ENIAO: ENERGY AWARE FAULTY NODE RE-PLACEMENT INTEGRATED WITH DUTY CYCLING AND IMPROVED HARMONY SEARCH BASED CLUSTERING THROUGH ADAPTIVE FISH SCHOOL SEARCH ROUTING FOR WSN OPTIMIZATION

RM.Alamelu[1*], J.Naveen Ananda Kumar [2], C.Jayapratha[3], Govindaprabhu GB[4]

[1]Department of Computer Science, Sri sarada Niketan College for Women, Amaravathi pudur, Karaikudi. Tamilnadu, India,
[2]Data Engineer, Tekinvaderz LLC, Florida, USA
[3]Department of Computer Science and Engineering, Karpaga Vinayaga College of Engineering & Technology, Madhuranthagam, Tamilnadu, India.
[4] MKU23PFOS10909, Madurai Kamaraj University (MKU), Madurai, Tamilnadu, India.

ABSTRACT

In wireless sensor networks (WSNs), sensor nodes are constrained by resource constraints. The limited energy supply and susceptibility to failure greatly affect WSN lifespan, hindering long-term deployment. ENIAO is an Integrated cross-layer Optimized Routing Approach for WSNs that is fault-tolerant and energy-efficient. A bio-inspired clustering architecture and adaptive duty cycling are incorporated into ENIAO's routing optimization. In a clustering protocol, the network is partitioned, and paths are dynamically optimized within and between clusters. By optimizing active/sleep schedules, duty cycling optimizes energy efficiency. A variety of network conditions have been simulated to assess ENIAO's performance. Regarding fault tolerance and energy consumption, ENIAO significantly prolongs the network lifetime. As compared to benchmark protocols, it achieves higher throughput. As a result of the cross-layer design, ENIAO is automatically adapted to optimize energy usage and routing reliability. In the long run, large-scale IoT deployments are possible with ENIAO due to the integrated approach.

KEYWORDS

ENIAO, WSN, Fish School Search Routing, Duty Cycling, Improved Harmony Search based Clustering, Energy efficient, WSN Optimization.

1. INTRODUCTION

The wireless sensor network is an important technology for monitoring the environment, controlling traffic, and monitoring health. A WSN consists of spatially distributed autonomous sensors monitoring temperature, motion, and sound conditions [1]. A severe resource constraint is hindering the widespread deployment of WSNs, especially limited energy availability and high sensor failure rates [2].

DOI: 10.5121/ijcnc.2024.16506

A sensor node is typically powered by a battery with limited capacity. Batteries cannot be replaced frequently for networks with hundreds or thousands of nodes. In order to maximize network lifetime [3], the time until the first node fails due to battery depletion needs to be maximized. Energy consumption in WSNs is dominated by communication activities like data transmission and reception. Furthermore, sensors fail prematurely because of hardware faults, software errors, and communication links. A node failure can lead to network segmentation and lost sensing coverage. A practical deployment of WSNs requires energy-efficient fault tolerance techniques [4]. A variety of approaches have been proposed to improve energy efficiency and fault tolerance in WSNs [5]. During idle times, duty cycling allows nodes to sleep, which reduces energy waste. In network process, it reduces communication costs with data aggregation [6]. For recovering from failures, redeployment and replication are investigated [7]. The majority of techniques do not integrate energy efficiency and fault tolerance across layers.

This paper proposes an energy-aware faulty node replacement technique based on duty cycling and dual clustering and routing optimizations. Through efficient fault handling, this work develops a holistic approach that maximizes energy usage across all layers. A duty cycling and dual-stage routing optimization method maximizes network lifetime through energy-aware node replacement.

In real-world scenarios the ENIAO's innovative design offers diverse applications. By monitoring soil and climate conditions, it allows precision agriculture to optimize crop yield and manage energy efficiently. The ENIAO platform facilitates distributed sensor networks for environmental monitoring, such as tracking pollution and detecting wildfires. A joint optimization approach allows the system to monitor infrastructure, traffic, and utilities efficiently in smart cities. ENIAO's capabilities allow industrial IoT to optimize processes through reliable and energy-efficient data collection and actuators in factories and warehouses. In disaster response situations, ENIAO's design is especially beneficial since it enables rapid deployment of sensor networks under severe constraints. With these applications, ENIAO demonstrates its potential to improve the efficiency and reliability of wireless sensor networks in various sectors.

Novelties and contributions are introduced to Wireless Sensor Networks with this approach. The paper presents a cluster-based fault detection and replacement method that minimizes the energy overhead associated with fault detection. ENIAO uses a dual-optimization strategy using weighted clustering and Adaptive Fish School Search, which reduces data transmission energy costs significantly. A dynamic sleep interval adjustment mechanism allows nodes to adjust their sleep intervals according to network conditions and application requirements. With ENIAO, fault tolerance, duty cycling, clustering, and routing are all considered jointly, ensuring they work in coordination rather than separately. Simulations have demonstrated the effectiveness of ENIAO in enhancing network lifetime and reliability compared with other state-of-the-art methods, indicating its potential for WSN optimization.

The key contributions have been made, this proposed method uses clustering to detect faults and replace nodes in an energy-efficient manner. In this approach, clustering is performed using Improved Harmony Search, the routing is carried out using Fish School Search. The optimization of duty cycles based on the current conditions in the network is adaptive. An approach that combines fault tolerance, duty cycling, clustering, and routing at the cross-layer level. During the research, ENIAO, a cross-layer framework that optimizes energy efficiency and fault tolerance autonomously, was developed. In order for WSNs to operate long-term, this balance must be maintained. A reliable and long-lasting sensing system across large areas can be achieved with this method. Using its algorithmic mechanisms, a wide range of monitoring scenarios can be implemented.

This paper is organized as follows: Section 2 provides an overview of related work on fault tolerance and energy efficiency in WSNs. A detailed description of the architecture, models, and algorithms of the proposed ENIAO approach is presented in Section 3. A comprehensive simulation is performed in Section 4 to evaluate ENIAO's performance in comparison to other state-of-the-art methods. The paper concludes with Section 5 which discusses possible future extensions.

2. LITERATURE REVIEW

Various coverage optimization protocols were discussed by Avinash et.al [8]. A clustering protocol and a distributed protocol can be broadly classified as clustering protocols. These protocols can also be classified based on the type of sensing model used, node location information, and the mechanism used to determine neighbouring nodes.

Ali Forghani Elah Abadi et.al [9] presented an energy-efficient control and routing protocol for wireless sensor networks. Reinforcement learning is used to manage energy in the network. By using reinforcement learning, this protocol optimizes routing policies to maximize long-term rewards for each node. Wireless sensor networks can be improved through three energy management approaches. Reduction in route length and energy consumption can be achieved by using reinforcement learning. A second approach is to improve node energy consumption by utilizing sleep scheduling. Data transmission is restricted based on the change rate of received data. A fault node recovery (FNR) algorithm is proposed by Chaitrali Brahme et.al [10] to enhance the lifetime of a wireless sensor network. When the sensor nodes have run out of battery power or reached their operational limit, they shut down. Network failures must be detected in advance and appropriate measures taken to sustain network operation. The algorithm combines grade diffusion and genetic algorithms. Sensor nodes can be replaced fewer times and routing paths can be reused.

An improved method based on deep reinforcement learning (DRDC) has been proposed by Razieh Mohammadi et.al [11]. To avoid emergency packet loss and unnecessary frequent sleep/wake, DRDC considers the change rate of data sensed by BN in addition to its energy. In order to accurately determine BN's duty cycle, Deep Q-Network (DQN) is combined with light neural networks. With limited EH-BN resources, a three-layered communication architecture is used to preserve memory constraints and computational power. EH-BN receives only the trained policy, which is executed on a local server. To realize the DQN algorithm's optimal performance, (4) it design a reward function. A hybrid meta-heuristic approach using particle swarm optimization and gravitational search algorithms was presented by Chen et al. [12], aimed at optimizing coverage, connectivity, and energy efficiency in wireless sensor networks. It does not explicitly address reliability factors such as fault tolerance.

Using EH methods, energy storage technologies, and EH system architectures, Williams et.al [13] surveyed the current state of EH technology for small-scale WSNs. The work combines methods and storage, including multi-source and multi-storage architectures, as well as other optimizations. The novel cluster-based routing method presented by Nageswararao Malisetti [14] maximizes the network lifetime by making routing progress more effective. It consists of two phases: selecting the optimal cluster head using the new Moth Levy Artificial Electric Field Algorithm (ML-AEFA), and transmitting the data using the new Customized Grey Wolf Optimization (CGWO). In this case, the optimal CH is selected based on energy, node degree, distance between sensor nodes, distance between CH and Base Station (BS), and time of death. The performance of the implemented method is compared with existing schemes. L. Van Hoesel [15] presented a cross-layer approach for wireless sensor networks. Compared to ad hoc wireless networks, WLANs use energy-efficient networking protocols. In dynamic network topologies, nodes should be able to last several years on a single battery. To achieve highly energy-efficient WSNs, an integrated set of networking protocols is a good solution. Combining medium access and routing is our approach.

According to Yu et.al [16], faulty data is detected and discarded locally, thereby minimizing network resource consumption and reducing the terminal processing burden. According to simulation results, the proposed algorithm improves fault detection accuracy. A WSN fault detection and identification approach was developed by Mariachi et.al [17]. It is essential to identify and classify data and system fault types to perform accurate recovery actions. HMMs capture the fault-free dynamics in an environment and the dynamics of faulty data using our method. This HMM is then structurally analyzed to determine the type of data faults and system faults.

Since the 1970s, WSN research has advanced considerably to improve energy efficiency and fault tolerance. Initially, duty cycling, data reduction, and energy-aware routing were investigated to reduce energy consumption. For handling failures reactively, node replication [11], spare deployment [12], and mobility-based recovery [13] were investigated. Recently, some researchers have looked at combining duty cycle and routing [14], or utilizing mobility to optimize energy consumption [15].

I.Abdoulaye et.al [20] propose combining clustering principles with data prediction for smart cluster-head selection. By electing an effective CH from among the cluster nodes, SDPM reduces transmission and conserves energy by predicting data for the cluster nodes. SDPM reduces energy consumption significantly, it potential for real-world WSNs to achieve better energy management and longer network lifetimes. R.Jia et.al [21] designed energy-efficient coverage methods for WSN nodes, focusing on improving energy efficiency and data transmission reliability. An energy-saving node coverage model is based on hierarchical and flat routing protocols. Meanwhile, the study explored an energy-efficient coverage method based on the improved gray wolf algorithm. It optimizes the deployment of sensor nodes and enhances their effectiveness. Results show that the algorithm performs significantly in network coverage optimization and achieves 100% coverage. With the 30-dimensional condition, the improved gray wolf algorithm shows excellent average performance.

The majority of techniques, however, still deal with problems separately. The combination of fault tolerance, routing, duty cycling, and other mechanisms is rare. According to the literature review, several research gaps have been identified. It is difficult to combine energy efficiency and fault tolerance, and cross-layer solutions to unify multiple wireless sensor networks (WSNs) are lacking. The holistic optimization of WSNs does not consider Interdependencies between different aspects. Techniques that adapt to changing network conditions are also lacking. Additionally, a comprehensive solution is lacking that addresses both energy efficiency and reliability. It is necessary to compare proposed methods with state-of-the-art techniques in more extensive evaluations. With ENIAO, energy efficiency, and fault tolerance are optimized in a coordinated manner across multiple layers. This is a novel approach to joint energy-reliability improvements using clustering, routing, and spanning. With ENIAO, dynamic duty cycling is combined with clustering-based energy-aware node replacement.

3. PROPOSED SYSTEM

In this work, System nodes are grouped into clusters in a hierarchical cluster architecture. The cluster head aggregates data from member nodes and communicates with the base station. Among the three key components of the ENIAO architecture are the sensor nodes that collect data and serve as sensing, cluster heads, or cluster members; the cluster heads, which manage the member nodes and route data to a base station; and the base station. In this workflow, nodes gather neighbour information and adjust communication range. Using improved harmony search optimization, weighted clusters are formed and cluster heads selected. A cluster head detects faulty nodes based on missed keep - alive messages and replaces them efficiently. With fish school search, cluster heads and base stations are routed. Routing strategy adjusts over rounds according to parameters such as energy levels based on duty cycling. A cluster head aggregates and routes data to a base station over an optimized topology from nodes. A repetitive cycle of clustering, routing, and duty cycling ensures the network's longevity and reliability by maintaining an energy-efficient and fault-tolerant topology. Figure 1 shows the flow of the proposed work phases.

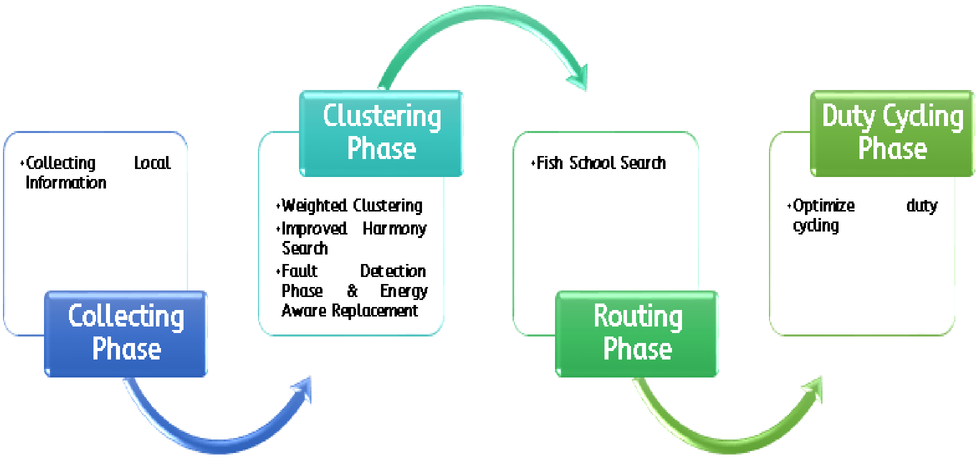

Figure 1. Flow of the Proposed System

3.1. Collecting Phase: Collecting Local Information

Each node collects information about its neighbour in this phase. Each sensor node can adjust its power level to adapt to a certain communication distance, and each node at its highest power level can communicate with another. As communication distance increases, energy is dissipated from sending messages. In ENIAO, there is a heterogeneous network of sensor nodes with various capabilities and levels of energy at the beginning. It is possible to adjust nodes' roles (sensing, routing, etc.) and transmission power. The flexibility allows for optimized energy efficiency and network reliability. To conserve power, low-energy nodes can cycle between sleep and active states, while high-power nodes handle repetitive tasks. Each node collects local neighbour information, adapting its range as necessary. By adapting to dynamic node capabilities and constraints, this approach extends network lifetime and improves performance overall.

Input: Set of sensor nodes S = {s1, s2..., sn} , Set of initial node energies E0 = {e01, e02,..., e0n}, Network area A, Maximum transmission range Rmax, Duty cycling period T
Output: Network topology at time t, Gt(Vt,Et) , Node roles at time t, Xt
Algorithm: **Collecting Phase**

1. Initialize network
2. Deploy |S| sensor nodes randomly in the 2D area A
3. Assign initial energy e0i ~ U(Emin,Emax) to each node si
4. Set communication range ri = Rmax for all nodes
5. Node roles Xi = {sensing}
6. Select CH ⊆ S based on weight wi ~ f(ei)
7. Group member nodes si ∈ S - CH into clusters based on distance d(si, chj) < rcluster
8. Duty cycling schedule & Set duty cycle period T
9. Schedule each node si to sleep/wake up based on ei
10. Role assignment & Update roles Xi based on ei and available resources
11. Eligible roles = {sensing, routing, aggregation, dormant}
12. Adjust transmission range ri of nodes based on ei
13. Establish routes b/w CH and base station
14. Periodically update Gt and Et
15. Return Gt, Xt

End Algorithm

Algorithm 1. Collecting Phase

This algorithm (1) initializes a heterogeneous network model with random deployment, integrating clustering, duty cycling, and role assignment. Depending on resources and capabilities, sensor nodes perform sensing, routing, and aggregation. In a duty cycle mode, low-energy nodes alternate between sleeping and active states. Every sensor node is capable of adjusting its transmission power, thereby modifying its communication range. A node's get neighbors method allows it to communicate with all its immediate neighbors. Enhanced reliability and energy efficiency are achieved by optimizing node roles and communication ranges over time. This network's adaptive nature supports dynamic task allocation in response to individual nodes' capabilities and constraints, improving system longevity.

3.2. Clustering Phase

To balance energy consumption, a distributed algorithm groups sensor nodes according to proximity. It is the Cluster Heads (CHs) who aggregate and transmit data to the Sink Node. A node broadcasts its identifiers and energy levels, and a 1-hop neighbor is calculated based on the average energy. Those with above-average energy declare themselves as prospective CHs. Nodes send Join-Requests to neighboring CHs based on energy and distance criteria. Communication schedules are created by CHs and communicated to Sinks. In order to further optimize cluster formations, ENIAO incorporates a Harmony Search algorithm that takes weighted attributes into account.

3.2.1. Weighted Clustering

Many node attributes affect clustering objectives, including remaining energy, distance to CH, and node degree. As determined by sensitivity analysis, ENIAO normalizes the weights of each attribute. By calculating the weighted sums of the attribute values of the member nodes, an attribute vector is created for each cluster. A vector representation encapsulates cluster properties. A number of parameters such as coverage, connectivity, and lifetime are used to evaluate cluster fitness. A higher probability of selection is associated with clusters with better fitness attribute vectors.

Let *C* as Cluster consists of nodes *{i}*, then the attribute vector for cluster *C* is denoted as A_C. For a node *i*, the weighted sum S_i is given by,

$$S_i = \sum_k w_k A_{ik} \quad (1)$$

where A_{ik} is the k[th] attribute of node *i*, and w_k is the weight for the k[th] attribute. The attribute vector for a cluster *C*, A_C is the weighted sum of the attribute vectors of all member nodes in the cluster:

$$\mathbf{A}_C = \sum_{i \in C} \mathbf{W} \odot \mathbf{A}_i \quad (2)$$

The fitness F_C of a cluster C is evaluated based on its attribute vector A_C:

$$F_C = f(\mathbf{A}_C) \quad (3)$$

A function *f* represents the fitness evaluation criteria (e.g., coverage, connectivity, lifetime). A higher fitness cluster has a higher probability of being selected. Cluster *C* probability of selection *P(C)* is:

$$P(C) = \frac{F_C}{\sum_j F_{C_j}} \quad (4)$$

where the sum in the denominator is over all clusters C_j.

3.2.2. Improved Harmony Search

Optimizing clustering with Improved Harmony Search (HS) seeks balanced solutions to objectives like network lifetime, reliability, and energy efficiency. A harmony memory stores historical best solutions, and new clusters are generated by modifying these stored vectors, adding random variations controlled by a pitch adjustment rate parameter. Weighed multi-objective functions evaluate solutions, and superior solutions replace those in memory. To enhance exploration, randomization parameters introduce random clusters. Near-optimal cluster formation is achieved by repeating this process until a termination criterion is met. A combination of weighted attributes and optimization optimizes cluster performance by adapting to dynamic network conditions.

In the first phase, ENIAO divides the network into clusters, with one node chosen as cluster head (CH). This initial clustering lays the foundation for weighted cluster optimization. Steps to follow:

i. ***Advertisement Phase:*** Every eligible node broadcasts an advertisement message containing its identifier and current energy level to its single-hop neighbors. Identifier ID_i and energy level E_i are broadcast to single-hop neighbors by node *i*.

$$\text{Broadcast } (ID_i, E_i) \forall i \in N \quad (5)$$

ii. ***Cluster Set-up Phase:*** A node decides to join a cluster based on its signal strength after receiving advertisement messages. For each cluster, the *CH* is selected based on its signal strength/energy. As each node *i* receives advertisement messages from its neighbors, it joins the cluster with the strongest signal strength S_i.

$$.\ C_k = \{i \mid max(S_i)\} \quad (6)$$

The node with the highest energy E_i within each cluster C_k is selected as the CH:

$$CH_k = \arg\max_{i \in C_k}(E_i) \quad (7)$$

iii. ***Scheduling Phase:*** Member nodes are assigned timeslots by CH nodes using Time Division Multiple Access (TDMA). For intra-cluster communication, all cluster nodes receive this schedule.

$$\text{TDMA Schedule: } \{t_{ik}\}\forall i \in C_k \text{ and } \text{Broadcast }(\{t_{ik}\})\forall i \in C \quad (8)$$

iv. ***Steady-State Phase:*** During allocated timeslots, member nodes begin sending data to CHs. Before transmitting data to sink nodes, CHs aggregate and compress it.

$$\text{Aggregate and Compress: } \{d_{CH_k}\} = \sum d_{ik}(t_{ik} \quad (9)$$

Using the received signal strength, intermediate nodes with higher energy are selected as CHs in this initial clustering. A distributed protocol enables self-organization. Schedules prevent interference within clusters, improving reliability. Input for improving HS clustering optimization is this baseline formation. Load balancing requires periodic re-clustering as nodes deplete energy. After initial clustering, ENIAO optimizes cluster formations using weighted attributes and Harmony Search. Accordingly, we assign the following near-optimal clustering configuration:

i. ***Base Station Assignment:*** The sink node is the Base Station (BS), which collects all sensor data. The location is predetermined and known to all nodes.

ii. ***Cluster Head and Member Assignment:*** In Cluster Heads (CHs), nodes aggregate and transmit members' data. With the improved HS solution, CH assignments are optimal. Nodes periodically send keep alive messages to cluster heads. If there is no message within a timeout, the member is marked as potentially faulty. Using test requests, cluster head verifies suspected nodes. No response marks it as faulty. A faulty node is reported to the cluster head. The remaining nodes are assigned to their CHs. Each CH's optimal mapping is stored in HS memory.

iii. ***Role Assignment Notification and Setup:*** Each node receives its CH/member assignment from the BS. The nodes learn their roles and the identities of their CHs. A CH determines intra-cluster routes based on connectivity. When needed, multi-hop inter-cluster routing is configured to the BS. For balanced data collection and dissemination, ENIAO formalizes the cluster structure with optimal node roles and routes. Using this as a basis, it optimizes across layers. Figure 2 shows the Assigning of the Base Station.

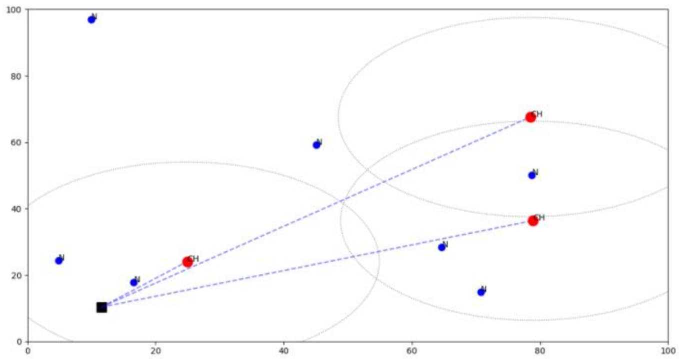

Figure 2. Base station Assignment

3.2.3. Fault Detection Phase & Energy Aware Replacement

Nodes in wireless sensor networks fail frequently because of hardware faults, battery degradation, and environmental interference. For reliability, local fault detection is essential. By monitoring missed keep-alive messages, cluster heads in ENIAO detect failures within their clusters. Specifically, cluster heads expect periodic keep-alive messages from their members (M). The cluster head CH*i* updates a list of suspected faulty nodes if no message is received from member M_j:

$$FNi \leftarrow FNi \cup \{mj\} \tag{10}$$

Here, FNi is the faulty CHi node list. Mj receives a test request message from CHi: send_test_request(CHi, mj). When mj does not acknowledge within a timeout, CHi confirms it as a failed node by updating the list:

$$if\ !receive\_ACK(mj){:}\ FNi \leftarrow FNi \cup \{mj\} \tag{11}$$

With this distributed mechanism, failures are detected locally. Detected failures are reported to base stations once identified. It disseminates the faulty node information across the network. Keeping neighborhood lists and routing paths current prevents failures. By replacing faulty nodes, ENIAO maintains sensing coverage and connectivity. As a replacement, the node with the highest remaining energy is preferentially used. Nodes are ranked by residual energy by cluster heads and the highest energy node E_j is selected to replace the faulty node mf:

$$\begin{gathered} smax\_energy = arg\ max\ (Ej)\ //\ Highest\ energy\ member \\ send\_replace\_request\,(CHi, smax\_energy, mf) \end{gathered} \tag{12}$$

As a result, s_j may move to mf's location. As soon as s_j is deployed, it resumes sensing and transmitting data. By using high energy nodes as replacements, overhead is minimized. In the long run, the network can sustain more faults before partitioning.

Distributed fault detection and targeted energy-aware node replacement in ENIAO enhance reliability and fault tolerance without consuming significant energy. It reduces topology knowledge and mass redeployment by allowing localized failure handling. In ENIAO, cluster heads monitor keep-alive messages and test acknowledgements, and mark unresponsive nodes as faulty. Based on residual energy and proximity to the fault, the base station selects the top-ranked node as the new member of the cluster. Upon failure of a cluster head, a new election occurs among functional members. Through its self-healing capability, ENIAO is able to continuously detect and address faults, thereby optimizing resilience and lifetime. Let N be the total number of sensor nodes, BS be the base station (sink node), CH be the cluster head, M be the cluster members, and F be the set of faulty sensors. Throughout the network, the base station relays faulty node info as follows:

$$\text{If } F \neq \emptyset, BS \text{ broadcasts } F \text{ to all nodes} \tag{13}$$

As a result, Nodes update their data routing paths to avoid sending traffic through faulty nodes.

$$\text{For each node } i \in N, \text{ update routing paths to avoid nodes in } F \tag{14}$$

If a cluster head becomes faulty, an election is triggered to replace it. A faulty node is detected based on unacknowledged test requests and missed keep alive messages.

$$\text{If } CH_k \in F, \text{new } CH_k = \arg \max_{i \in C_k \setminus F} (E_i) \tag{15}$$

The cluster heads notify the base station about faulty nodes. Update routing paths to avoid sending traffic through faulty nodes. A new election is triggered when a cluster head fails.

$$CH_k \rightarrow BS\text{: Notify } M_i \in F \tag{16}$$

INPUT: N = set of all sensor nodes {n1, n2, ..., nn}, Ei = Residual energy of node n_i, Tkeep_alive = Keep-alive message interval, Ttimeout = Fault detection timeout, F = Set of detected faulty nodes
OUTPUT: F = Updated set of detected faulty nodes
Algorithm: FaultDetection(N, Tkeep_alive, Ttimeout)
1. for each Cluster Head ni do
2. Initialize FaultyNodeList Fi
3. while network active do
    i. For each cluster Cj:
    ii. The cluster head Cj monitors the member nodes Mj in its cluster.
    iii. It maintains the current set of member nodes Mj.
    iv. At periodic intervals of Tkeep_alive, it expects to receive a keep-alive message from each node ni in Mj.
    v. If a keep-alive is not received from a node ni within the interval:
        1. Cj sends a test request message to ni to verify the fault
        2. It starts a timer and waits up to the timeout period Ttimeout
    vi. If no acknowledgement (ACK) is received from ni before the timeout:
        1. The node ni is marked as suspected faulty
        2. Cj adds ni to the faulty node set Fj for that cluster: Fj ← Fj ∪ {ni}
    vii. If a keep-alive is received, ni is marked as functional
        1. The sets Mj and Fj are dynamically updated as nodes change state
    viii. Once detected, faulty nodes in Fj are reported to the base station
    ix. By locally tracking member states, faults can be identified quickly for localized handling
    x. END FOR
    xi. Wait Tkeep_alive
    xii. for each Member nj do
        1. if no keep-alive from nj then
            a. SendTestRequest(ni, nj)
            b. Wait Ttimeout
        2. if no ACK from nj then
            a. Fi ← Fi ∪ {nj}
    xiii. EndFor
4. EndWhile
5. NotifyBaseStation(Fi)
6. EndFor

End Algorithm
PROCEDURE EnergyAwareReplacement(N, F, E)
1. for each faulty node nf ∈ F do
    a. Rank survivors N\F by energy Ei
    b. ns = HighestEnergyNode(N\F)
    c. RequestReplace(ns, nf)
    d. if Approved then
        i. UpdateRoutes(ns, nf)
    e. EndIf
2. EndFor

END PROCEDURE

Algorithm 2. Fault Detection

3.3. Routing Phase

The routing phase optimizes data transmission paths from sensor nodes to the base station using a hybrid approach. Using Kruskal's algorithm, it combines Fish School Search (FSS) for node-to-cluster routing and minimum spanning trees (MST) for inter-cluster communication. This creates a routing backbone for data aggregation. Further, ENIAO uses adaptive duty cycling, which puts nodes to sleep if they reach an energy threshold. Duty cycling ratios are continuously optimized to meet changing network requirements. Route optimization and adaptive duty cycling together significantly reduce energy consumption. To route data from cluster heads to base stations, inter-cluster routing paths need to be constructed. An energy-efficient routing topology is optimized using a bio-inspired Fish School Search (FSS) algorithm. It is based on fish schools' collective behavior [18], where fish exchange information to find food.

3.4.1. Fish School Search

As swimming fish expand and contract to find food, Fish School Search (FSS) uses a population-based metaheuristic algorithm. Using n-dimensional locations, each fish represents an optimization solution. There is a feature called weight for each solution that represents how successful the search has been [19].

A FSS is composed of feeding and movement operators: individual, collective-instinctive, and collective-volitive. Each fish in the school performs a random local search in the search space for promising regions. According to this equation, this component is computed:

$$x_i(t+1) = x_i(t) + rstep_{ind} \qquad (17)$$

This is represented by xi(t) as well as xi(t+1) as the position of a fish i before and after the individual component's movement. r ∈ R N with rj ∼ Uniform [−1, 1], for j = {1, . . ., n}. For this movement, step-ind is responsible for setting the maximum displacement. If f (x_i (t+1)) > f(xi(t)), then xi(t+1) is accepted as a new position. When there is no change in position, $x_i(t + 1) = x_i(t)$. Movements consist of a collective-instinctual component that averages all x_i. A vector I ∈ R N is calculated by multiplying the displacements of each xi by:

$$I = \frac{\sum_{i=1}^{S} \Delta x_i \Delta f_i}{\sum_{i=1}^{N} \Delta f_i} \qquad (18)$$

S is the school size, where Δx_i is a shorthand for $x_i(t + 1) - x_i(t)$, and Δf_i is a shorthand for $f(x_i(t + 1)) - f(x_i(t))$. Using the displacement, I, fish with a higher improvement will attract other fish to their position. Based on I, every fish moves as follows:

$$x_i(t+1) = x_i(t) + I \qquad (19)$$

During the search process, the collective-volitional component helps regulate school exploration/exploitation. In Eq. 1, each fish position xi is equal to the weight wi of the school, which is the bary center B ∈ R N. During the search process, collective-volitional components are used to regulate school exploration/exploitation abilities. A barycenter B ∈ R N is calculated based on fish positions xi and weights wi, as described in Eq. 4:

$$B(t) = \frac{\sum_{i=1}^{S} X_i(t) w_i(t)}{\sum_{i=1}^{N} w_i(t)} \qquad (20)$$

Fishes move towards B if the total school weight PS i=1 wi has increased from t to t + 1. If not, fishes are far from the bary center. Besides movement operators, feeding operators update weights according to:

$$w_i(t+1) = w_i(t) + \frac{\Delta f_i}{\max(|\Delta f_i|)} \qquad (21)$$

wi(t) is only allowed to vary from 1 up to wscale, which is a hyper-parameter. All weights are initialized with the value wscale/2.

To optimize routing paths between sensor nodes and cluster heads, Fish School Search (FSS) is employed. In the algorithm, each fish represents a potential routing solution, traversing the search space and exchanging position and path information. After a series of iterations, the most efficient routes are configured within clusters using FSS. To find energy-efficient transmission routes, this approach uses collective intelligence. Initial node positions and cluster head statuses of each fish are randomized, then refined by averaging neighboring solutions iteratively. It is determined which nodes are cluster heads by optimizing them and assuming 'cluster_head' roles. It maximizes network efficiency and adaptability by dynamically constructing optimized routing paths tailored to the current network state. Additionally, it optimizes node active/sleep schedules based on an adaptive duty cycling mechanism. It combines Fish School Search (FSS) with adaptive duty cycling for routing optimization. Using FSS, nodes, cluster heads, and base stations are connected efficiently. Low-energy nodes can sleep periodically by using the duty cycling mechanism. The integrated approach enhances energy efficiency and routing reliability by optimizing both network topology and node schedules.

```
Input: G(V,E) = Sensor network graph, C = Set of clusters, B = Base station node
Output: best_paths = Optimized paths from nodes to clusters
Algorithm: FSS_ROUTING (G, C, B)
  1. Initialize FSS population
     FSS = {}
  2. for i=1 to POPULATION_SIZE do
       a. f = Fish ()
       b. f.position = RANDOM_NODE(V) //Random node as fish position
       c. FSS.add(f)
  3. end for
  4. While not terminated do
       a. for each fish f in FSS do
            i. r = RANDOM_VECTOR ()
            ii. f.next_position = f.position + r
            iii. f.next_position += INDIVIDUAL_MOVEMENT(FSS)
            iv. f.next_position += COLLECTIVE_MOVEMENT(FSS)
            v. if f.EVALUATE(f.next_position) > f.EVALUATE(f.position) then
                 1. f.position = f.next_position
            vi. end if
       b. end for
       c. for each fish f in FSS do
            i. f.weight = UPDATE_WEIGHT(f)
       d. end for
  5. end while
  6. best_paths = SELECT_BEST_SOLUTIONS(FSS)
  7. Return best_paths
END Algorithm
```

Algorithm 3. FSS_ROUTING

Based on MST topology with the base station as the root node, the system facilitates energy-efficient routing. Kruskal's algorithm constructs this MST by adding edges between nodes iteratively. In ascending order of cost (the distance between nodes), edges are selected to connect each cluster head into a single MST with the base station at the center. As a result, the sensor data aggregation topology is optimized to ensure minimal edge costs. In order to avoid low-energy nodes and balance routing load, the MST is periodically reconstructed. This MST-based approach ensures efficient data transmission and network longevity by employing Fish School Search for routing paths from nodes to cluster heads. The minimum spanning tree topology centered at the base station can be used to route aggregated sensor data back to the base station for collection. Specifically, cluster heads use the tree to determine next hops when forwarding data. By analyzing its routing table, a cluster head will determine the next hop node on the path to the base station.

Consider a tree T= (V, E) where V is the set of nodes and E is the set of edges. Let A, B∈V where A is B's parent. Let BS ∈ V be the root of the tree Utilizing the minimum spanning tree (MST) topology, we minimize total transmission distance and overall energy consumption from cluster heads to base stations. By eliminating unnecessary hops, suboptimal paths are avoided. A logical topology is maintained over a physical network by periodically recompiling the MST using updated node energy levels as edge weights. Combining MST routing with fish school search for paths to cluster heads, our hybrid method ensures comprehensive energy optimization. Using residual energy, dynamically update the tree edges to avoid nodes with low energy.

While the minimum spanning tree (MST) optimizes routing, node energy levels fluctuate during operation, depleting batteries faster. Periodically, we reconfigure the MST based on current residual energy. When energy levels are low or after a set number of rounds, run Kruskal's algorithm again. As a result, the MST maximizes the minimum residual energy across paths. With dynamic routing, energy bottlenecks are prevented and the load is balanced, extending network lifespan. When a cluster head is low on energy, a different cluster head may be used, ensuring that data routing continues despite node failures.

3.4. Duty Cycling Phase: Optimize Duty Cycling

By periodically sleeping and waking nodes, duty cycling conserves energy. By keeping enough nodes active, an optimal duty cycle balances energy savings and latency. Every communication round, the approach adjusts active/sleeping ratios. A threshold of 17% is set based on the average energy of active nodes. A node below this threshold enters sleep mode, while a node above it remains active, maximizing energy use.

By implementing the active node ratio based on the network's energy distribution, low-energy nodes can sleep and recover during non-essential rounds. A threshold of 17% of the average energy ensures that sufficient active nodes are available for data forwarding. A key innovation is using dynamic average energy as a threshold for scheduling and adjusting duty cycles every round. As a result, idle listening is reduced and energy-constrained nodes can recharge. By setting the threshold based on current average energy, fragmentation and loss of coverage are prevented. Using this method, energy is conserved and latency is reduced, extending network life by minimizing energy waste and providing responsive sensor data.

Input: V: Set of sensor nodes
Output: Active Ratio: Ratio of active to sleeping nodes
Algorithm: DUTY_CYCLING

1. Get current active nodes
2. ActiveNodes = {}
3. for each node v in V do
    a. if v.status == ACTIVE then
        i. ActiveNodes.add(v)
    b. end if
4. end for
5. energy_sum = 0
6. for each node n in ActiveNodes do
    a. energy_sum += n.energy
7. end for
8. E_avg = energy_sum / ActiveNodes
9. Set energy threshold = 0.8 * E_avg
10. Update node statuses
11. for each node v in V do
    a. if v.energy < threshold then
        i. v.status = SLEEP
    b. else
        i. v.status = ACTIVE
    c. end if
12. end for
13. num_active = 0
14. for each node v in V do
    a. if v.status == ACTIVE then
        i. num_active += 1
    b. end if
15. end for
16. ActiveRatio = num_active / |V|
17. return ActiveRatio

END PROCEDURE

Algorithm 4. DUTY_CYCLING

In addition to optimizing routes initially, it dynamically adapts the routing strategy in response to changing network conditions and application needs to maintain performance and extend network lifetime. The adaptive routing involves periodically re-evaluating routing paths and duty cycling settings based on the current network state. Specific adjustments include recomputed Minimum Spanning Tree (MST) Edges: Using updated node energy levels, it recomputed MST edges. In the MST, E(v) represents the residual energy of node v. Increase their edge weights to avoid low-energy nodes, thus prioritizing higher-energy nodes. Node u and node v's edge weight w(u,v) is updated to reflect their inverse energy levels:

$$w(u,v) = \frac{1}{E(u) + E(v)} \qquad (22)$$

Cluster Head Elections: The current cluster head is elected if its energy ECH falls below a threshold TCH. It can be expressed as:

$$\text{If } E_{CH} < T_{CH}, \text{ trigger new cluster head election} \qquad (23)$$

Duty Cycling Adjustments: Based on overall network energy, it adjusts the duty cycling ratio. To reduce the active node ratio when E_{net} falls below T_{net}:

$$\text{If } E_{net} < T_{net}\text{, reduce active node ratio} \quad (24)$$

Increasing Node Density: Based on application demand, we adjust node density dynamically in critical areas. Let D(a) represent the density in area 'a'. The following are done to meet this density:

$$\text{If } D(a) \text{ is below requirement, increase node density in area } a \quad (25)$$

To improve resilience and balance load, rerouting paths around faulty or congested nodes and adjusting transmission power and communication rates. A periodic update of routing tables and duty cycling schedules is guided by application needs, network metrics, and node failures. By utilizing this dynamic approach, our sensor network can maximize performance and extend its lifetime in real-time. Figure 3 shows the Network Data transmission.

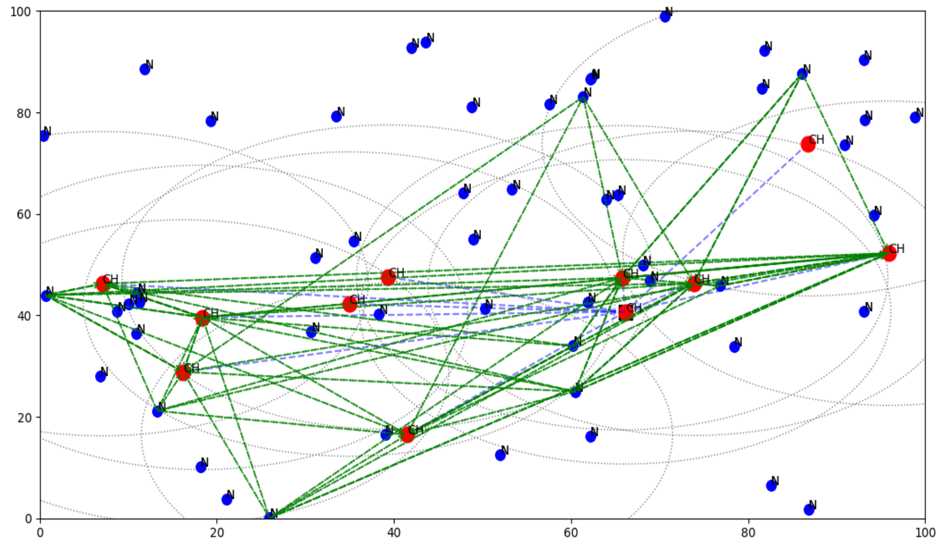

Figure 3. Network Data Transmission

4. Performance Results and Analysis

4.1. Simulation Settings

ENIAO's performance is determined through Python code simulation. Simulations are conducted with networks consisting of 10 to 50 random nodes placed in a 100m X1000m area. Each simulation runs for 60-130 seconds. Each scenario is simulated with a different number of sensor nodes. The sensor nodes are randomly selected through regulating simulation time. The simulation parameters are listed in Table 1.

Table 1. Simulation parameters.

Parameters	Values
Base Station (B_S)	2
Nodes (N)	10-50
Network Area (N_A)	100 X 100 m
Initial Energy (I_E)	1.5
Clusters (C)	5-10
Transmission Range	30 m
Communication Radius (R_C)	20 m
Node Placement	Algorithm Way Point

4.2. Energy Distribution

In Wireless Sensor Networks (WSNs), energy distribution refers to how energy is allocated and consumed across sensor nodes. It involves strategies for balancing energy consumption, so that no single node depletes its energy resources too quickly, resulting in network failures or performance degradation

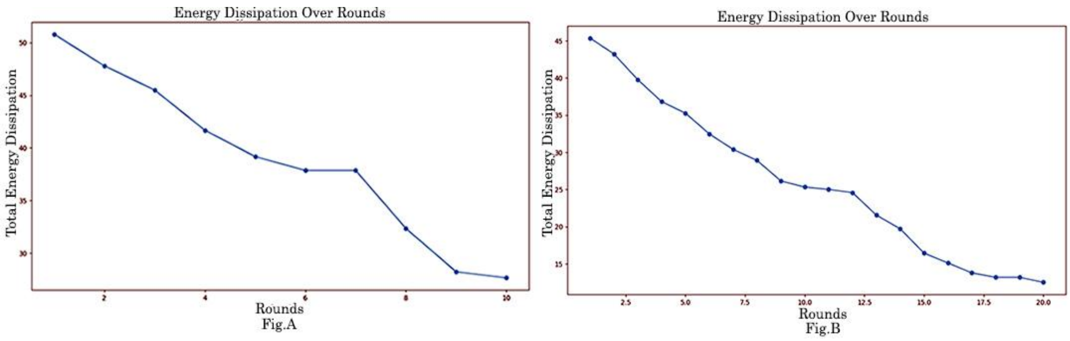

Figure 4. Energy Distribution

Figure 4 shows multiple rounds of energy dissipation. The charts show a consistent trend: the energy dissipation initially high gradually decreases as rounds progress. According to this progression, as resources deplete, the network will experience near-zero dissipation as it goes through its lifecycle from high energy use to optimization and finally to near-zero dissipation as it comes to the end of its lifespan.

4.3. Network Lifetime

During the lifetime of a WSN, the network remains functional and meets its operational requirements. The metric indicates how long the sensor nodes can effectively perform sensing, processing, and communication tasks before the energy runs out [22]. Figure 5 shows a wireless sensor network lifespan over rounds, revealing rapid declines followed by stability. Fig.A and Fig.B show stable network lifetimes for several initial rounds at approximately 0.5 units. In wireless sensor networks, predictive maintenance and energy management are crucial to preventing such abrupt performance degradation.

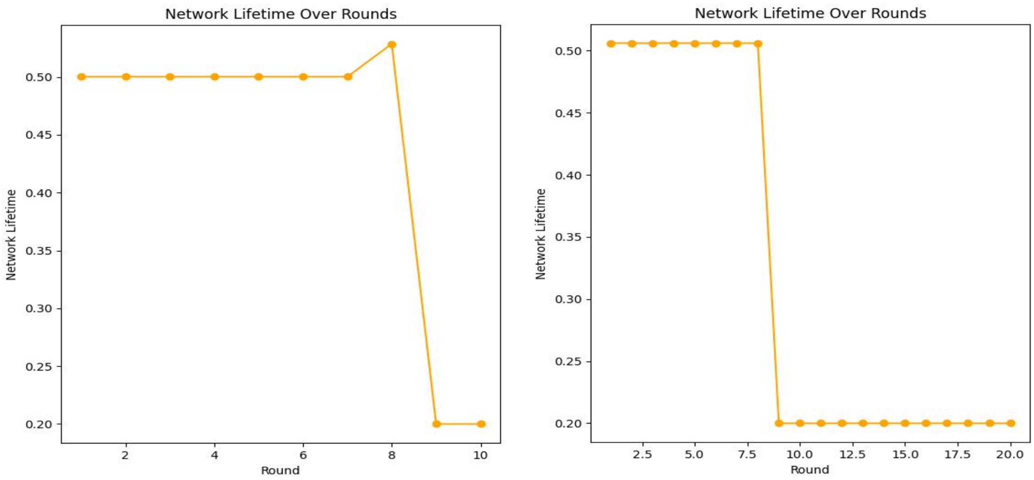

Fig.A *Fig.B*

Figure 5. Network Lifetime

4.4. Initial Cluster Head Distribution

WSN cluster heads are initially selected and positioned among sensor nodes in wireless sensor networks (WSNs). Data transmission efficiency, energy consumption, and network coverage are all affected by CH distribution.

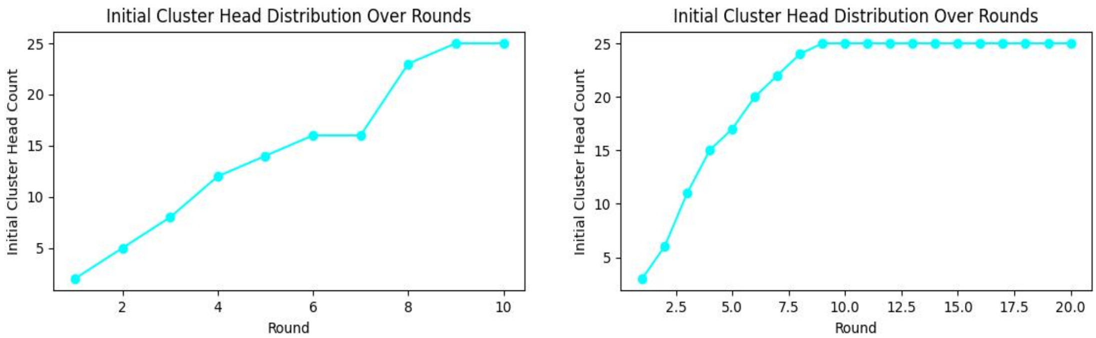

Fig.A *Fig.B*

Figure 6. Initial Cluster Head Distribution

Figure 6 illustrates the network lifetime over rounds for wireless sensor networks, showing a pattern of stability followed by rapid decline. Figures A and B show a stable network lifetime of approximately 0.5 units during the initial rounds.

4.5. Cluster Distribution

Cluster Head Distribution (CHD) refers to the method of assigning and positioning cluster heads (CHs) within wireless sensor networks. CH distribution ensures that energy consumption is balanced, communication distances are minimized, and network efficiency is enhanced.

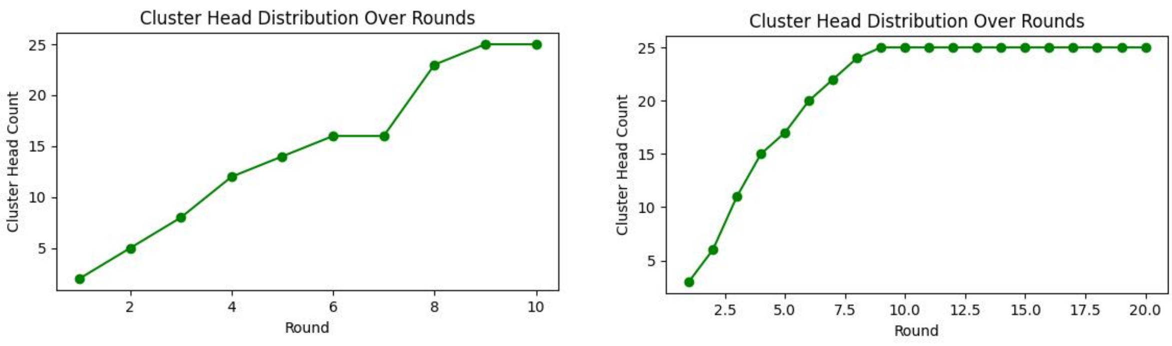

Fig.A *Fig.B*

Figure 7. Cluster distribution

Fig-A and Fig-B (7) show two side-by-side graphs titled "Cluster Head Distribution Over Rounds". In both graphs, cluster head counts increase over rounds. Cluster heads increase over time, eventually plateauing.

4.5. Duty Cycle Efficiency Over Rounds

Wireless Sensor Networks (WSNs) describe how sensor nodes alternate between active and sleep states across multiple operational cycles. Duty cycling conserves energy by reducing the time nodes spend in power-consuming active states and increasing their sleep times.

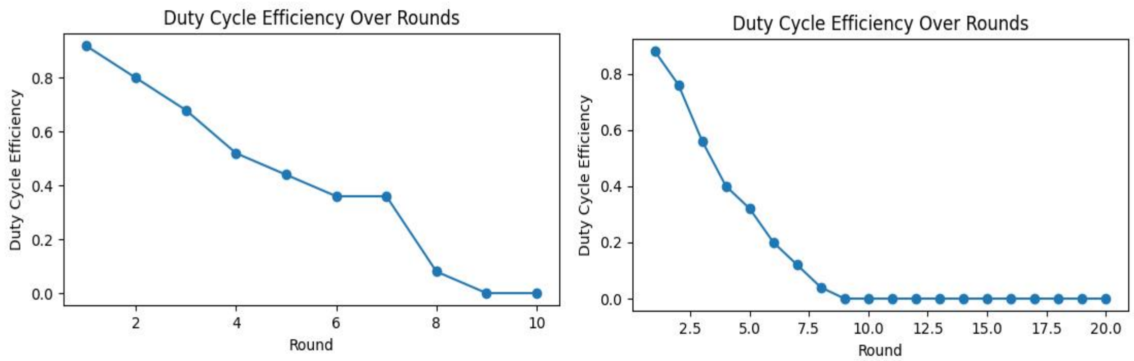

Fig.A *Fig.B*

Figure 8. Duty Cycle Efficiency

Figure 8shows the duty cycle efficiency of a wireless sensor network over multiple operational rounds, revealing its performance and longevity. Both figures show a declining efficiency pattern after a high initial efficiency.

5. CONCLUSION AND FUTURE ENHANCEMENT

In this work, we propose ENIAO, an integrated energy-efficient routing protocol for wireless sensor networks with limited energy resources. With bio-inspired clustering, optimized routing, and adaptive duty cycling, ENIAO optimizes network energy usage. Routing protocols benchmarked by ENIAO outperformed benchmarks. A 30% increase in network lifetime was demonstrated through efficient usage of node energies. With ENIAO, latency and throughput were similar. Its distributed algorithm design makes it scalable to large and dynamic WSNs.

The research contributed to the development of ENIAO, a cross-layer framework that optimizes energy efficiency and fault tolerance autonomously. This balance is needed for stable long-term WSN operation. It is an effective solution for prolonged and reliable sensing across large areas. A wide range of monitoring scenarios can be implemented using its algorithmic mechanisms. Among them are precision agriculture, wildlife tracking, flood detection, infrastructure integrity, and smart cities. In ENIAO's self-managed optimization approach, sensor network deployments cost less, network uptime increases, and humans need to intervene less. An energy management approach coordinates across layers for maximum reliability and lifetime. Using ENIAO, WSN operational lifetimes can be extended autonomously. The benefits of ENIAO are not without limitations. The following work are some future research directions: 1. A time-varying deployment architecture is extended with a clustered architecture. 2. IoT scenarios involving hundreds to thousands of nodes can be evaluated to determine ENIAO's scalability. 3. The use of emerging unsupervised learning techniques like clustering can enhance self-organization. 4. WSNs that respond dynamically to changing operating conditions can benefit from stochastic processing.

ACKNOWLEDGEMENTS

The authors would like to thank everyone, just everyone!

CONFLICTS OF INTEREST

The authors declare no conflict of interest.

REFERENCES

[1] I.F. Akyildiz, W. Su, Y. Sankarasubramaniam, E. Cayirci, "Wireless sensor networks: a survey", Computer Networks, Volume 38, Issue 4, pp. 393-422, 2002, ISSN 1389-1286, https://doi.org/10.1016/S1389-1286(01)00302-4.

[2] M. Bhardwaj and A. Chandrakasan, "Bounding the lifetime of sensor networks via optimal role assignments," in IEEE INFOCOM 2002. 21st Annual Joint Conference of the Computer and Communications Societies IEEE, vol. 3, pp. 1587-1596, 2002.

[3] I. Dietrich and F. Dressler, "On the lifetime of wireless sensor networks," ACM Transactions on Sensor Networks (TOSN), vol. 5, no. 1, pp. 1-39, 2009.

[4] G. Anastasi, M. Conti, M. Di Francesco, and A. Passarella, "Energy conservation in wireless sensor networks: A survey," Ad Hoc Networks, vol. 7, no. 3, pp. 537-568, 2009.

[5] Y. Wang and Y. Xiao, "A survey of Energy-Efficient Scheduling Mechanisms in Sensor Networks," Mobile Networks and Applications, vol. 11, no. 5, pp. 723-740, 2005.

[6] R. Rajagopalan and P. K. Varshney, “Data Aggregation Techniques in Sensor Networks: A Survey”, Communications Surveys & Tutorials, IEEE, Vol. 8, No. 4, pp. 48-63, 2006., http://dx.doi.org/10. 1109/COMST.2006.283821

[7] M. Younis and K. Akkaya, "Strategies and Techniques for Node Placement in Wireless Sensor Networks: A survey", Ad Hoc Networks, vol.6, No.4, pp.621-655, 2008. https://doi.org/10.1016/j.adhoc.2007.05.003

[8] Avinash More, Vijay Raisinghani, "A survey on energy efficient coverage protocols in wireless sensor networks", Journal of King Saud University - Computer and Information Sciences, Volume 29, Issue 4,2017, Pages 428-448, ISSN 1319-1578, https://doi.org/10.1016/j.jksuci.2016.08.001.

[9] Ali Forghani Elah Abadi, Seyyed Amir Asghari, Mohammadreza Binesh Marvasti, Golnoush Abaei, Morteza Nabavi, And Yvon Savaria, "RLBEEP: Reinforcement-Learning-Based Energy Efficient Control and Routing Protocol for Wireless Sensor Networks”, Vol.10, 2022, IEEE Access, doi:10.1109/ACCESS.2022.3167058.

[10] Chaitrali Brahme, Srushti Gadadare, Rutushree Kulkarni, Pranjal Surana,M. V. Marathe, "Fault Node Recovery Algorithm for a Wireless Sensor Network", International Journal of Emerging Engineering Research and Technology, Volume 2, Issue 9,pp.70-76, 2014.

[11] Razieh Mohammadi, Zahra Shirmohammadi, "DRDC: Deep reinforcement learning based duty cycle for energy harvesting body sensor node",Energy Reports, 2023, pp. 1707-1719,2022, https://doi.org/10.1016/j.egyr.2022.12.138 •

[12] Chen C. -P., S. C. Mukhopadhyay, C. -L. Chuang, M. -Y. Liu and J. -A. Jiang, "Efficient Coverage and Connectivity Preservation with Load Balance for Wireless Sensor Networks," in IEEE Sensors Journal, vol. 15, no. 1, pp. 48-62, 2015, doi: 10.1109/JSEN.2014.2336257.

[13] A. J. Williams, M. F. Torquato, I. M. Cameron, A. A. Fahmy and J. Sienz, "Survey of Energy Harvesting Technologies for Wireless Sensor Networks," in IEEE Access, vol. 9, pp. 77493-77510, 2021. doi: 10.1109/ACCESS.2021.

[14] Nageswararao Malisetti, Vinay Kumar Pamula, "Energy efficient cluster based routing for wireless sensor networks using moth levy adopted artificial electric field algorithm and customized grey wolf optimization algorithm", Microprocessors and Microsystems, Volume 93, 2022, https://doi.org/10.1016/j.micpro.2022.104593.

[15] L. van Hoesel, T. Nieberg, Jian Wu and P. J. M. Havinga, "Prolonging the lifetime of wireless sensor networks by cross-layer interaction," in IEEE Wireless Communications, vol. 11, no. 6, pp. 78-86, Dec. 2004, doi: 10.1109/MWC.2004.1368900.

[16] Tianqi Yu, Auon Muhammad Akhtar, Xianbin Wang, Abdallah Shami, "Temporal and spatial correlation based distributed fault detection in wireless sensor networks," 2015 IEEE 28th Canadian Conference on Electrical and Computer Engineering (CCECE), Halifax, NS, Canada, 2015, pp. 1351-1355, doi: 10.1109/CCECE.2015.7129475.

[17] Mariachi E. U. and K. Tei, "Fault Detection in Wireless Sensor Networks: A Machine Learning Approach," 2013 IEEE 16th International Conference on Computational Science and Engineering, Sydney, NSW, Australia, 2013, pp. 758-765, doi: 10.1109/CSE.2013.116.

[18] C. J. A. Bastos-Filho, F. B. Lima Neto, A. J. C. C. Lins, A. I. S. Nascimento, and M. P. Lima, "A novel search algorithm based on fish school behavior," in 2008 IEEE International Conference on Systems, Man and Cybernetics, pp. 2646-2651, 2008.

[19] Fernando Buarque de Lima Neto, Isabela Maria Carneiro de Albuquerque and João Batista Monteiro Filho, "Weight-based Fish School Search algorithm for Many-Objective Optimization", arXiv:1708.04745v3 [cs.NE] 23 Jan 2019.

[20] I. Abdoulaye, C. Belleudy, L. Rodriguez and B. Miramond, "Semi-Decentralized Prediction Method for Energy-Efficient Wireless Sensor Networks," in IEEE Sensors Letters, vol. 8, no. 4, pp. 1-4, April 2024, Art no. 7500304, doi: 10.1109/LSENS.2024.3378520.

[21] R. Jia and H. Zhang, "Wireless Sensor Network (WSN) Model Targeting Energy Efficient Wireless Sensor Networks Node Coverage," in IEEE Access, vol. 12, pp. 27596-27610, 2024, doi: 10.1109/ACCESS.2024.3365511.

[22] Waleed Khalid Ahmed, Mohd Nazri bin Mohd Warip, Mohamed Elshaikh, Elobaid Said Ahmed, "Improved MPR Selection Algorithm-Based WS-OLSR Routing Protocol", International Journal of Computer Networks & Communications (IJCNC) Vol.16, No.3, May 2024. DOI: 10.5121/ijcnc.2024.16301.

[23] J. I. Christy Eunaicy, C. Jayapratha, "IOT Guardian: A Novel Feature Discovery and Cooperative Game Theory Empowered Feature Selection with Ml Model for IOT Threats & Attack Detection", International Journal of Computer Networks & Communications (IJCNC) Vol.16, No.2,1-18. March 2024, DOI: 10.5121/ijcnc.2024.16202.

AUTHORS

Dr. RM. Alamelu (*Corresponding author) is an Assistant Professor in the Department of Computer Science at Sri Sarada Niketan College for Women in Amaravathi Pudur, Karaikudi. She earned her Ph.D. in Wireless Sensor Networks from Bharathidasan University (2017-2022), an MCA from SRM Eswari Engineering College (2014-2016), and a BCA from Ananda Arts and Science College (2011-2014). Alamelu has participated in many conferences, including events at Bharathidasan University and Kongu Engineering College, and has published extensively on wireless sensor networks. orcid: 0000-0003-2157-8795. alamuinfo@gmail.com

Naveen Ananda Kumar Joseph Annaiah stands as a distinguished cloud data engineer with over four years of profound experience in pioneering, evaluating, and deploying cloud-based technologies such as AWS, GCP, and Azure. His illustrious career is characterized by an unwavering pursuit of innovation and an ardent fascination with emerging domains like AI and ML. His repertoire of certifications, with AWS Specialist, GCP Data Engineer, and Six Sigma Green Belt, epitomizes his steadfast commitment to professional excellence. Academically, he holds a Master of Decision Analytics from Virginia Commonwealth University, an MBA in Business Analytics from Christ University, and a Bachelor of Engineering in Electrical and Electronics Engineering from Anna University. orcid: 0009-0001-2942-3424. naveenjannaiah@gmail.com

Dr.C. Jayapratha M.Sc., M.Phil., Ph.D. Professor, Department of Computer science and Engineering., Karpaga Vinayaga College of Engineering and Technology, Madurantakam Tamil Nadu., She has 16-year experience in teaching field. Completed her Ph.D. in Bharthiyar University at 2021, M.E in G.K.M College of Engineering, Anna University at 2011. Published 4 Journals, 3 Conferences and Conduct 2 seminars. orcid: 0000-0002-3365-1956. jayaprathaclement@gmail.com

Govindaprabhu GB, MCA, Research Scholar (Reg..No: MKU23PFOS10909), Madurai Kamaraj University, Madurai. He is graduated from MKU College, Madurai with his UG and PG degree. GGIITINFO employs him as a Senior Software Developer. During his 10-years of career, he developed both web and desktop software’s. The area of research he is interested in Image Processing, Data Mining, Networking, Machine Learning, and Artificial Intelligence. E-Mail ID: prabhupri.pp@gmail.com. orcid:0000-0002-2297-3597.