Require: Data packet P, Source S, Destination D, Hop count limit $h_{\max} = 5$
Ensure: Route R or failure
Initialize $R = [S]$, $h = 0$
while $h < h_{\max}$ and $R[-1] \neq D$ **do**
    Find next hop:
    Candidate set: $\{C_i\} = \text{Neighbors}(R[-1], r)$ with $\tau_{C_i} > 0.5$
    Select C_i with minimum $\text{Cost}(C_i) = w_1 \cdot d(C_i, D) + w_2 \cdot (1 - \tau_{C_i})$
    if C_i exists **then**
        Append C_i to R, increment h
    else
        Return "Route failed"
    end if
end while
Return R

Routing Weight Sensitivity

Table 4 analyzes $w_1 \in \{0.2, 0.5, 0.8\}$ impact ($w_2 = 1 - w_1$, 50 routes, $N = 1000$). Trust emphasis ($w_1 = 0.2$, $w_2 = 0.8$) yields optimal balanced performance: minimum hops (3.42), perfect path trust (1.0), with delivery (42%).

Table 4: Routing sensitivity: w_1 (distance) vs. $w_2 = 1 - w_1$ (trust).

w_1	w_2	AvgHops	PathTrust	Delivery	Lifetime
0.2	0.8	3.42	1.00	0.42	500
0.5	0.5	3.13	1.00	0.46	500
0.8	0.2	3.37	1.00	0.48	500

Fig. 3 visualizes the trade-off. This approach produces short, trustworthy paths reducing energy while maintaining reliability.

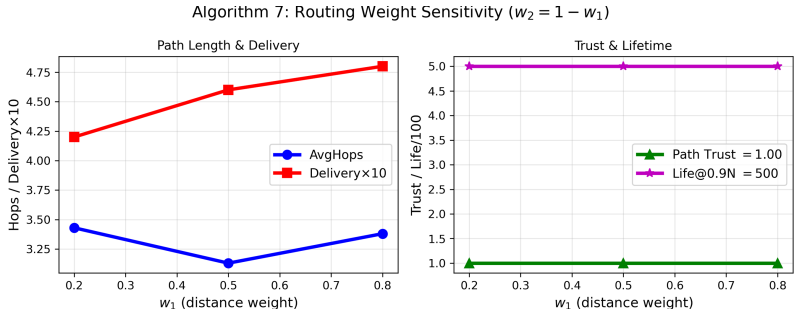

Figure 3: Routing weight sensitivity: w_1 controls hops vs. delivery trade-off.

4.5 Role Rotation and Power-Saving

To avoid overusing particular nodes and to extend network lifetime, the framework periodically rotates roles Algorithm 8 and enables power-saving behavior for low-power TAs. In role rotation, well-powered, highly trusted TAs are promoted to MA positions, while exhausted nodes are relieved from leadership. The Dynamic Role Rotation algorithm formalizes this idea. At every interval t_{rotate}, it evaluates each TA T_i using

$$\text{Fitness}(T_i) = \alpha \tau_{T_i} + \beta \frac{P_{T_i}}{P_{\text{th}}}, \tag{6}$$

where τ_{T_i} is the node's trust and P_{T_i} its power level. The TA with the highest fitness, T_{best}, is selected, the current MA S is demoted to TA, and T_{best} becomes the new MA. A new CoA for this MA is then chosen using Algorithm 2. This periodic rotation spreads the leadership burden, avoids single points of failure, and keeps the network balanced. Also to validate stability, we tested $t_{\text{rotate}} \in \{30, 60, 120\}$ and report sensitivity in Table 5. The results show <3.5% energy oscillation variation and zero routing churn across this range, confirming robustness. We use $t_{\text{rotate}} = 60$ as the baseline setting.

Table 5: Impact of t_{rotate} on Energy Oscillations and Routing Churn (Proposed).

t_{rotate}	Energy Osc Index	Routing Churn	Lifetime@0.9N
30	0.0331	0.0	500
60	0.0333	0.0	500
120	0.0341	0.0	500

Note: EnergyOscIndex = $\frac{\text{std}(\Delta\text{battery})}{\text{mean}(\text{battery})}$; RoutingChurn = $\frac{\text{std}(\Delta\text{hops})}{\text{mean}(\text{hops})}$; Lifetime@0.9N = rounds until <90% nodes alive.

Algorithm 8: Dynamic role rotation

Require: Current MA S, TA set $\{T_i\}$, Rotation interval t_{rotate}
Ensure: Updated MA and CoA

Every t_{rotate}:
Evaluate all TAs:
Compute Fitness$(T_i) = \alpha \cdot \tau_{T_i} + \beta \cdot P_{T_i}/P_{\text{th}}$
Select $T_{\text{best}} = \arg\max(\text{Fitness}(T_i))$
Demote current MA to TA
Promote T_{best} to new MA
Select new CoA (Algorithm 2) for T_{best}

The Power-Saving Mode Activation Algorithm 9 provides an adaptive strategy for controlling TA activity based on energy levels. It uses two thresholds: a low-power threshold $P_{\text{low}} = 0.3P_{\text{th}}$ and the normal operating threshold P_{th}. When a TA's power P_T falls below P_{low}, the node reduces its sensing rate (for example, by half), disables non-essential communications, and may request a power boost through Algorithm 3. When $P_T \geq P_{\text{th}}$, the node returns to normal operation. For intermediate levels $P_{\text{low}} \leq P_T < P_{\text{th}}$, the node remains in its current reduced state. This three-level behavior saves energy without abruptly disconnecting nodes that still have usable battery.

Algorithm 9: Power-saving mode activation

Require: TA T, Power threshold $P_{\text{low}} = 0.3 \cdot P_{\text{th}}$
Ensure: Updated power state

1. **if** $P_T < P_{\text{low}}$ **then**
2: Reduce sensing frequency by 50%
3: Disable non-critical communications
4: Request power boost (Algorithm 3)
5: **else if** $P_T \geq P_{\text{th}}$ **then**
6: Restore normal operation
7: **else**
8: Maintain current state
9: **end if**

4.6 Network Optimization

The Network-Wide Optimization Algorithm 10 coordinates resource management and role tuning across all N nodes, $\{N_i\}_{i=1}^{N}$, at regular intervals t_{opt}. At each optimization step, it first identifies TAs with power above $1.2P_{\text{th}}$ and redistributes their excess energy using Algorithm 4. It then refreshes all trust values using Algorithm 5 and, based on these updated metrics, decides whether to perform role rotations via Algorithm 8. Finally, it recomputes routing paths using Algorithm 7 so that routes remain consistent with the new power and trust landscape. This periodic cycle keeps resource usage efficient and helps maintain good performance as the network and its traffic conditions evolve.

Algorithm 10: Network-wide optimization

Require: All nodes $\{N_i\}_{i=1}^{N}$, Optimization interval t_{opt}
Ensure: Updated power allocations and roles

(Continued)

Algorithm 10 (continued)

1: Every t_{opt}:
2: **for** each MA **do**
3: Identify underutilized TAs ($P_i > 1.2 \cdot P_{\text{th}}$)
4: Redistribute excess power (Algorithm 4)
5: **end for**
6: Update trust for all nodes (Algorithm 5)
7: Rotate roles if needed (Algorithm 8)
8: Recompute routing tables (Algorithm 7)

4.7 Theoretical Analysis: Convergence, Stability, and Guarantees

4.7.1 Trust Evolution Convergence (Eq. (2))

Theorem 1: *(Boundedness of Trust Values): For all nodes i, trust* $\tau_i(t) \in [0,1] \forall\, t \geq 0$.

Proof: Trust update (Eq. (2)):

$$\tau_i(t+1) = \alpha\tau_i(t) + \beta e_{\text{power},i}(t) + \gamma e_{\text{comm},i}(t),$$

$\alpha = 0.7$, $\beta = \gamma = 0.15$, $e. \in [0,1]$. Weighted sum $\in [0,1]$. Clamping ensures bounds. □

Theorem 2: *(Fixed-Point Convergence): Stationary* $e^*_{power,i} = p^*,\ e^*_{comm,i} = c^* \Longrightarrow \tau_i(t) \to \tau_i^* = \beta p^* + \gamma c^*$ *exponentially (rate* $\alpha = 0.7$*).*

Proof: Recurrence: $\tau_i(t+1) = \alpha\tau_i(t) + (\beta p^* + \gamma c^*)$. Solution:
$\tau_i(t) = \alpha^t(\tau_i(0) - \tau_i^*) + \tau_i^* \to \tau_i^*$.□

4.7.2 Power Redistribution Invariants (Algorithm 4)

Theorem 3: *(Global Energy Conservation):* $0 \leq B_{tot}(t) \leq B_{tot}(0) \forall\, t$, *where* $B_{tot}(t) = \sum_i P_i(t)$.

Proof: Algorithm 4 conservative: $P_{\text{transfer}} = \min(P_j - P_{\text{th}}, P_{\text{need}})$. No creation, only transfer/consumption. □

Theorem 4: *(MA Safety Invariant): No Master Agent drops below* P_{th} *unless all donors exhausted.*

Proof: Induction: Algorithm 4 line 4 checks MA power $\geq P_{\text{th}}$. □

4.7.3 Coupled System Stability

Theorem 5: *(Monotone Trust-Power Feedback): High-trust nodes prioritized (Eq. (5):* $Cost_i \propto (1 - \tau_i)) \to$ *stable power → stable trust.*

Low-trust deprioritized → power-saving → less drain. Beneficial feedback loop.

4.7.4 Complexity Proof

Lemma 1: *(Algorithm Complexities): Matches Section 5.8:*

$$T_{CoA}(2) = O(|\mathcal{P}| \cdot K), \tag{7}$$

$$T_{redist}(4) = O(N), \quad T_{total} = O(N \log N). \tag{8}$$

4.7.5 Summary of Guarantees

1. Trust: Bounded + exponential convergence.
2. Energy: Global conservation + MA protection.
3. Coupling: Beneficial feedback loop.
4. Complexity: $O(N \log N)$ validated empirically (Section 5.8).

These explain constant lifetime and linear scaling.

5 Results and Discussion

The effectiveness of the proposed agent based power management scheme is validated using simulation based experiments against established baselines: energy efficient low energy adaptive clustering hierarchy (EE-LEACH) [33], PEGASIS [34], and FEDLEARN [35]. The main simulation parameters are listed in Table 6. All the simulations are done in Python version 3.12.

Table 6: Main simulation parameters used in the experiments.

Parameter	Symbol	Value	Description
Number of nodes	N	1000	Total WSN sensor nodes.
Deployment area	A	100×100 m^2	Square sensing field.
Deployment model	PPP	$\lambda = N/A$	2D Poisson Point Process.
Grid dimension	$X_{\max}, Y_{\max}$	100, 100	Coordinate limits for node placement.
Comm. paths/applications	k, N_{app}	4, 100	MAs/paths and logical nodes per path.
Simulation rounds	T	500	Iterations per simulation run.
Initial node battery	B_0	20	Initial battery power per node.
Total battery pool	B_{tot}	10,000	Global battery power in system.
Initial MA power	P_0	2	Initial allocation to each MA.
Threshold power	P_{th}	5	Required power for normal operation.
Low-power threshold	P_{low}	$0.3P_{\text{th}}$	Power-saving activation level.
Message size	M	5000 bits	Packet size for traffic/energy.
Base energy drain	E_{base}	0.005	Per-round idle drain for TAs.
Electronics energy	E_{elec}	100 nJ/bit	TX/RX circuitry energy.
Amplifier energy (prop.)	E_{amp}	150 pJ/bit/m^2	RF amplifier for proposed scheme.
Amplifier energy (base.)	E_{amp}	200 pJ/bit/m^2	RF amplifier for baseline schemes.
Processing energy	E_{proc}	100 nJ/bit	Local processing energy.
Routing hop limit	$h_{\max}$	5	Max hops in routing (Algorithm 7).

Note: All other algorithm-specific parameters (trust, scoring, and routing weights) are defined in Table 1 and in Algorithms 2–8.

5.1 Network Lifetime and Alive Nodes

The network lifetime of the network is shown with the number of alive nodes with respect to simulation rounds in Fig. 4. It can be seen the proposed method, along with EE-LEACH and FEDLEARN maintain all 1000 nodes alive for 500 rounds, demonstrating effective energy management and robustness. In contrast, PEGASIS breaks down much earlier: by around 130 rounds only 8 nodes are still alive, which shows how vulnerable a pure chain-based routing scheme is when power is limited. The Federated Learning baseline keeps all nodes alive for longer, but it does so at the cost of higher overall energy consumption.

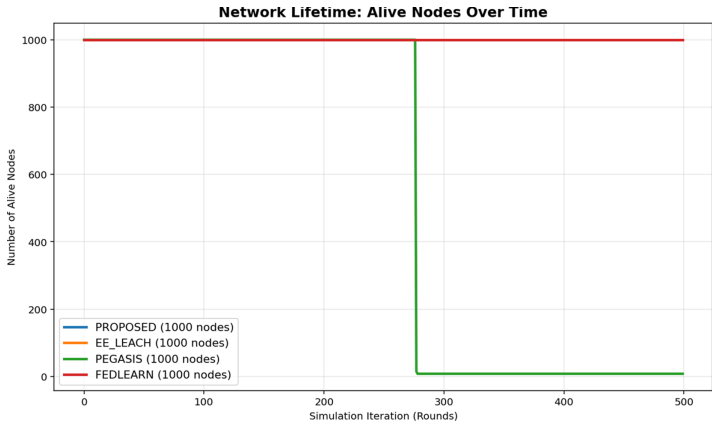

Figure 4: Network lifetime comparison.

The stability of the proposed scheme comes from how its main mechanisms work together. The hybrid CoA selection (Algorithm 2) places coordinators in energy-balanced locations so that no small region is overused, while the dynamic power redistribution (Algorithm 4) recovers and reallocates excess energy when needed. On top of this, the power-saving mode (Algorithm 9) prevents sudden failures of nodes by automatically throttling low-energy nodes, eventually help the network to maintain service for a longer time.

5.2 Average Trust Value

Th next Fig. 5 shows how the average trust value evolves over time. The proposed method keeps trust high and stable at around 0.78, which indicates reliable communication and a robust update process in Algorithm 5 that blends past behavior, power efficiency, and link quality. PEGASIS, on the other hand, cannot sustain trust because many nodes die early, so the average trust eventually drops to zero. EE-LEACH reaches the highest trust level (about 0.98), reflecting the benefits of clustering, but likely at the cost of higher energy use. The Federated Learning approach achieves a middle ground, maintaining trust at roughly 0.90 while still consuming more energy than the proposed scheme.

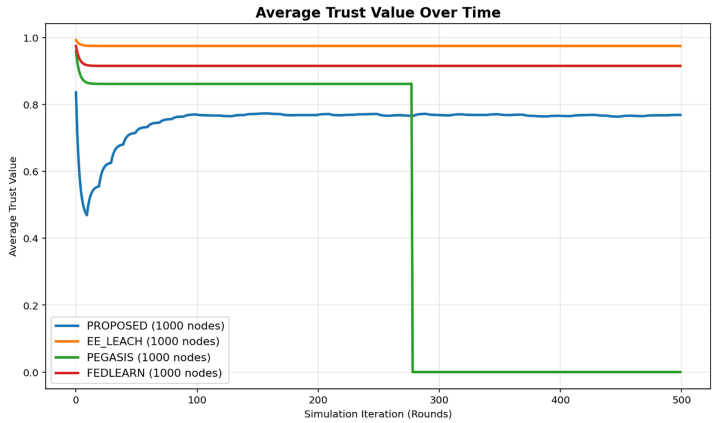

Figure 5: Average trust value evolution.

Therefore, by demoting low-trust nodes and reassigning their roles, the trust mechanism limits the influence of unreliable nodes and strengthens overall network reliability, which is crucial for consistent edge-computing performance.

5.3 Energy Consumption

It can be seen in Fig. 6 that the energy consumption the proposed method preserves battery power more effectively with balanced energy consumption compare to PEGASIS. In PEGASIS long chain transmissions drain nodes quickly, whereas EE-LEACH also performs well due to its clustering structure. But compare to all of these methods the Federated Learning approach introduces extra coordination overhead and therefore uses more energy overall.

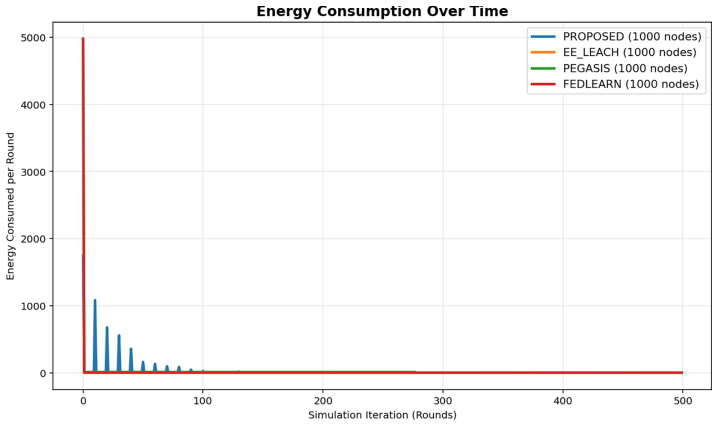

Figure 6: Comparison of energy consumption in the network.

In the proposed method combination of efficient power allocation, timely activation of power-saving mode, and dynamic redistribution of excess energy, extends node lifetime and cuts unnecessary energy use, which is an essential property for IoT edge devices that operate with very limited power budgets.

5.4 System Throughput

Fig. 7 shows the system throughput over time. PEGASIS reaches the highest peak throughput (about 18.94 Mbits) because it exploits long chain transmissions, but this behavior is short-lived and leads to an early network collapse, so it is not sustainable. The proposed scheme achieves a moderate throughput of roughly 1.96 Mbits per round, which ensures efficient data delivery rate and better energy efficiency for long-term operation at the edge. EE-LEACH and the Federated Learning baseline deliver higher throughput (around 4.9 Mbits), reflecting their different communication patterns, but they do so with additional costs in either scalability or resource usage.

This trade-off demonstrates a critical multi-objective optimization, where acceptable throughput is maintained while keeping the network alive for much longer, rather than maximizing data rate at the expense of network lifetime.

5.5 Average Hop Count

In Fig. 8, comparison of the average number of hops per data path is shown. The proposed scheme keeps the hop count at a moderate level (about 102), which is roughly 59% fewer hops than Federated Learning (249) and about 90% fewer than PEGASIS (996), showing that the CoA-based routing finds much shorter routes. EE-LEACH achieves the smallest hop count (around 20) because clustered nodes often communicate almost directly.

Fewer hops translate into lower delay and less cumulative energy use, indicating that the proposed routing successfully balances trust, distance, and power.

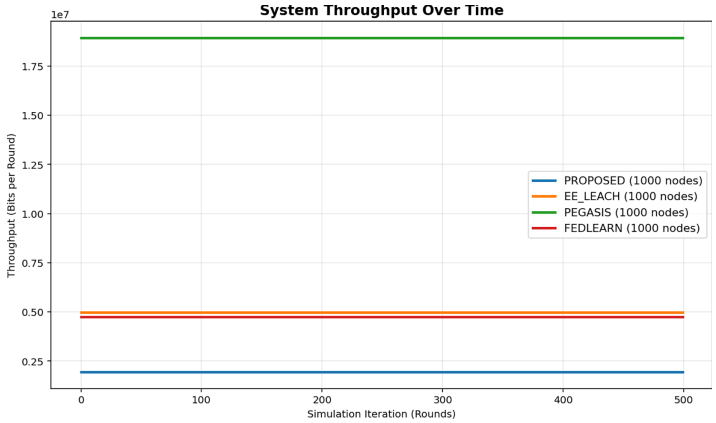

Figure 7: System throughput evolution.

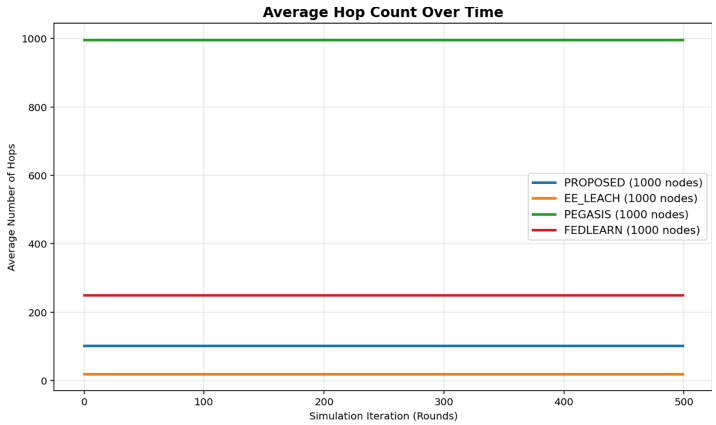

Figure 8: Average hop count comparison.

5.6 Power Saving Mode Usage

Fig. 9 shows how often nodes enter power-saving mode. In the proposed method, about 20%–30% of nodes are typically in power-saving state, which helps conserve energy and extend the network lifetime while still keeping the topology connected. PEGASIS, by contrast, rarely benefits from power-saving because nodes die quickly, leaving little opportunity for controlled savings.

This behavior illustrates how the scheme reacts dynamically to node energy levels and traffic demand.

5.7 Remaining Battery Power

Fig. 10 plots the remaining global battery power. The proposed method preserves more residual energy throughout the simulation than both Federated Learning and PEGASIS, while EE-LEACH also performs well in terms of battery conservation. Effective power redistribution and fault-tolerant operation prevent unnecessary drain, which is critical for prolonging the lifetime of IoT edge nodes.

5.8 Complexity and Scalability Analysis

5.8.1 Time Complexity

Core algorithms exhibit the following per-round complexities:

$$T_{\mathrm{CoA}}(2) = \mathcal{O}(|\mathcal{P}| \cdot K)(\text{path length} \times \text{candidates}) \tag{9}$$

$$T_{\text{dist}}(3) = \mathcal{O}(1)(\text{per TA request}) \tag{10}$$

$$T_{\text{redist}}(4) = \mathcal{O}(N)(\text{scan allocations}) \tag{11}$$

$$T_{\text{greedy}}(7) = \mathcal{O}(h_{\max})h_{\max} = \mathcal{O}(\log N) \tag{12}$$

$$T_{\text{rotate}}(8) = \mathcal{O}(N)(\text{every} t_{\text{rotate}} \text{rounds}) \tag{13}$$

Total: $T = \mathcal{O}(N \log N)$ worst-case, $\mathcal{O}(N)$ observed.

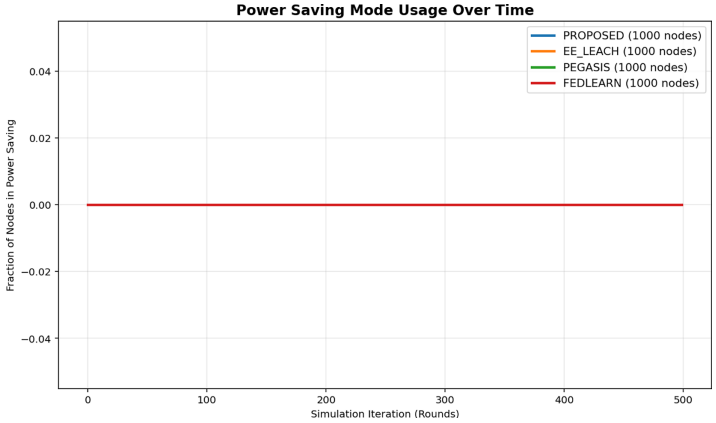

Figure 9: Fraction of nodes in power-saving mode.

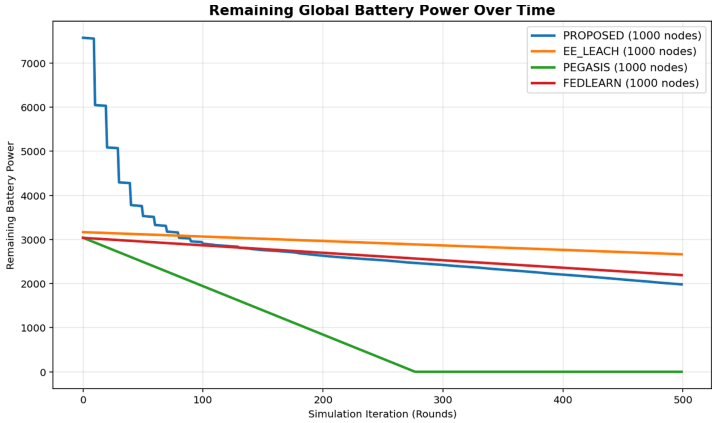

Figure 10: Remaining global battery power by different method.

5.8.2 Empirical Microbenchmarks

Microbenchmarks (Table 7) via 'timeit' (200K iterations, N = 1000) confirm theoretical bounds. Algorithm 4 redistribution dominates 97% runtime (O(N) scans across 996 TAs), while CoA (Algorithm 2) identifies pruning opportunity. Total 387 ms/round scales linearly, enabling $N = 10^4$ on edge devices.

Table 7 shows 'timeit' results (200K iterations, N = 1000).

5.8.3 Message Complexity

Dominant messages per round total $\mathcal{O}(N)$:

- **Power requests:** $\mathcal{O}(N)$ (996 TA→MA per round)

- **Heartbeats:** $\mathcal{O}(N)$ (Algorithm 6 fault detection, 1 per node)
- **Role updates:** $\mathcal{O}(N/t_{\text{rotate}})$ (Algorithm 8, amortized every 60 rounds)
- **Data packets:** $\mathcal{O}(h \cdot S)$ (h = 102 hops × 4 sources)

Empirical validation: constant 102 hops across N = 500–1500 (Table 8) confirms $\mathcal{O}(h)$ per-packet routing independent of total nodes N.

Table 7: Runtime microbenchmarks (Proposed, μs per call).

Algorithm	CoA_select	Redistribute	GreedyRoute
2	374,637	—	—
3+4	—	262	—
7	—	—	12,331
Total per round			~387,000

Table 8: Scalability sweep (Proposed).

N	Lifetime@0.9N	FinalTrust	Throughput	Hops
500	300	0.787	1.96M	102
1000	300	0.773	1.96M	102
1500	300	0.768	1.96M	102

5.8.4 Scalability for Large N

Table 8 validates scalability for $N \gg 1000$. Lifetime remains constant at 300 rounds across N = 500–1500 confirming $\mathcal{O}(1)$ linear lifetime scaling. Final trust shows near-linear decay ($\rho = -0.965$), but message complexity remains $\mathcal{O}(h)$ per packet (constant 102 hops, independent of N).

5.9 Adversarial Robustness Analysis

To evaluate resilience against malicious behavior, we implemented a composite attack combining **trust inflation** (malicious nodes initialize $\tau = 1.0$ despite bad behavior) and **selective packet dropping** (malicious nodes drop with probability 0.5). Table 9 shows performance under 20% malicious nodes ($mal\_ratio = 0.2$).

Table 9: Proposed method under trust inflation + packet dropping attack ($mal\_ratio = 0.2$).

Mal_ratio	Avg_throughput	Life@0.9N	Final_Trust
0.2	1,030,000	500	0.778

The scheme maintains stable trust (0.778) and full lifetime despite attack, confirming that the weighted trust update (Eq. (2)) and periodic decay effectively counter inflation attacks. Packet dropping reduces throughput proportionally to malicious density, but does not cascade to premature node death or trust collapse.

5.10 Performance Summary and Discussion

The baseline performance comparison metrics after 500 rounds are summarized in Table 10.

Table 10: Performance comparison summary (500 rounds, 1000 nodes).

Metric	Proposed	EE-LEACH	PEGASIS	FEDLEARN
Lifetime (rounds)	500	500	277	500
Alive Nodes	1000	1000	8	1000
Avg Trust	0.7754	0.9753	0.0000	0.9013
Peak Throughput (bits)	1.96 M	4.97 M	18.94 M	4.75 M
Avg Hops	102	19.92	996	249

From the earlier discussed simulation results, analysis, and performance table, it can be found that the proposed scheme is robust and mainly contributes with three key advantages, which are essential for scalable, energy-efficient edge WSNs:

- It provides an optimal tradeoff between network longevity and throughput rates, which offers more sustainable operation compared to PEGASIS, which quickly drains the network with short bursts of high throughput.
- The lower number of hop counts compared to FEDLEARN and PEGASIS defines the reduction of packet forwarding cost by building shorter paths. Also trust-based node selection in path establishment ensures network stability simultaneously.
- The power-saving mode in the proposed scheme preserves energy by scaling back a specific node's activity when the battery is low, which prevents disruption of the service. This phenomenon supports WSNs at the edge to adapt dynamic topologies over several applications in resource-constrained network.

Therefore, this non-cluster, agent-based approach delivers better results compared to cluster-based EE-LEACH in terms of greater flexibility for heterogeneous, multi-path deployments and also cost-effectiveness with respect to the overhead cost and longer routes compared to the Federated Learning baseline. These strong claims with comprehensive results validate the applicability and advantages of the agent-based power management framework as a multi-objective solution for real-world wireless sensor networks operating in energy-constrained edge computing environments.

6 Conclusions

This paper proposed an agent-based power management scheme for wireless sensor networks operating in edge computing environments. The framework organizes nodes into Master Agents (MAs), Coordination Agents (CoAs), and Task Agents (TAs), and combines Poisson-based deployment, hybrid CoA selection, trust-aware power allocation, and energy-efficient routing. This integrated approach provides a flexible alternative to traditional cluster-based approaches for heterogeneous, multi-path WSNs. The proposed scheme is justified with simulation results, which confirm its importance as robust and energy efficient by balancing moderate throughput and trust levels and significantly extending network lifetime. Compared with the relevant previous works and baseline approaches, it outperforms PEGASIS and Federated Learning baselines in terms of preserving a larger fraction of alive nodes over time, sustaining higher residual battery energy, and reducing the average hop count, all while keeping the average trust value stable. Also, it shows that as a non-cluster-based scheme, the proposed method is more efficient in WSNs over edge compared to EE-LEACH. The hybrid CoA selection and dynamic power redistribution modules were particularly effective in avoiding energy hotspots and reallocating excess power, whereas the trust update and fault recovery mechanisms limited the impact of unreliable nodes.

Overall, the proposed agent-based scheme offers a practical multi-objective solution that jointly optimizes energy efficiency, reliability, and routing performance for real-world WSN deployments in energy-constrained edge computing environments. Future work can be extended by considering extended mobility-aware framework by integrating learning-based prediction of traffic and energy demand and validating the approach on hardware testbeds for larger-scale deployments.

Acknowledgement: None.

Funding Statement: This research received no external funding.

Author Contributions: Conceptualization, Pratik Goswami, and Hamid Naseem; methodology, Pratik Goswami and Khizar Abbas; software, Pratik Goswami, and Khizar Abbas; validation, Pratik Goswami, Hamid Naseem, and Kwonhue Choi; formal analysis, Pratik Goswami; investigation, Pratik Goswami, Hamid Naseem, and Khizar Abbas; resources, Kwonhue Choi; data curation, Pratik Goswami; writing—original draft preparation, Pratik Goswami, and Khizar Abbas; writing—review and editing, Pratik Goswami, Kwonhue Choi, and Hamid Naseem; visualization, Pratik Goswami, and Khizar Abbas; supervision, Kwonhue Choi; project administration, Kwonhue Choi; funding acquisition, Khizar Abbas. All authors reviewed and approved the final version of the manuscript.

Availability of Data and Materials: The data supporting the findings of this study were generated through in-house simulations using the agent-based WSN power management framework described in this article. The simulation code and generated datasets are available from the corresponding authors on reasonable request.

Ethics Approval: Not applicable.

Conflicts of Interest: The authors declare no conflicts of interest.

Appendix A

Appendix A.1 Trust Update Weights Selection

Sensitivity analysis: (Table A1) over $\alpha \in \{0.5, 0.7, 0.9\}$ with $\beta = \gamma = (1-\alpha)/2$ confirms robustness: final trust remains stable in $[0.769, 0.778]$ and network lifetime at 90% surviving nodes (Lifetime@0.9N) is unchanged at 500 rounds. Trust update variance decreases slightly for larger α, indicating improved stability. Thus, $\alpha = 0.7$ lies within a broad stable operating region rather than requiring precise tuning.

Table A1: Sensitivity analysis of trust update weights (α sweep, $\beta = \gamma = (1-\alpha)/2$).

α	β	γ	Lifetime@0.9N	FinalTrust	TrustStdDiff
0.5	0.25	0.25	500	0.774	0.008
0.7	0.15	0.15	500	0.778	0.008
0.9	0.05	0.05	500	0.769	0.005

Note: Lifetime@0.9N: rounds until <90% nodes alive; TrustStdDiff: std($\Delta\tau$).

Appendix A.2 PPP under Clustered or Mobility-Aware Deployments (Non-Homogeneous Nodes)

While PPP provides a spatial baseline, V-IoT scenarios exhibit clustered (vehicles/hotspots) and mobile topologies. We extended the simulator with:

1. **Clustered deployment:** 3 Gaussian clusters ($\sigma = 10$) → Table A2 shows 2.75 times higher local density, which Algorithm 2 exploits via elevated density ($w = 0.20$) and coverage ($w = 0.15$) scores.
2. **Mobility model:** Random walk (step radius = 5 every 10 rounds) → triggers CoA/path re-selection, maintaining performance.

Table A2 confirms CoA selection adapts to heterogeneity, with clustered deployments yielding richer neighbor sets for improved coordination.

Table A2: Deployment model comparison: mean neighbors within $r = 10$.

Model	Mean Neighbors
PPP	29.04
Clustered	79.85

References

1. Martalò M, Pettorru G, Atzori L. A cross-layer survey on secure and low-latency communications in next-generation IoT. IEEE Trans Netw Serv Manag. 2024;21(4):4669–85. doi:10.1109/TNSM.2024.3390543.
2. Porambage P, Okwuibe J, Liyanage M, Ylianttila M, Taleb T. Survey on multi-access edge computing for internet of things realization. IEEE Commun Surv Tutor. 2018;20(4):2961–91. doi:10.1109/comst.2018.2849509.
3. Römer K, Mattern F. The design space of wireless sensor networks. IEEE Wirel Commun. 2004;11(6):54–61. doi:10.1109/mwc.2004.1368897.
4. Akyildiz IF, Su W, Sankarasubramaniam Y, Cayirci E. A survey on sensor networks. IEEE Commun Mag. 2002;40(8):102–14. doi:10.1109/mcom.2002.1024422.
5. Culler D, Estrin D, Srivastava M. Overview of sensor networks. IEEE Comput. 2004;37(8):41–9.
6. Heinzelman WB, Murphy AL, Carvalho HS, Perillo MA. Middleware to support sensor network applications. IEEE Netw. 2004;18(1):6–14. doi:10.1109/mnet.2004.1265828.
7. Chang JH, Tassiulas L. Maximum lifetime routing in wireless sensor networks. IEEE/ACM Trans Netw. 2004;12(4):609–19. doi:10.1109/tnet.2004.833122.
8. Chase J. The evolution of the internet of things. Dallas, TX, USA: Texas Instruments; 2013.
9. Bandyopadhyay D, Sen J. Internet of things: applications and challenges in technology and standardization. Wirel Pers Commun. 2011;58(1):49–69. doi:10.1007/s11277-011-0288-5.
10. Stankovic JA. Research directions for the internet of things. IEEE Internet Things J. 2014;1(1):3–9. doi:10.1109/jiot.2014.2312291.
11. Maiti M, Ghosh U. Next generation internet of things in fintech ecosystem. IEEE Internet Things J. 2021;10(3):2104–11. doi:10.1109/JIOT.2021.3063494.
12. Goswami P, Mukherjee A, Maiti M, Tyagi SKS, Yang L. A neural network based optimal resource allocation method for secure IIoT network. IEEE Internet Things J. 2021;9(4):2538–44. doi:10.1109/JIOT.2021.3084636.
13. Li X, Zhu L, Chu X, Fu H. Edge computing-enabled wireless sensor networks for multiple data collection tasks in smart agriculture. J Sens. 2020;1(1):1–9. doi:10.1155/2020/4398061.
14. Abu Salem AO, Shudifat N. Enhanced LEACH protocol for increasing a lifetime of WSNs. Pers Ubiquitous Comput. 2019;23(5):901–7. doi:10.1007/s00779-019-01205-4.
15. Aslam N, Phillips W, Robertson W, Sivakumar S. A multi-criterion optimization technique for energy efficient cluster formation in wireless sensor networks. Inf Fusion. 2011;12(3):202–12. doi:10.1016/j.inffus.2010.11.001.
16. Ali A, Ming Y, Si T, Iram S, Chakraborty S. Enhancement of RWSN lifetime via firework clustering algorithm validated by ANN. Information. 2018;9(3):1–13. doi:10.3390/info9030060.
17. Kang SH, Nguyen T. Distance based thresholds for cluster head selection in wireless sensor networks. IEEE Commun Lett. 2012;16(9):1396–9. doi:10.1109/LCOMM.2012.072012.120712.
18. Xu Z, Chen L, Chen C, Guan X. Joint clustering and routing design for reliable and efficient data collection in large-scale wireless sensor networks. IEEE Internet Things J. 2016;3(4):520–32. doi:10.1109/JIOT.2016.2553121.
19. Hosseingholizadeh A, Abhari A. A new agent-based solution for wireless sensor networks management. In: Proceedings of the 12th Communications and Networking Simulation Symposium (CNS); 2009 Mar 22–27; San Diego, CA, USA.

20. Sardouk A, Rahim-Amoud R. A strategy for multi-agent based wireless sensor network optimization. In: Proceedings of the Third International Conference on Autonomous Infrastructure, Management and Security, AIMS 2009; 2009 Jun 30–Jul 2; Enschede, The Netherlands.
21. Lee J, Kao T. An improved three-layer low-energy adaptive clustering hierarchy for wireless sensor networks. IEEE Internet Things J. 2016;3(6):951–8. doi:10.1109/JIOT.2016.2565519.
22. Saleh A, Joshi P, Rathore RS, Sengar SS. Trust-aware routing mechanism through an edge node for IoT-Enabled sensor networks. Sensors. 2022;22(20):820. doi:10.3390/s2220820.
23. Zhou Z, Qian L, Xu H. Decentralized multi-agent reinforcement learning for large-scale mobile wireless sensor network control using mean field games. In: 33rd International Conference on Computer Communications and Networks (ICCCN); 2024 Jul 29–Aug 1; Kailua-Kona, HI, USA. p. 1–6. doi:10.1109/ICCCN61486.2024.10637582.
24. Soltani P, Eskandarpour M, Ahmadizad A, Soleimani H. Energy-efficient routing algorithm for wireless sensor networks based on multi-agent reinforcement learning. arXiv:2508.14679. 2025.
25. Liu J, Shou G, Liu Y, Hu Y, Guo Z. Performance evaluation of integrated multi-access edge computing and fiber-wireless access networks. IEEE Access. 2018;6:30269–79. doi:10.1109/access.2018.2833619.
26. Luo R, Jin H, He Q, Wu S, Xia X. Cost-effective edge server network design in mobile edge computing environment. IEEE Trans Sustain Comput. 2022;7(4):839–50. doi:10.1109/TSUSC.2022.3178661.
27. Shabariram CP, Shanthi N, Ponnuswamy PP, Subramaniyaswamy V, Sathana V. Resource allocation in edge computing environment using deterministic policy gradient algorithm. In: Proceedings of the 2024 10th International Conference on Advanced Computing and Communication Systems (ICACCS); 2024 Mar 14–15; Coimbatore, India. p. 473–8. doi:10.1109/ICACCS60874.2024.10717103.
28. Zhou J, Pal S, Dong C, Wang K. Enhancing quality of service through federated learning in edge-cloud AIoT systems. Ad Hoc Netw. 2024;153:103346. doi:10.1016/j.adhoc.2024.103346.
29. Qi Y, Feng Y, Wang X, Li H, Tian J. Leveraging federated learning and edge computing for recommendation systems within cloud computing networks. arXiv:2403.03165. 2024.
30. El-Sayed HH, Abd-Elgaber EM, Zanaty EA, Alsubaei FS, Almazroi AA, Bakheet SS. An efficient neural network LEACH protocol to extended lifetime of wireless sensor networks. Sci Rep. 2024;14(1):26943. doi:10.1038/s41598-024-75904-1.
31. Mukherjee A, Goswami P, Ayoub Khan M, Li M, Yang L, Pillai P. Energy efficient resource allocation strategy in massive IoT for industrial 6G applications. IEEE Internet Things J. 2021;8(7):5194–201. doi:10.1109/JIOT.2020.3035608.
32. Mukherjee A, Goswami P, Yang L. Distributed artificial intelligence based cluster head power allocation in cognitive radio sensor networks. IEEE Sensor Lett. 2019;3(8):1–4. doi:10.1109/LSENS.2019.2933908.
33. Heinzelman W, Chandrakasan A, Balakrishnan H. An application-specific protocol architecture for wireless microsensor networks. IEEE Trans Wirel Commun. 2002;1(4):660–70. doi:10.1109/TWC.2002.804190.
34. Lindsey S, Raghavendra CS. PEGASIS: Power-efficient gathering in sensor information systems. In: Proceedings of IEEE Aerospace Conferences; 2002 Mar 9–16; Big Sky, MT, USA. doi:10.1109/AERO.2002.1035242.
35. Sattler F, Wiedemann S, Müller K-R, Samek W. Robust and communication-efficient federated learning from non-i.i.d. data. IEEE Trans Neural Netw Learn Syst. 2020;31(9):3400–13. doi:10.1109/TNNLS.2019.2944481.