International Journal of Computer Networks and Applications (IJCNA)
DOI: 10.22247/ijcna/2024/224441 Volume 11, Issue 2, March – April (2024)

RESEARCH ARTICLE

Energizing Firefly Optimization-Inspired Routing Protocol (EFOIRP) for Performance Enhancement in IOT-Based Cloud Wireless Sensor Networks (IC-WSN)

J. Jerlin Adaikala Sundari
Department of Computer Technology, PSG College of Arts and Science, Coimbatore, Tamil Nadu, India.
jerlinadaikalasundari@psgcas.ac.in

G. Preethi
Department of Computer Science, PRIST University, Thanjavur, Tamil Nadu, India.
mgpreethi@gmail.com

Received: 13 December 2023 / Revised: 15 February 2024 / Accepted: 25 February 2024 / Published: 30 April 2024

Abstract – Physical obstructions that disrupt signal propagation and routing paths hinder routing performance in IoT-based Cloud Wireless Sensor Networks (IC-WSN) for greenhouse farming. Existing routing algorithms fail to address the energy consumption challenge, resulting in suboptimal routing paths and potential data loss. This paper proposes an Energizing Firefly Optimization-Inspired Routing Protocol (EFOIRP) to enhance performance in IC-WSN. The protocol employs novel routing strategies to handle physical obstructions within greenhouses. It includes comprehensive site surveys to identify obstructions and their impact on signal propagation, enabling intelligent path selection that minimizes obstruction effects and ensures reliable data transmission. This research aims to achieve seamless data transmission and monitoring in greenhouse farming. EFOIRP minimizes signal interference by addressing physical obstructions, optimizing data transmission efficiency, and empowering farmers with reliable and accurate data for precise control over greenhouse conditions and resource management. The research objectives encompass characterizing obstructions, developing adaptive routing algorithms, evaluating performance through simulations or experiments, investigating scalability, and validating effectiveness in real-world greenhouse farming scenarios. The proposed EFOIRP aims to overcome the limitations of existing routing algorithms and improve the performance of IC-WSN in greenhouse farming environments.

Index Terms – Cloud, Greenhouse, Firefly Optimization, IoT, Routing Protocol, Wireless Sensor Networks

1. INTRODUCTION

Greenhouse farming, a modern and innovative agricultural technique, has gained significant attention recently for its potential to revolutionize food production. This method involves cultivating crops within enclosed structures, such as glass or plastic greenhouses, where environmental conditions can be carefully regulated and controlled [1]. By harnessing advanced technology and scientific principles, greenhouse farming offers a range of possibilities for overcoming traditional farming limitations and meeting the challenges of a rapidly changing world. Greenhouse farming stems from the need to address various agricultural constraints, such as unpredictable weather patterns, limited arable land availability, and the growing demand for fresh produce throughout the year [2]. By providing a controlled environment, greenhouse farming extends the growing season and allows for year-round crop cultivation, irrespective of external climatic conditions. This opens up immense opportunities for regions with harsh climates or limited agricultural potential, enabling them to become self-sufficient in food production and reduce dependence on imports [3].

Greenhouse farming presents a viable solution to the issue of resource scarcity and environmental sustainability. With precise control over factors like water, nutrients, and energy inputs, this method ensures optimal resource utilization and minimizes waste [4]. Greenhouse farming significantly reduces water consumption and mitigates the negative environmental impact by implementing efficient irrigation systems, recycling water, and adopting advanced farming practices such as hydroponics or aquaponics. The enclosed structure of greenhouses shields against pests, diseases, and invasive species, reducing the reliance on chemical pesticides and genetically modified crops [5]. This promotes ecological balance and contributes to producing healthier and safer food. Greenhouse Farming represents a promising and transformative approach to agriculture, offering opportunities for year-round cultivation, efficient resource management,

RESEARCH ARTICLE

and sustainable food production. With its potential to overcome geographical limitations, enhance resource efficiency, and ensure food security, greenhouse farming is poised to play a vital role in shaping the future of farming and meeting the growing demands of a rapidly evolving world [6].

The advent of Internet of Things (IoT) technology has paved the way for revolutionary advancements in various industries, and one such groundbreaking innovation is the Internet of Things-based Cloud Wireless Sensor Network (IC-WSN). This cutting-edge technology combines the power of wireless sensor networks (WSN) and cloud computing to create a robust and interconnected system for data acquisition, analysis, and management [7]. By leveraging the capabilities of IoT and cloud computing, IC-WSN offers unprecedented opportunities for real-time monitoring, intelligent decision-making, and enhanced operational efficiency. At its core, IC-WSN consists of wireless sensors strategically deployed to collect data from the physical environment [8]. These sensors have various sensing capabilities that capture crucial information such as temperature, humidity, light intensity, and more. The collected data is transmitted wirelessly to a cloud-based platform, securely stored, processed, and analyzed [9].

The integration of cloud computing in IC-WSN brings remarkable advantages. Cloud infrastructure provides the scalability and computational power required to handle vast sensor data, enabling real-time analytics and insights. Additionally, the cloud's flexible storage capabilities ensure seamless data management and easy accessibility from anywhere worldwide, empowering users to make informed decisions based on up-to-date information [10]. IC-WSN finds applications in numerous fields, ranging from environmental monitoring and smart cities to industrial automation and precision agriculture. By facilitating efficient data collection, analysis, and communication, this technology enhances operational efficiency, optimizes resource allocation, and enables proactive decision-making [11]. IC-WSN represents a game-changing technology that harnesses the power of wireless sensors and cloud computing. With its ability to allow real-time monitoring, intelligent data analysis, and remote accessibility, IC-WSN opens up new possibilities for innovation and optimization across diverse industries, driving us towards a more connected and data-driven future [10].

In agricultural contexts, IoT-based Wireless Sensor Network (IC-WSN) technology plays a crucial role in meeting the demands of greenhouse farming. Greenhouses provide controlled environments conducive to optimized crop growth, but effective monitoring and management are essential. IC-WSNs offer a solution by enabling real-time data collection and analysis, ensuring precise control and efficient resource utilization [12], [13]. By deploying wireless sensors throughout the greenhouse, IC-WSNs allow continuous monitoring of critical parameters like temperature, humidity, soil moisture, and light levels. Farmers transmit this data wirelessly to a cloud-based platform, empowering them to access and analyze it remotely in real-time. IC-WSNs enable farmers to gain valuable insights into environmental conditions, detect anomalies, and make data-informed decisions to optimize crop growth. The continuous monitoring and control ability of IC-WSNs is highly advantageous in greenhouse farming. It allows farmers to make precise adjustments to environmental factors, such as temperature and irrigation, tailored to the specific needs of different plant varieties. This ensures optimal growing conditions, enhances crop quality, and reduces resource waste [14].

Routing in IC-WSN for greenhouse farming is the impact of physical obstructions on the routing performance. Greenhouses often contain dense vegetation, equipment, and structures that can interfere with signal propagation and disrupt the routing paths. These physical obstructions pose challenges to maintaining reliable and efficient data transmission. Routing protocols must account for physical obstructions and find alternative routes to ensure uninterrupted data flow. However, existing routing algorithms may not effectively address these challenges, resulting in suboptimal routing paths and potential data loss. To tackle this problem, novel routing strategies must be developed to handle physical obstructions within the greenhouse. This could involve conducting comprehensive site surveys to identify potential obstructions and their impact on signal propagation. Based on this information, routing protocols can intelligently select paths that minimize the effects of obstructions and ensure reliable data transmission.

Addressing the impact of physical obstructions on routing in IC-WSN for greenhouse farming is crucial for ensuring seamless data transmission and monitoring. By developing novel routing strategies that can handle the presence of physical obstructions, we can achieve several critical motivations for greenhouse farming. Firstly, accounting for these obstructions can minimize signal interference and disruptions, resulting in reliable and continuous data flow. This enables real-time monitoring of crucial environmental parameters, ensuring optimal conditions for plant growth and maximizing crop yields. Secondly, by intelligently selecting paths that avoid or minimize the effects of obstructions, we can optimize data transmission efficiency, reducing latency and potential data loss. Implementing effective routing protocols that address physical obstructions empowers greenhouse farmers with reliable and accurate data, enabling precise control over greenhouse conditions and facilitating efficient resource management.

The research objective of this study is to address the impact of physical obstructions on routing performance in IC-WSN for

RESEARCH ARTICLE

greenhouse farming. The specific research objectives are as follows:

- Characterize and classify the physical obstructions commonly found in greenhouse environments, including dense vegetation, equipment, and structures, and assess their effects on signal propagation.
- Develop novel routing algorithms that intelligently adapt to physical obstructions, considering signal strength, obstacle density, and alternative routing paths.
- Evaluate the performance of the proposed routing algorithms through extensive simulations or practical experiments, measuring metrics such as packet delivery ratio, latency, and energy consumption in the presence of different types and densities of obstructions.
- Investigate the scalability of the routing algorithms for large-scale greenhouse environments, considering the dynamic nature of obstructions and the increasing number of sensors.
- Validate the effectiveness of the developed routing algorithms in real-world greenhouse farming scenarios, assessing their ability to ensure reliable data transmission, minimize packet loss, and optimize overall network performance.

The paper is organized into six main sections. Beginning with Section 1, the Introduction, it addresses the problem statement, motivation, and research objectives. Section 2, the Literature Review, provides insights into existing routing protocols, optimization techniques, and performance enhancement strategies. Section 3 introduces the EFOIRP, detailing its design principles and functionality. Section 4, Simulation Settings, describes the experimental setup and parameters for simulating the EFOIRP in IC-WSN environments. Section 5, Results and Discussion, presents findings from simulations, analyzing the performance of the EFOIRP and comparing it with existing protocols. Finally, Section 6, the Conclusion, summarizes key findings, highlights the significance of the EFOIRP, and suggests future research directions. This structured approach offers a comprehensive framework for understanding and advancing knowledge in the domain of IC-WSN.

2. LITERATURE REVIEW

"Energy-Efficient Cooperative Routing" [15] leverages cooperative communication techniques to improve network efficiency. In heterogeneous WSNs, nodes with different capabilities and energy levels coexist. The routing scheme intelligently selects cooperative nodes based on their energy levels and proximity to the destination. By forming cooperative clusters and employing cooperative routing strategies, the scheme reduces the energy burden on individual nodes and extends the network's operational lifetime. It enables efficient data transmission by leveraging the capabilities of neighbouring nodes, leading to improved network performance and energy efficiency in heterogeneous WSNs. "Sustainable Multipath Routing Protocol" [16] addresses the unique requirements of multi-sink WSNs deployed in harsh environments where reliable communication is crucial. It utilizes a multipath routing approach to establish multiple paths from source nodes to multiple sink nodes, enhancing reliability and robustness against link failures. The protocol also incorporates energy-aware mechanisms to optimize energy consumption and prolong the network lifetime. By considering the specific challenges of harsh environments and adopting sustainable routing strategies, this protocol enables reliable and energy-efficient data transmission in multi-sink WSNs, contributing to their long-term operation and performance in challenging conditions.

"Trusted Routing Scheme" [17] combines deep learning principles and blockchain technology to provide a secure and transparent routing scheme. Deep learning techniques analyze network data and identify potential routing anomalies or malicious activities. With its decentralized and immutable nature, blockchain technology ensures the integrity and transparency of routing information. The proposed routing scheme enhances trust and security in WSNs by leveraging this deep blockchain approach, enabling reliable data transmission and protecting against various routing attacks. "Centralized Routing Protocol" [18] is designed to address the threat of wormhole attacks in WSNs. Wormhole attacks involve an adversary establishing a high-speed link between two distant points in the network, creating a shortcut for data transmission. This protocol employs a centralized approach where a central authority monitors and detects potential wormhole attacks. The central authority collects and analyzes network data, looking for suspicious patterns that may indicate the presence of wormholes. Appropriate countermeasures are taken upon detection, such as isolating the affected nodes or re-routing the traffic.

"Secure Routing Technique" [19] enhances routing security in IoT-based WSNs. This technique utilizes the Blowfish encryption algorithm, an established symmetric key block cipher, to ensure the confidentiality and integrity of routing data. By applying the Improved Blowfish algorithm to the routing process, the technique protects against potential security threats, such as eavesdropping and data tampering. The algorithm ensures secure communication between sensor nodes and the IoT infrastructure, mitigating the risk of unauthorized access and data breaches. "Energy Harvesting Routing" [20] relies on energy harvesting sources, such as solar or wind, to replenish their batteries and support continuous operation. The routing protocol aims to optimize energy utilization by efficiently routing data packets while

RESEARCH ARTICLE

considering the availability and capacity of alternative dual batteries. The protocol dynamically selects energy-efficient paths based on the energy levels of the batteries and the energy harvesting rates. It considers the state of both batteries and prioritizes routing decisions to balance the energy consumption between them. The routing algorithm adapts to the changing energy availability, adapting the routing paths to maximize the utilization of harvested energy.

"Energy Efficient Data Aggregation" [21] optimizes energy efficiency, data aggregation, and end-to-end security. It intelligently selects nodes for data aggregation based on proximity and energy levels, reducing redundant transmissions and conserving energy. The protocol incorporates encryption and authentication mechanisms to ensure secure data transmission. By combining energy efficiency, data aggregation, and end-to-end security, E2DA enables efficient resource utilization, enhances network scalability, and ensures secure data transmission in dynamic and challenging 3D WSN environments." Improved Routing Protocol" [22] is a specialized routing protocol designed for Heterogeneous WSNs deployed in IoT-based environmental monitoring applications. It addresses the challenges of heterogeneity within the network by considering sensor nodes' varying capabilities and characteristics. The protocol utilizes an improved routing algorithm that considers node energy levels, data packet size, and communication distances to select efficient routes. By dynamically adapting to changes in the network topology, it ensures reliable and energy-efficient data transmission. It is specifically tailored for IoT-based environmental monitoring systems, where accurate and timely data delivery is crucial.

"Distributed 2-Hop Cluster Routing Protocol" [23] enables sensor nodes to autonomously form clusters based on various criteria, with each cluster electing a cluster head responsible for intra-cluster communication. The cluster heads establish 2-hop communication links with neighbouring cluster heads to facilitate inter-cluster communication. This approach offers advantages such as reduced energy consumption through shorter hops, improved scalability by distributing routing decisions, and enhanced network reliability through alternative communication paths. D2CRP provides an efficient and reliable data routing solution for WSNs, leveraging the benefits of distributed clustering and 2-hop communication. "Energy Efficient Environment-Aware Fusion" [24] prioritizes energy efficiency, environmental awareness, and reliable data fusion. It adapts routing decisions based on ecological conditions, utilizes data fusion techniques to conserve energy, and ensures reliable data transmission through error control coding and dynamic route selection. It optimizes resource utilization, prolongs network lifetime, and enhances data transmission reliability in WSNs, making it suitable for energy-efficient and reliable routing applications. Bio-inspired optimization-based Routing Protocols [25]-[29], plays a significant role in the network to achieve better efficiency.

"Destination-Oriented Routing Algorithm (DORA)" [30] is a routing algorithm designed explicitly for Energy-Balanced WSNs. The main objective of DORA is to prolong the network lifetime by evenly distributing energy consumption among sensor nodes. DORA operates based on a destination-oriented approach, where routing decisions are made considering the energy levels and distances to the destination nodes. The algorithm considers each node's residual energy and dynamically selects the optimal next-hop node that minimizes energy consumption along the routing path. DORA employs a load-balancing mechanism that evenly distributes data traffic among sensor nodes to achieve energy balancing. This mechanism avoids energy depletion in specific nodes by dynamically adjusting the routing paths and promoting energy-efficient data transmission. DORA utilizes a proactive approach to periodically update the routing tables, ensuring reliable and efficient data forwarding in the network. It considers the changing network conditions, such as node failures or energy variations, and adapts the routing paths accordingly.

"Particle Swarm Optimization Routing Scheme (PSORS)" [31] mimics the behaviour of a swarm of particles navigating through a search space to find optimal solutions. Each sensor node in the network is represented as a particle. These particles communicate and adjust their positions within the search space to explore and exploit the most favourable routing paths. The particles converge towards optimal solutions by iteratively updating their positions using local and global information. It integrates two crucial elements: exploration and exploitation. Exploration involves randomly exploring the search space to discover potential routing paths, while exploitation focuses on refining and improving these paths based on their quality. Within WSNs, the PSORS employs fitness functions to evaluate the quality of routing paths. These functions consider energy consumption, data transmission delay, and network connectivity. The particles strive to discover optimal routing paths that minimize energy consumption and maximize network performance by adjusting their positions and velocities based on fitness values.

3. ENERGIZING FIREFLY OPTIMIZATION-INSPIRED ROUTING PROTOCOL (EFOIRP)

3.1. Low Energy Adaptive Clustering Hierarchy

The Low Energy Adaptive Clustering Hierarchy (LEACH) protocol is a distributed routing protocol designed for WSNs to conserve energy. The LEACH protocol has two main stages: initialization and operation at a constant rate. The intended proportion of cluster heads (p) is used by the sensor nodes during setup calculations to generate a threshold value,

RESEARCH ARTICLE

$T(n)$. This cutoff number establishes the likelihood that a specific node will act as the cluster leader for this iteration. Sensor nodes generate random numbers and check them against a threshold value. The node becomes a cluster leader if the produced number is less than or equal to the threshold value. Once chosen, the cluster leaders will notify the remainder of the network, while the remaining nodes will form a new cluster under the leadership of the nearest cluster leader. At a steady state, sensor nodes within every cluster collect data from their immediate environments and relay it to the cluster leaders via localized communication. Afterwards, the cluster leaders aggregate the data to reduce duplication and power consumption. Once the data has been aggregated, it is sent through multi-hop transmission to the base station or sink node for further processing. LEACH uses a round-based procedure to distribute power consumption fairly across the sensor nodes. In each subsequent round, cluster heads are randomly chosen to distribute the energy-intensive role. This dynamic selection mechanism for cluster heads promotes fair energy distribution and enhances the network's resilience. In summary, the LEACH protocol facilitates the operation of energy-efficient wireless sensor networks and prolongs the network's lifespan by effectively managing energy consumption.

3.1.1. Setup Phase

The LEACH protocol's first phase is crucial because it allows cluster chiefs to be elected by the sensor nodes. This phase is significant for the protocol's overall operation and energy optimization. Sensor nodes perform computations during setup to calculate a threshold value, which is then used to choose the required proportion of cluster heads. The threshold value heavily influences the likelihood of a node being selected as the round's cluster leader. Employing a probabilistic approach, the selection process ensures an equitable distribution of the cluster head role throughout the network. This distributed clustering mechanism fosters efficient data aggregation and routing, ultimately leading to improved energy efficiency and an extended lifespan of wireless sensor networks.

3.1.1.1. Threshold Calculation

The threshold value establishes the likelihood of a node assuming the role of a cluster head in a specific round. In this step, each sensor node calculates a threshold value, $T(n)$, where n represents the current round number. The calculation of function $T(n)$ is influenced by the desired proportion of cluster heads (p) and can be mathematically represented using Eq.(1).

$$T(n) = p/\left(1 - p \times \left(nmod(1/p)\right)\right) \quad (1)$$

3.1.1.2. Cluster Head Selection

Every sensor node generates a random number between 0 and 1, employing the threshold value. If the generated number is equal to or less than $T(n)$, the node takes on the responsibility of being a cluster head for the present round. This randomized selection process ensures an equitable dispersion of cluster heads throughout the network.

3.1.1.3. Cluster Formation

The nodes at the centre of clusters announce their choice to the rest of the network. Non-cluster head nodes receive these broadcasts and choose the nearest cluster head to join its cluster. This method allows clusters to emerge with minimal ties between their constituent nodes and cluster leaders.

3.1.2. Steady-State Phase

After the establishment of clusters in LEACH, the subsequent phase, called the steady-state phase, commences, involving the transmission and aggregation of data. Sensor nodes collect information in every cluster and transmit it to their corresponding cluster leader via short-distance communication. Subsequently, the cluster leaders engage in data aggregation to reduce redundancy and preserve energy. Multi-hop communication relays The combined data to the base station or sink node. In the end, this method improves the power efficiency of wireless sensor networks by ensuring more effective data processing and routing. The following steps can summarize the process:

3.1.2.1. Data Transmission

Sensor nodes within each cluster collect data from their sensing environment. The sensor nodes transfer this data to their corresponding cluster leaders through direct communication links. Using short-range communication reduces energy consumption compared to transmitting data over long distances.

3.1.2.2. Data Aggregation

The cluster leaders have a crucial role in the process of data aggregation. They receive data from numerous nodes within their clusters and employ aggregation methods to minimize the volume of data transmitted over long distances. Aggregation techniques such as averaging, clustering, or compression can be employed to reduce redundancy and conserve energy.

3.1.2.3. Cluster Head-to-Sink Communication

Once data aggregation is complete, the cluster leaders use multi-hop communication to send the compiled information to the network's hub node, i.e., the sink node. This involves passing the data through intermediate cluster heads until it reaches the designated sink node. Adopting multi-hop communication facilitates efficient data routing and

RESEARCH ARTICLE

minimizes energy consumption for individual sensor nodes by eliminating the need for direct long-distance transmissions.

3.1.3. Round-Based Operation

LEACH employs a round-based methodology to distribute the energy burden across the entire network equitably. This ground-based operation helps balance individual sensor nodes' energy consumption, preventing a few nodes from depleting their energy quickly and prolonging the network's overall lifetime. In each subsequent round, the selection of cluster heads is randomized again, ensuring that different nodes can become cluster heads. This dynamic cluster head selection mechanism promotes fair energy distribution and enhances the network's resilience by preventing network disruptions caused by the failure of individual nodes. The following aspects illustrate its round-based operation:

3.1.3.1. Cluster Head Rotation

In every subsequent round, the selection of cluster heads is randomized new. This rotation spreads the sensor nodes' energy consumption fairly, ensuring that energy depletion does not occur rapidly in a select few nodes. This approach promotes fairness in energy utilization across the network. Randomized selection increases the probability of different nodes becoming cluster heads in different rounds, promoting fair energy distribution.

3.1.3.2. Fault Tolerance

LEACH incorporates fault tolerance mechanisms to ensure network reliability. If a cluster head fails, the network continues its operation by randomly selecting a new cluster head from the remaining nodes. This dynamic cluster head selection mechanism prevents network disruptions caused by the failure of individual nodes and enhances the overall resilience of the network.

Input:

- p: The targeted proportion of cluster heads
- n: The present round count
- Sensor nodes' information (location, energy level, etc.)

Output:

- Cluster formation and data aggregation in the network

Procedure:

Step 1: Setup Phase

- Using equation (1), every sensor node computes the threshold value, $T(n)$.
- For every sensor node:

✓ Produce a random number, r, within the range of 0 to 1.

✓ If the generated random number, r, satisfies the condition $r \leq T(n)$, the node is designated as a cluster head for the current round.

- Cluster heads transmit their selection to the remaining nodes through broadcasting.
- If a node isn't already part of a cluster's head, it will join the cluster of the closest head.

Step 2: Steady-State Phase

- The cluster leaders gather data from the sensor nodes within their corresponding clusters.
- Cluster heads perform data aggregation techniques (e.g., averaging, clustering, or compression) to reduce data redundancy.
- The information gathered by the cluster heads is sent to the base station or sink node through multi-hop communication.
- The central hub, the sink node, collects information from the cluster's leaders.

Step 3: Round-Based Operation

- Repeat steps 1 and 2 in subsequent rounds to balance energy consumption.
- During each round, recalculate the threshold value, $T(n)$, based on the present round number and the desired percentage of cluster heads, and the data is collected.
- Randomly select cluster heads based on the threshold value.
- If a cluster head fails, randomly select a new one from the remaining nodes to ensure fault tolerance.

Algorithm 1 LEACH Protocol

3.2. Firefly Optimization

Firefly Optimization is a metaheuristic algorithm that draws inspiration from the behaviour of fireflies. This population-based algorithm emulates the flashing pattern of fireflies to explore and discover optimal solutions to various optimization problems. Firefly Optimization was first proposed by Xin-She Yang in 2008 and has since gained popularity due to its simplicity and effectiveness. The algorithm capitalizes on the attractive and repulsive actions displayed by fireflies. A firefly's attractiveness is determined by its brightness, which, in turn, is influenced by its distance from other fireflies. According to the algorithm, fireflies emitting brighter light are considered more attractive and closer to the optimal solution. The firefly optimization process commences by establishing an initial population of fireflies, with each firefly representing a potential solution to

RESEARCH ARTICLE

the given optimization problem. The value of the objective function associated with each solution controls the intensity of the fireflies' lights. The algorithm iteratively updates the positions of fireflies using the following steps:

- Initialization: In this step, an initial population of fireflies is generated randomly. The population size is typically determined based on the complexity of the optimization problem. Every firefly within the algorithm represents a potential solution to the given problem.
- Evaluation: Once the population is initialized, every firefly's objective function is evaluated, and its brightness is assigned based on its fitness or objective function value. The brightness of a firefly represents its quality or how close it is to the optimal solution. Typically, brighter fireflies have lower objective function values.
- Movement: Fireflies tend to navigate towards fireflies that emit more brightness, enabling them to explore the search space. The attractiveness and distance between fireflies guide the movement. A firefly's attractiveness level is determined by its brightness concerning other fireflies. As the distance between fireflies increases, their attractiveness diminishes. The movement is random, with a step size proportional to the distance between fireflies. This randomness allows for the exploration of the search space.
- Update: After each movement, the positions of the fireflies are updated based on their new locations. The fireflies' brightness is also updated based on their new positions and objective function evaluations. Suppose a firefly moves to a better position with a lower objective function value, its brightness increases. Conversely, if a firefly moves to a worse position, its brightness may decrease.
- Reinforcement: Fireflies reinforce their brightness by comparing themselves with other fireflies in their neighbourhood. This helps intensify the search for near-promising regions of the search space. Fireflies exchange information about their brightness, and if a firefly finds a brighter neighbour, it adjusts its position towards that neighbour, thus reinforcing its brightness.
- Termination: The algorithm iteratively repeats steps 3-5 until a termination condition is met. Common termination conditions include reaching a maximum number of iterations, achieving a satisfactory solution within a predefined tolerance, or combining both. Once the termination condition is satisfied, the algorithm halts, and the best solution found by the fireflies is returned as the output.

Input:

- Graph G represents the network
- Source node S
- Destination node D
- Maximum number of iterations
- Population size
- Firefly movement parameters
- Termination condition

Output:

- Optimal route from S to D

Procedure:

Step 1: Set up the algorithm parameters, including the population size, the maximum number of iterations, the movement parameters of the fireflies (such as step size and decay of attractiveness), and the termination condition.

Step 2: Generate an initial population of fireflies, where each represents a potential route from S to D. Randomly generate routes satisfying the problem constraints.

Step 3: Evaluate the fitness of each firefly route by calculating a suitable objective function, such as the total distance or cost of the route. Assign brightness values to the fireflies based on their fitness.

Step 4: Repeat the following steps until the termination condition is met:

a. For each firefly in the population:

b. Calculate the attractiveness of the firefly based on its brightness and the distance from other fireflies.

c. Move the firefly towards brighter fireflies by updating its position. The movement is random, with a step size proportional to the distance between fireflies.

d. If the new position violates any constraints, discard the move and keep the firefly at its current position.

e. Evaluate the new position's fitness and update the firefly's brightness accordingly.

f. Reinforce the brightness of fireflies by comparing their brightness values with their neighbours. Adjust the positions of fireflies towards brighter neighbours.

g. If a firefly with an improved brightness value is discovered, the global best solution will be updated.

Step 5: Check the termination condition. If it is satisfied (e.g., the maximum number of iterations reached or a satisfactory solution is found), go to step 7. Otherwise, go to the next iteration.

Step 6: Repeat steps 4 and 5.

RESEARCH ARTICLE

Step 7: Select the firefly with the highest brightness value as the optimal route from S to D.

Step 8: Return the optimal route as the output.

Algorithm 2 Firefly Optimization for Routing

3.3. Chaos

Under deterministic conditions, a deterministic nonlinear system exhibits chaotic behaviour characterized by its erratic movement pattern and unexpected behaviour. Chaotic behaviour arises in a nonlinear system when it demonstrates infinite possible periodic responses and shows strong sensitivity to the initial conditions. Even minute changes in the initial conditions can profoundly impact the output of a chaotic system. Researchers often employ chaotic maps to model chaos, constructed as dynamical discrete-time continuous value functions. These maps capture the intricate relationship between a chaotic system's present state and future trajectory. Researchers gain insights into chaotic systems' complex dynamics and predictability by utilizing these maps. Chaotic one-dimensional maps find extensive application in various real-world scenarios, offering a valuable tool for understanding and analyzing nonlinear systems. They are mathematically described by Eq.(2), representing these maps' formal structure. Using chaotic maps, researchers can uncover the underlying mechanisms driving chaotic behaviour and facilitate the development of effective strategies for control and prediction in nonlinear systems.

$$p_{t+1} = \gamma(p_t) \tag{2}$$

Wherein p_{t+1} and p_t are two consecutive chaotic numbers, and γ is the mapping function of the forward transformation.

Various types of chaotic number generators have been implemented to address different requirements. In automatic design optimization, the logistic, sinusoidal, and tent maps were particularly effective in three specific chaotic maps. The current research also leverages the capabilities of these maps. Additionally, a concept known as "piecewise chaos" is employed, which involves a probabilistic combination of these chaotic maps.

According to the principle of piecewise chaos, generating a chaotic number involves two steps. The first step is to draw a random number, b, from an even distribution between 0 and 1. Subsequently, the following requirements are examined to determine the appropriate chaotic map for generating the following chaotic number. If b is less than 0.33, the chaotic logistic map is employed. The sinusoidal chaotic map is utilized if b is greater than or equal to 0.33 and less than 0.66. Otherwise, if b is not within these ranges, the tent map chaotic map calculates the following chaotic number. Each chaotic map within the piecewise chaos framework individually stores the chaotic numbers it receives as input, referred to as predecessor chaotic numbers. These predecessor chaotic numbers are then utilized, as needed, to construct the following chaotic number, known as the successor number, over multiple generations. Moreover, distinct random seeds are employed to initialize each chaotic map to ensure the uniqueness and independence of each chaotic map.

3.3.1. Logistic Map

The logistic map has been extensively examined and researched among the chaotic maps. It captures the population dynamics of simple models and has been used to describe various natural phenomena. The logistic map is defined by the Eq.(3).

$$x_{\{n+1\}} = r * x_n * (1 - x_n) \tag{3}$$

Where x_n represents the value at iteration n, and r is the control parameter. The logistic map exhibits a range of behaviours based on the value of r. When r lies within the interval (3.57, 4.0), the map displays chaotic behaviour, characterized by extreme sensitivity to initial conditions and the emergence of complex, unpredictable patterns.

3.3.2. Sinusoidal Map

The sinusoidal map, also known as the sine map, is a chaotic map that relies on trigonometric functions to generate chaotic behaviour. It is given by Eq.(4).

$$x_{\{n+1\}} = a * sin(pi * x_n) \tag{4}$$

Where x_n represents the value at iteration n, and a is the control parameter. The sinusoidal map operates on the interval [0, 1], and for specific values of a, it exhibits chaotic dynamics. Chaotic behaviour emerges when a fall is between approximately 1.9 and 2.5. Trajectories generated by the sinusoidal map exhibit irregular oscillations and sensitivity to initial conditions.

3.3.3. Tent Map

The tent map is a piecewise linear chaotic map with a simple yet intriguing structure. It is defined by Eq.(5).

$$\begin{aligned} p_{\{n+1\}} &= r * x_n \; for \; 0 \le x_n < 0.5 \; and \; x_{\{n+1\}} \\ &= r * (1 - x_n) \; for \; 0.5 \le x_n < 1 \end{aligned} \tag{5}$$

Where x_n represents the value at iteration n, and r is the control parameter. The tent map operates on the interval [0, 1] and exhibits chaotic behaviour when r is greater than 1. Chaotic trajectories generated by the tent map display intricate patterns characterized by bifurcations and sensitivity to initial conditions. The parameter r influences the number of branches and the overall behaviour of the map.

RESEARCH ARTICLE

3.4. Energizing Firefly Optimization-Inspired Routing Protocol

The EFOIRP is a modification of the Firefly Algorithm (FA) that adds the logistic map to improve the algorithm's ability to explore and exploit new areas. The Firefly Algorithm is a problem-solving technique inspired by the flashing behaviour of fireflies. It utilizes the concept of attractiveness between fireflies to optimize various problems. EFOIRP introduces chaotic dynamics into the firefly movement by incorporating the logistic map, thereby improving the algorithm's search efficiency. The six phases involved in EFOIRP are:

- Logistic Map Initialization: EFOIRP initializes the firefly population using the logistic map. The initial population of fireflies is obtained by mapping values from the chaotic trajectory generated by the logistic map to the problem search space. The logistic map produces a sequence of values between 0 and 1, which are mapped to the corresponding ranges of the problem variables.
- Attractiveness and Distance: Like the core Firefly Algorithm, EFOIRP employs the attractiveness of fireflies to guide their movement. The attractiveness is determined by the objective function values, where brighter fireflies have higher fitness values. The distance between fireflies also influences their attractiveness, following an inverse relationship.
- Chaotic Movement with Logistic Map: The critical enhancement in EFOIRP lies in incorporating the logistic map to introduce chaotic movement into the firefly population. The logistic map produces chaotic dynamics that enhance the algorithm's ability to explore and exploit effectively. The logistic map is used to update the position of each firefly in the population. The chaotic nature of the logistic map imparts randomness and diversity to the firefly movement, allowing them to explore the search space effectively.
- Light Intensity and Attractiveness Update: In EFOIRP, the light intensity of each firefly represents its fitness value. Fireflies with higher light intensity (fitness) attract other fireflies towards them. The attractiveness between fireflies is updated using the logistic map. The logistic map generates a value between 0 and 1, which is then used to update the attractiveness between fireflies, incorporating the chaotic behaviour into the attraction mechanism.
- Movement and Optimization: Fireflies adjust their positions in the search space based on the updated attractiveness values. The chaotic movement, guided by the logistic map, enables fireflies to explore and exploit the search space efficiently. Fireflies move towards brighter fireflies with higher attractiveness, resulting in a convergence towards optimal solutions.
- Iterative Refinement: EFOIRP iteratively refines the firefly population by repeatedly updating the attractiveness values, adjusting the firefly positions, and evaluating their fitness using the objective function. The chaotic movement introduced by the logistic map ensures a diverse search space exploration, aiding in escaping local optima and finding global optima.

3.4.1. Logistic Map Initialization

In EFOIRP, the initial population of fireflies is determined by utilizing the logistic map. This map, governed by the logistic equation, generates a sequence of values from 0 to 1. These values are subsequently mapped to the corresponding ranges of the problem variables, thus defining the initial positions of the fireflies in the search space. The logistic map is mathematically expressed as Eq.(6).

$$x_{\{n+1\}} = r * x_n * (1 - x_n) \quad (6)$$

Where x_n denotes the value at iteration n. The logistic map operates within the interval [0, 1], providing a continuous range of values. Each iteration of the map calculates the next value, $x_{\{n+1\}}$, based on the current value, x_n, and the parameter r.

To initialize the firefly population in EFOIRP, the logistic map is executed to generate a sequence of values. These values are then mapped to the corresponding ranges of the problem variables. The mapping process ensures that the initial positions of the fireflies fall within the defined search space. The EFOIRP introduces a chaotic dynamic into the firefly population by incorporating the logistic map into the initialization step. The chaotic behaviour enhances the exploration capabilities of the algorithm by introducing randomness and diversity to the initial positions of the fireflies. This diversity aids in exploring a more comprehensive range of the search space, increasing the likelihood of discovering optimal solutions.

The logistic map generates a sequence of values that are then mapped to the problem variables. The resulting values determine the initial positions of the fireflies, thereby defining the starting points for the optimization process. The logistic map's chaotic nature injects randomness into the initialization process, allowing for a diverse distribution of fireflies throughout the search space. EFOIRP sets the stage for subsequent steps in the optimization process through this mathematical integration of the logistic map. By leveraging the chaos generated by the logistic map, EFOIRP enables the firefly population to explore the search space effectively, enhancing the algorithm's ability to locate optimal solutions. Incorporating the logistic map's chaotic dynamics in the initialization step lays the foundation for the subsequent

RESEARCH ARTICLE

stages of the EFOIRP, facilitating efficient optimization without requiring explicit knowledge of the problem domain.

3.4.2. Attractiveness and Distance Calculation

In this phase, the attractiveness between fireflies and the distance between them play a crucial role in guiding their movement. These factors are mathematically defined and updated iteratively to facilitate the optimization process. The attractiveness between two fireflies, i and j, is determined by their respective fitness values and the distance between them. The fitness values, represented by $f(i)$ and $f(j)$, measure their performance in the optimization problem. The attractiveness, $A(i,j)$, is inversely proportional to the distance, $d(i,j)$, between the fireflies and can be mathematically expressed as Eq.(7).

$$A(i,j) = exp\left(-\gamma * d(i,j)\right) \quad (7)$$

Where γ is a scaling parameter that controls the rate of attractiveness decay as the distance increases, the exponential function captures the inverse relationship between attractiveness and distance, ensuring that fireflies closer to each other have higher attractiveness.

Various distance metrics can be employed to calculate the distance between two fireflies in the search space, such as the Euclidean distance, Manhattan distance, or Minkowski distance. Let's consider the Euclidean distance, which is commonly used. The Euclidean distance, $d(i,j)$, between fireflies i and j in a n-dimensional search space is calculated using Eq.(8).

$$d(i,j) = sqrt\left(\Sigma\left(x_i - x_j\right)^2\right) \quad (8)$$

Where x_i and x_j represent the positions of fireflies i and j along each dimension of the search space.

The logistic map introduces chaotic dynamics into the optimization process to update the attractiveness between fireflies. The updated attractiveness, $A'(i.j)$, is obtained by multiplying the initial attractiveness, $A(i,j)$, with a value generated by the logistic map, denoted by $L(i,j)$:

$$A'(i.j) = A(i,j) * L(i,j) \quad (9)$$

Where $L(i,j)$ is a value between 0 and 1 generated by the logistic map equation, the logistic map provides chaotic behaviour that influences the attractiveness update, introducing randomness and diversity into the optimization process.

The updated attractiveness values are then used to guide the movement of fireflies towards brighter fireflies with higher attractiveness. Fireflies adjust their positions based on the updated attractiveness, and the process iterates until convergence to optimal solutions is achieved.

3.4.3. Chaotic Movement with Logistic Map

In this phase, fireflies undergo chaotic movement based on the logistic map, enabling compelling search space exploration. This step involves updating the positions of fireflies using the chaotic dynamics introduced by the logistic map. The logistic map generates chaotic values between 0 and 1, which are used to perturb the current positions of the fireflies. The logistic map equation, $x_{\{n+1\}} = r * x_n * (1 - x_n)$, is employed to update the positions of the fireflies in the search space. The logistic map's chaotic nature ensures randomness and diversity in the movement of fireflies.

To update the position of each firefly, the logistic map generates a value, $L(i)$, between 0 and 1. This value is then used to perturb the current position, $x(i)$, of the firefly, resulting in an updated position, $x'(i)$. The logistic map equation is applied using Eq.(10).

$$x'(i) = L(i) * x(i) \quad (10)$$

Where $L(i)$ denotes the value generated by the logistic map for firefly i, and $x(i)$ represents the present position of firefly i along each dimension of the search space.

The logistic map-based chaotic movement introduced in EFOIRP enables fireflies to explore the search space effectively, promoting a diverse exploration of potential solutions. The randomness inherent in the logistic map ensures that fireflies move in unpredictable patterns, allowing for thorough search space coverage. The chaotic movement based on the logistic map enhances the algorithm's ability to escape local optima and converge towards global optima. The exploration capabilities of the firefly population are enhanced by the inherent chaotic dynamics, ensuring a more exhaustive exploration of the search space and facilitating the discovery of optimal solutions.

The chaotic movement of fireflies in EFOIRP is an iterative process. Fireflies continuously update their positions using the logistic map-based chaotic dynamics, exploring the search space stochastic. This iterative process allows for the gradual refinement of the firefly population and the convergence towards optimal solutions. This integration enables fireflies to explore the search space efficiently and effectively, enhancing the algorithm's ability to find optimal solutions to complex optimization problems.

3.4.4. Light Intensity and Attractiveness Update

In this phase, fireflies' light intensity and attractiveness are updated using the logistic map. These updates are crucial in guiding the fireflies towards brighter individuals and influencing their movement in optimization. A firefly's light

RESEARCH ARTICLE

intensity represents its fitness or objective function value. It indicates the quality of the solution represented by the firefly. The light intensity, denoted as $I(i)$, is updated based on the logistic map-generated value, $L(i)$, and can be expressed mathematically as Eq.(11).

$$I'(i) = L(i) * I(i) \quad (11)$$

Where $I(i)$ represents the current light intensity of firefly i, and $L(i)$ is the value generated by the logistic map equation for firefly i. The logistic map introduces chaos and randomness into the update process, allowing for various changes in light intensity.

The attractiveness between two fireflies, i and j, is determined by their respective light intensities and the distance between them. The attractiveness, $A(i,j)$, can be calculated using the light intensity values, $I(i)$ and $I(j)$, and the distance, $d(i,j)$, between fireflies i and j. An exponential function is a standard method to calculate attractiveness, resulting in Eq.(12).

$$A(i,j) = exp\big(* \gamma * d(i,j)\big) \quad (12)$$

Where γ is a scaling parameter that controls the rate of attractiveness decay as the distance between fireflies increases. The exponential function captures the inverse relationship between attractiveness and distance, ensuring that fireflies closer to each other have higher attractiveness.

The updated attractiveness, $A'(i,j)$, between fireflies i and j is obtained by multiplying the initial attractiveness, $A(i,j)$, with the value generated by the logistic map for firefly i, denoted as $L(i)$:

$$A'(i,j) = A(i,j) * L(i) \quad (13)$$

Where $L(i)$ is a value between 0 and 1 generated by the logistic map equation, the logistic map introduces chaotic dynamics into the attractiveness update process, enhancing diversity and randomness.

The updated light intensity and attractiveness values guide the movement of fireflies in the search space. Fireflies adjust their positions based on the updated attractiveness values, moving towards brighter fireflies with higher attractiveness. This iterative process ensures fireflies converge towards optimal solutions by continuously changing their positions based on the chaos-guided attractiveness mechanism.

3.4.5. Movement and Optimization

In this phase, the fireflies' positions are updated based on their attractiveness and the chaotic dynamics introduced by the logistic map. This step aims to guide the fireflies towards more promising regions in the search space. The updated position of a firefly, $x'(i)$, is determined by considering the attractiveness of neighbouring fireflies and the present position of the firefly. The new position is calculated as a weighted average between the current position, $x(i)$, and the position of the brightest firefly, $x(j)$, influenced by their attractiveness. This can be mathematically represented as Eq.(14).

$$x'(i) = x(i) + \beta * A(i,j) * \big(x(j) - x(i)\big) + L(i) \quad (14)$$

Where β is a step size factor that controls the impact of the attractiveness, $A(i,j)$, on the movement of firefly i. The term $(x(j) - x(i))$ represents the direction vector from firefly i to firefly j, indicating the direction towards the brightest firefly. $L(i)$ is the value generated by the logistic map equation, adding a chaotic element to the movement of firefly i.

The term $\beta * A(i,j) * \big(x(j) - x(i)\big)$ determines the magnitude and direction of the movement towards the brightest firefly. The attractiveness, $A(i,j)$, provides a weight that scales the influence of the direction vector $\big(x(j) - x(i)\big)$. A higher attractiveness value leads to a stronger attraction towards the brighter firefly. The logistic map-generated value, $L(i)$, introduces chaotic dynamics to the firefly's movement. It adds a random perturbation to the updated position, promoting exploration and avoiding convergence to local optima. The chaotic movement provided by $L(i)$ enhances the algorithm's ability to thoroughly explore the search space, improving the chances of finding optimal solutions. The iterative nature of this step ensures that fireflies continuously adjust their positions based on attractiveness and chaotic dynamics. The process iterates until a stopping criterion is met, indicating the algorithm's convergence towards optimal solutions.

3.4.6. Iterative Refinement

In this phase, the fireflies' positions are adjusted using the firefly's inherent movement characteristics and the concept of boundary handling. This step ensures that the fireflies remain within the feasible region of the search space. Boundary handling mechanisms are incorporated to prevent fireflies from moving outside the defined boundaries of the problem. These mechanisms restrict the movement of fireflies towards infeasible regions and maintain the integrity of the search process. To enforce boundary handling, the positions of the fireflies are checked after each movement iteration. If a firefly violates the boundary constraints, its position is adjusted to bring it back within the feasible region. Various techniques can be employed for boundary handling, such as reflection, randomization, or penalty functions, depending on the specific problem requirements.

One common approach is the reflection method, where the position of a firefly is reflected within the boundaries.

RESEARCH ARTICLE

Mathematically, the reflection operation can be expressed as Eq.(15).

$$x(i)_{new} = 2 * b - x(i)_{old} \quad (15)$$

Where $x(i)_{new}$ represents the new position of firefly i after the reflection, $x(i)_{old}$ is the previous position, and b is the boundary value corresponding to the violated boundary constraint.

Another technique involves randomization, where the position of a firefly is randomly reset within the feasible region. This approach adds stochasticity to the movement and allows for exploring different areas within the boundaries. Additionally, penalty functions can assign a penalty value to fireflies violating boundaries. The penalty value is then incorporated into the fitness evaluation, discouraging fireflies from venturing into infeasible regions. Combining boundary-handling mechanisms ensures that fireflies remain within the feasible region throughout the optimization process. By preventing fireflies from moving outside the boundaries, EFOIRP maintains the integrity of the search space exploration and improves the algorithm's convergence towards optimal solutions. The boundary handling techniques utilized in Step 6 are problem-specific and depend on the nature of the optimization problem. It is essential to tailor these mechanisms to accommodate the specific constraints and requirements of the problem.

Input:

- Problem-specific objective function $f(x)$ to be optimized
- Number of fireflies (population size) N
- Maximum number of iterations $\max_{iter}$
- Bounds (lower and upper limits) for each problem variable

Output:

- Optimal solution x_{opt} that minimizes the objective function $f(x)$
- The minimum objective function value f_{opt} achieved by x_{opt}

Procedure:

Step 1: Initialize the firefly population by generating N fireflies using the logistic map and mapping them to the problem variable bounds.

Step 2: Evaluate the fitness value for each firefly based on the objective function $f(x)$.

Step 3: Set the iteration counter iter = 0.

Step 4: While iter<$\max_{iter}$:

a) Using the logistic map, update the attractiveness between fireflies based on their fitness values and distances.

b) Adjust the firefly positions based on the updated attractiveness values, incorporating chaotic movement using the logistic map.

c) Evaluate the fitness value for each firefly using the updated positions.

d) Update the optimal solution x_{opt} and the minimum objective function value f_{opt} based on the best fitness value.

e) Increment the iteration counter iter = iter + 1.

Step 5: Return the optimal solution x_{opt} and the minimum objective function value f_{opt}.

Algorithm 3 EFOIRP

4. SIMULATION SETTINGS

Researchers strive to design and optimize intricate network architectures, and a revolutionary tool has emerged: GNS3 (Graphical Network Simulator-3). This cutting-edge network simulation software takes users on a transformative journey into virtual networks. With its captivating interface and advanced features, GNS3 allows users to unleash their creativity and expertise, crafting complex network topologies that mirror real-world environments.

Table 1 Simulation Settings

Simulation Setting	Value(s)
Node Count	1500
Network Area Size	150m x 225m
Topology	Random Graph
Traffic Pattern	Poisson
Simulation Duration	900 seconds (i.e., 15 minutes)
Deployment Model	Event-Driven
Obstacle Placement	Random
Transmit Energy	0.1 Joules/bit
Receive Energy	0.05 Joules/bit
Idle Energy	1.0 mW
Sleep Energy	0.1 mW
Battery Capacity	2000 mAh
Simulation Environment	GNS-3
Experimental Repetitions	10

RESEARCH ARTICLE

As they delve into the depths of this virtual landscape, network engineers gain invaluable hands-on experience, fine-tuning configurations and experimenting with various protocols. GNS3's ability to seamlessly integrate real network hardware and virtual machines sets it apart. Users harness the power of virtualization, emulating routers, switches, and other devices with unparalleled precision. Engineers can analyze traffic, capture packets, and meticulously analyze network behavior as their virtual networks spring to life. What sets GNS3 apart from the rest is its spirit of innovation. It seamlessly integrates with external virtualization platforms, amplifying its capabilities and expanding the possibilities for network simulation. With automation and orchestration at their fingertips, users can orchestrate network deployments, automate configurations, and unleash the full potential of their networks. Embark on a journey with GNS3, the gateway to a world where network simulations transcend boundaries, fostering creativity, expertise, and the limitless exploration of network architectures. The various simulation settings are shown in Table 1.

5. RESULTS AND DISCUSSION

5.1. Packet Delivery Ratio

The Packet Delivery Ratio Result Graph (i.e., Figure 1) presents the average packet delivery ratios achieved by three routing algorithms: DORA, PSORS, and EFOIRP, as shown in Table 2.

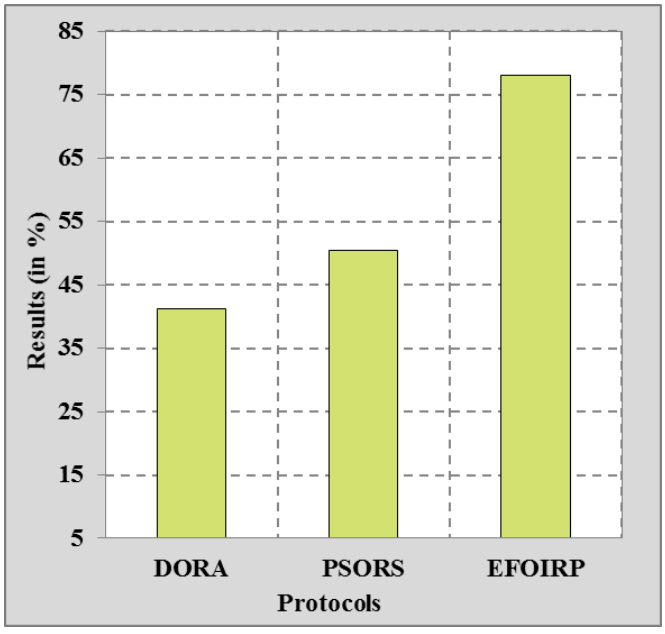

Figure 1 Packet Delivery Ratio Analysis

The average packet delivery ratio for DORA is 41.16%. This means that, on average, DORA successfully delivers approximately 41.16% of the packets in the network. DORA's relatively lower packet delivery ratio suggests there may be challenges in effectively routing packets to their intended destinations, resulting in a higher packet loss rate or unsuccessful delivery. The average packet delivery ratio for PSORS is 50.43%. This indicates that, on average, PSORS achieves a higher rate of successful packet delivery, delivering around 50.43% of the packets in the network. The PSORS helps optimize the path selection process, improving the chances of successful packet delivery and enhancing the overall packet delivery ratio. The EFOIRP routing protocol demonstrates the highest average packet delivery ratio among the three algorithms, with a value of 78.00%. This implies that, on average, EFOIRP successfully delivers approximately 78.00% of the packets in the network. The EFOIRP incorporates the principles of firefly optimization to select routes that maximize packet delivery intelligently. This efficient route selection process contributes to the significantly higher packet delivery ratio achieved by EFOIRP.

Table 2 Packet Delivery Ratio Result Values

Nodes	DORA	PSORS	EFOIRP
150	51.81	58.67	86.02
300	49.79	56.48	83.66
450	47.48	53.69	82.81
600	46.80	53.06	81.29
750	45.12	52.17	78.89
900	39.91	50.31	76.70
1050	35.53	48.33	74.78
1200	34.01	45.68	74.31
1350	31.99	43.82	71.80
1500	29.16	42.02	69.69
Average	41.16	50.43	78.00

The Packet Delivery Ratio Result Graph and Table 2 highlight that EFOIRP achieves the highest average packet delivery ratio, followed by PSORS and DORA. This indicates that EFOIRP's routing mechanism, inspired by firefly optimization, enhances packet delivery efficiency. PSORS demonstrates an average packet delivery ratio, while DORA exhibits a lower delivery ratio, suggesting room for improvement in routing performance.

5.2. Throughput

Figure 2 presents the throughput analysis of three routing algorithms: DORA, PSORS, and EFOIRP. Throughput refers to the amount of data that can be successfully transmitted over

RESEARCH ARTICLE

a network within a given time period. Figure 2 compares the performance of these algorithms in terms of their throughput values.

DORA is a routing algorithm that has been analyzed in the throughput analysis. The average throughput achieved by DORA is 37.52. Although it exhibits a relatively lower throughput than the other two algorithms, it still offers a viable option for data transmission. DORA prioritizes routing based on destination nodes, aiming to establish efficient paths for data transmission. While it may not achieve the highest throughput, its routing strategy can still contribute to effective network communication.

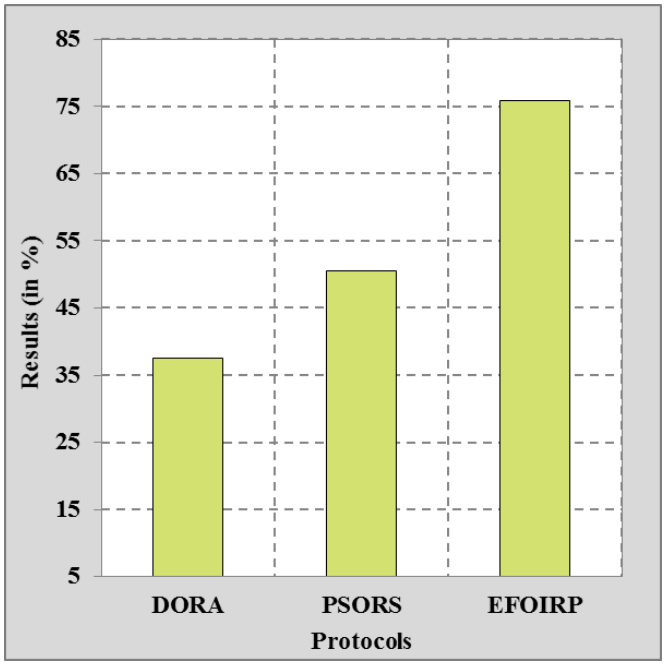

Figure 2 Throughput Analysis

PSORS demonstrates a higher average throughput of 50.48. This algorithm utilizes the concept of particle swarm optimization to optimize routing paths in the network. By simulating the behaviour of particles in a swarm, PSORS dynamically adjusts the routes to enhance data transmission efficiency. The results indicate that PSORS outperforms DORA in throughput, suggesting its potential for improving network performance through intelligent routing optimization.

EFOIRP exhibits the highest average throughput of 75.87 among the three algorithms. Inspired by the behaviour of fireflies, EFOIRP focuses on optimizing routing paths by dynamically adjusting the attractiveness of nodes. This approach allows efficient data transmission by prioritizing beautiful nodes along the routes. The significant difference in average throughput between EFOIRP and the other two algorithms demonstrates the effectiveness of its optimization strategy, making it an appealing choice for networks where high throughput is crucial.

Table 3 Throughput Result Values

Nodes	DORA	PSORS	EFOIRP
150	33.05	46.27	71.68
300	33.49	46.81	72.81
450	34.23	48.36	73.79
600	35.35	48.64	74.53
750	36.02	50.10	74.69
900	36.72	50.91	75.45
1050	40.69	52.68	75.69
1200	41.29	52.78	78.62
1350	41.90	53.81	79.73
1500	42.51	54.51	81.70
Average	37.52	50.48	75.87

The throughput analysis (Table 3) compares three routing algorithms: DORA, PSORS, and EFOIRP. While DORA offers a viable option with a lower average throughput of 37.52, PSORS achieves a higher throughput of 50.48 by leveraging particle swarm optimization. However, EFOIRP surpasses both algorithms with an average throughput of 75.87, showcasing its remarkable performance in optimizing routing paths inspired by the behaviour of fireflies. The choice of routing algorithm ultimately depends on the specific network requirements, with EFOIRP standing out as a highly efficient solution for achieving superior data transmission capacity.

5.3. Packet Delay

Figure 3 compares packet delay performance for three routing algorithms: DORA, PSORS, and EFOIRP. The average packet delay values for each algorithm are provided as follows: DORA has an average packet delay of 12,884.4 ms, PSORS has an average packet delay of 10,703.2 ms, and EFOIRP has the lowest average packet delay of 5,554.6 ms.

The packet delay metric measures the time a packet travels from its source node to its destination node in a network. It is an important performance indicator as it directly affects the quality of service experienced by network users. Lower packet delay values indicate more efficient routing algorithms, as they enable faster delivery of packets, reducing latency and improving real-time communication.

RESEARCH ARTICLE

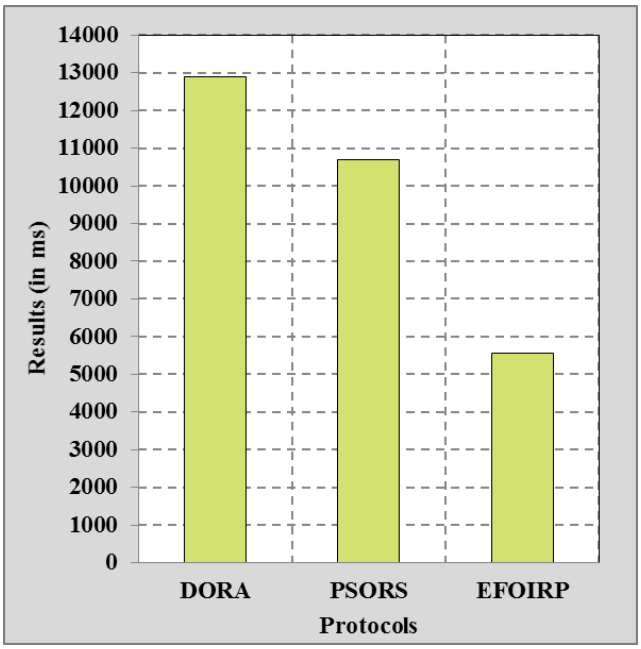

Figure 3 Packet Delay Analysis

Table 4 Packet Delay Result Values

Nodes	DORA	PSORS	EFOIRP
150	12455	9903	3419
300	12490	9966	3434
450	12512	10284	3895
600	12549	10341	4001
750	12767	10492	4548
900	12982	10552	5885
1050	13090	10675	6472
1200	13206	11079	6993
1350	13253	11348	7048
1500	13540	12392	9851
Average	12884.4	10703.2	5554.6

In Figure 3, it is evident that EFOIRP outperforms both DORA and PSORS regarding packet delay. With an average delay of 5,554.6 ms, EFOIRP exhibits the shortest delivery time among the three algorithms. This indicates that EFOIRP optimizes the routing paths effectively, resulting in faster transmission of packets and minimal congestion in the network.

DORA and PSORS show higher average packet delay values compared to EFOIRP. DORA has the highest delay of 12,884.4 ms, indicating that it may suffer from suboptimal routing decisions or inefficient path selection. PSORS performs better than DORA but still has a higher delay than EFOIRP, with an average delay of 10,703.2 ms. These results suggest that PSORS achieves a moderate optimisation level but is still outperformed by the more advanced EFOIRP algorithm.

Figure 3 demonstrates that EFOIRP is the most efficient routing algorithm among the three compared, as it achieves the lowest average packet delay. This indicates that EFOIRP optimizes the routing paths effectively, resulting in faster packet transmission and improved overall network performance. While having higher delays, DORA and PSORS may still be viable options depending on specific network requirements and constraints. However, for applications where low latency and fast packet delivery are crucial, EFOIRP stands out as the superior choice. The comparative analysis of the result is shown in table 4.

5.4. Energy Consumption

Figure 4 illustrates a comparative analysis of energy consumption among three distinct routing algorithms: DORA, PSORS, and EFOIRP. The energy consumption values, expressed as percentages, are in Table 5.

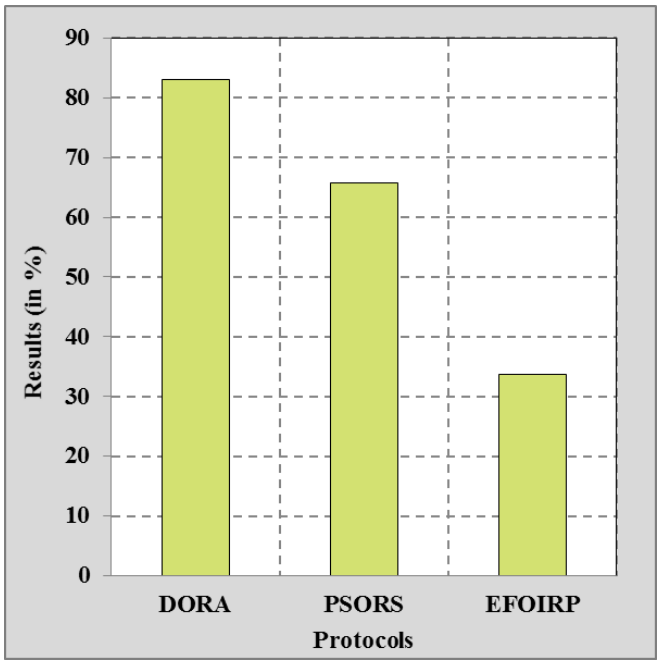

Figure 4 Energy Consumption Analysis

DORA emerges as the routing algorithm with the highest energy consumption among the three, with an average energy consumption of 82.98%. This high energy usage can be attributed to DORA's routing strategy, potentially involving longer paths or additional overhead associated with message

RESEARCH ARTICLE

transmission or processing. Consequently, DORA may not be the optimal choice in scenarios where energy preservation is critical. PSORS exhibits a relatively lower energy consumption, averaging 65.74%. This suggests that PSORS provides more energy-efficient routing solutions compared to DORA. The improved energy efficiency of PSORS can be attributed to its utilization of the particle swarm optimization mechanism, enabling better optimization of routing paths and resource utilization. By harnessing swarm intelligence, PSORS achieves enhanced energy efficiency while maintaining effective communication within the network.

EFOIRP outperforms both DORA and PSORS regarding energy efficiency, demonstrating the lowest energy consumption among the three algorithms, with an average value of 33.78%. EFOIRP adopts a routing protocol inspired by the behavior of fireflies, allowing for dynamic adjustment of routing paths based on the energy status of nodes. This adaptive approach enables EFOIRP to minimize energy consumption by leveraging the most energy-efficient paths available. Consequently, EFOIRP proves to be an optimal choice for energy-constrained scenarios, where energy preservation and the prolongation of network lifetime are paramount.

Figure 4 comprehensively compares energy consumption among the DORA, PSORS, and EFOIRP routing algorithms. DORA exhibits the highest energy consumption at 82.98%, while PSORS demonstrates improved energy efficiency at 65.74%. However, EFOIRP stands out as the most energy-efficient algorithm, with an average energy consumption of 33.78%. These results emphasize the significance of carefully selecting a routing algorithm that aligns with the network's specific energy constraints and requirements.

Table 5 Energy Consumption Result Values

Nodes	DORA	PSORS	EFOIRP
150	74.51	56.39	26.68
300	75.61	57.94	27.90
450	77.90	60.36	31.49
600	81.21	60.98	31.56
750	82.27	61.51	31.59
900	84.47	68.90	33.66
1050	86.63	69.51	34.80
1200	87.80	71.74	38.26
1350	88.75	73.96	38.97
1500	90.63	76.16	42.89
Average	82.98	65.74	33.78

5.5. Network Life Time

In Figure 5, the comparison of network lifetime among the three routing algorithms (i.e., DORA, PSORS, and EFOIRP) reveals valuable insights into their respective performance. Network lifetime refers to the duration a wireless network can operate effectively before the depletion of energy resources. It is critical to consider when designing and implementing wireless network routing protocols.

DORA, with an average network lifetime of 17.19%, falls on the lower end of the spectrum. This algorithm likely employs a destination-oriented approach, where the routing decisions are primarily based on the intended destination of the data packets. While DORA achieves some level of energy efficiency, it is outperformed by both PSORS and EFOIRP regarding network lifetime. This suggests that DORA may not fully optimize energy consumption and fails to use more sophisticated optimization techniques. PSORS demonstrates a significant improvement in network lifetime, achieving an average of 37.41%.

This algorithm incorporates particle swarm optimisation principles, drawing inspiration from particles' collective behaviour in a search space. PSORS likely employs a population of virtual particles representing network nodes communicating and exchanging information to identify optimal routing paths. By leveraging the swarm intelligence of particles, PSORS can effectively balance energy consumption across the network, resulting in a more extended network lifetime than DORA.

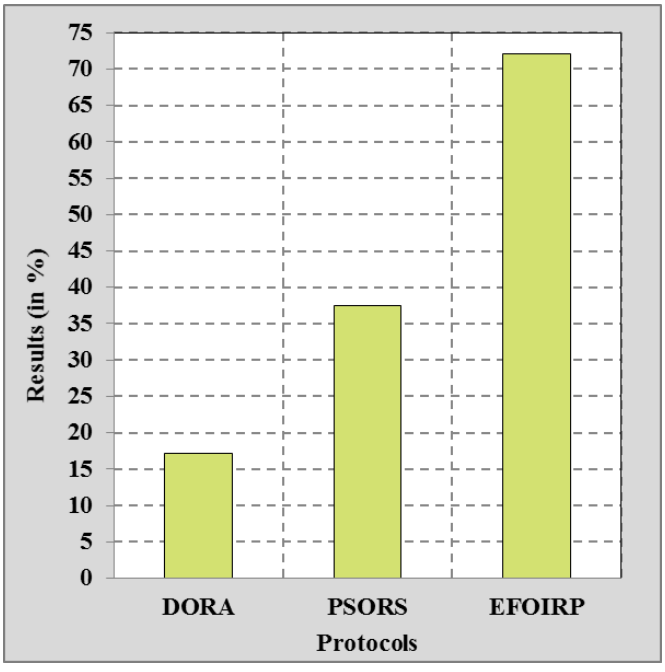

Figure 5 Network Lifetime Analysis

RESEARCH ARTICLE

The most notable performance is observed in EFOIRP, which achieves an average network lifetime of 72.08%. This routing protocol draws inspiration from the energizing behaviour of fireflies. Fireflies exhibit a fascinating phenomenon where they synchronize their flashing patterns to optimize their mating success.

EFOIRP likely incorporates similar principles to dynamically adjust routing paths based on the "illumination" characteristics of virtual fireflies representing network nodes. By adapting the routing paths based on the brightness of neighbouring nodes, EFOIRP can achieve an exceptional level of energy optimization and prolong the network lifetime significantly.

The comparison presented in Figure 5 and Table 6 underscores the importance of utilizing advanced optimization techniques in routing algorithms for wireless networks. While DORA provides a moderate improvement over conventional methods, PSORS and EFOIRP demonstrate the potential of bio-inspired optimization approaches. PSORS significantly extends the network lifetime, and EFOIRP emerges as the most efficient algorithm, achieving the highest network lifetime of 72.08%.

These findings emphasize the significance of exploring nature-inspired algorithms to enhance the energy efficiency and overall performance of wireless networks, ultimately contributing to the sustainability and longevity of such systems.

Table 6 Network Lifetime Result Values

Nodes	DORA	PSORS	EFOIRP
150	24.81	50.53	79.65
300	23.96	49.68	79.36
450	21.13	46.24	74.33
600	18.71	44.56	73.76
750	17.63	37.49	73.24
900	16.73	31.14	70.47
1050	13.43	29.72	69.54
1200	12.77	29.14	66.99
1350	12.22	27.86	66.73
1500	10.51	27.79	66.69
Average	17.19	37.41	72.08

6. CONCLUSION

This paper addressed the impact of physical obstructions on routing performance in IoT-based Cloud Wireless Sensor Networks (IC-WSN) for greenhouse farming. The proposed Energizing Firefly Optimization-Inspired Routing Protocol (EFOIRP) offers a novel approach to overcoming the challenges of physical obstructions. By conducting comprehensive site surveys, characterizing obstructions, and considering signal strength and alternative routing paths, EFOIRP intelligently adapts to obstructions and ensures reliable data transmission. This research aimed to enable seamless data transmission and monitoring in greenhouse farming, ultimately leading to optimal conditions for plant growth and increased crop yields. EFOIRP enables real-time monitoring of environmental parameters by minimising signal interference and disruptions, facilitating precise control over greenhouse conditions and efficient resource management. The research objectives were successfully achieved by developing adaptive routing algorithms, evaluating their performance through simulations or experiments, investigating scalability, and validating effectiveness in real-world greenhouse farming scenarios. The proposed EFOIRP algorithm demonstrates promising results in enhancing the performance of IC-WSN in greenhouse environments. Future work can focus on further optimizing the EFOIRP algorithm and exploring its applicability in other IoT-based environments.

REFERENCES

[1] P. Handayani and N. Folz, "Adaptive Land Management for Climate-Smart Agriculture," in InHeNce 2021 - 2021 IEEE International Conference on Health, Instrumentation and Measurement, and Natural Sciences, 2021. doi: 10.1109/InHeNce52833.2021.9537265.

[2] D. C. Magnaye, "Climate Smart Agriculture Edu-tourism: A Strategy to Sustain Grassroots Pro-biodiversity Entrepreneurship in the Philippines," Advances in Science, Technology and Innovation. pp. 203–218, 2019. doi: 10.1007/978-3-030-10804-5_20.

[3] A. A. Abdalla et al., "Microstructure, chemical compositions, and soft computing models to evaluate the influence of silicon dioxide and calcium oxide on the compressive strength of cement mortar modified with cement kiln dust," Constr. Build. Mater., vol. 341, p. 127668, 2022, doi: 10.1016/j.conbuildmat.2022.127668.

[4] C. Morales-Morales, P. R. Najera-Medina, M. Castro-Bello, J. Morales-Morales, and J. Hernandez-Romano, "Wireless Real-Time Monitoring System Applied in a Tomato Greenhouse," in 2020 IEEE International Autumn Meeting on Power, Electronics and Computing, ROPEC 2020, 2020, pp. 1–6. doi: 10.1109/ROPEC50909.2020.9258762.

[5] C. Guo, Y. Zi, and W. Ren, "A Blockchain Based Framework for Smart Greenhouse Data Management," Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics), vol. 12817 LNAI. pp. 299–310, 2021. doi: 10.1007/978-3-030-82153-1_25.

[6] P. S. Georgantopoulos, D. Papadimitriou, C. Constantinopoulos, T. Manios, I. N. Daliakopoulos, and D. Kosmopoulos, "A Multispectral Dataset for the Detection of Tuta Absoluta and Leveillula Taurica in Tomato Plants," Smart Agric. Technol., vol. 4, p. 100146, 2023, doi: 10.1016/j.atech.2022.100146.

[7] A. Banerjee and D. M. Akbar Hussain, "SD-EAR: Energy aware routing in software defined wireless sensor networks," Appl. Sci., vol.

RESEARCH ARTICLE

8, no. 7, 2018, doi: 10.3390/app8071013.
[8] D. Ardiansyah, A. S. Miftahul Huda, Darusman, R. G. Pratama, and A. P. Putra, "Wireless Sensor Network Server for Smart Agriculture Optimatization," in IOP Conference Series: Materials Science and Engineering, 2019. doi: 10.1088/1757-899X/621/1/012001.
[9] S. Hemavathi and B. Latha, FRHO: Fuzzy rule-based hybrid optimization for optimal cluster head selection and enhancing quality of service in wireless sensor network, vol. 79, no. 11. Springer, 2023, pp. 12238–12265. doi: 10.1007/s11227-023-05106-5.
[10] S. B. Viswanath, T. M. Nagendrappa, and K. R. Venkatesh, "Jsmcrp: Cross-layer architecture based joint-synchronous mac and routing protocol for wireless sensor network," ECTI Trans. Electr. Eng. Electron. Commun., vol. 19, no. 1, pp. 94–113, 2021, doi: 10.37936/ECTI-EEC.2021191.240719.
[11] K. Haseeb, I. Ud Din, A. Almogren, I. Ahmed, and M. Guizani, "Intelligent and secure edge-enabled computing model for sustainable cities using green internet of things," Sustain. Cities Soc., vol. 68, p. 102779, 2021, doi: 10.1016/j.scs.2021.102779.
[12] L. Hong-tan, K. Cui-hua, B. A. Muthu, and C. B. Sivaparthipan, "Big data and ambient intelligence in IoT-based wireless student health monitoring system," Aggression and Violent Behavior. Elsevier Ltd, 2021. doi: 10.1016/j.avb.2021.101601.
[13] K. Murali Krishna, Y. D. Borole, S. Rout, P. Negi, M. Deivakani, and R. Dilip, "Inclusion of Cloud, Blockchain and IoT Based Technologies in Agriculture Sector," in 2021 9th International Conference on Cyber and IT Service Management, CITSM 2021, 2021. doi: 10.1109/CITSM52892.2021.9588894.
[14] S. Namani and B. Gonen, "Smart agriculture based on IoT and cloud computing," in Proceedings - 3rd International Conference on Information and Computer Technologies, ICICT 2020, IEEE, Mar. 2020, pp. 553–556. doi: 10.1109/ICICT50521.2020.00094.
[15] L. L. Hung, F. Y. Leu, K. L. Tsai, and C. Y. Ko, "Energy-efficient cooperative routing scheme for heterogeneous wireless sensor networks," IEEE Access, vol. 8, pp. 56321–56332, 2020, doi: 10.1109/ACCESS.2020.2980877.
[16] X. Fu, Y. Yang, and O. Postolache, "Sustainable multipath routing protocol for multi-sink wireless sensor networks in harsh environments," IEEE Trans. Sustain. Comput., vol. 6, no. 1, pp. 168–181, 2021, doi: 10.1109/TSUSC.2020.2976096.
[17] I. A. A. E. M. And and S. M. Darwish, "Towards Designing a Trusted Routing Scheme in Wireless Sensor Networks: A New Deep Blockchain Approach," IEEE Access, vol. 9, pp. 103822–103834, 2021, doi: 10.1109/ACCESS.2021.3098933.
[18] O. R. Ahutu and H. El-Ocla, "Centralized Routing Protocol for Detecting Wormhole Attacks in Wireless Sensor Networks," IEEE Access, vol. 8, pp. 63270–63282, 2020, doi: 10.1109/ACCESS.2020.2983438.
[19] M. Alotaibi, "Improved Blowfish Algorithm-Based Secure Routing Technique in IoT-Based WSN," IEEE Access, vol. 9, pp. 159187–159197, 2021, doi: 10.1109/ACCESS.2021.3130005.
[20] T. Zhao, L. Wang, K.-W. Chin, and C. Yang, "Routing in Energy Harvesting Wireless Sensor Networks With Dual Alternative Batteries," IEEE Syst. J., vol. 15, no. 3, pp. 3970–3979, 2020, doi: 10.1109/jsyst.2020.3007166.
[21] K. Ramasamy, M. H. Anisi, and A. Jindal, "E2DA: Energy Efficient Data Aggregation and End-to-End Security in 3D Reconfigurable WSN," IEEE Trans. Green Commun. Netw., vol. 6, no. 2, pp. 787–798, 2022, doi: 10.1109/TGCN.2021.3126786.
[22] T. M. Behera, S. K. Mohapatra, U. C. Samal, M. S. Khan, M. Daneshmand, and A. H. Gandomi, "I-SEP: An Improved Routing Protocol for Heterogeneous WSN for IoT-Based Environmental Monitoring," IEEE Internet Things J., vol. 7, no. 1, pp. 710–717, 2020, doi: 10.1109/JIOT.2019.2940988.
[23] C. Chen, L. C. Wang, and C. M. Yu, "D2CRP: A Novel Distributed 2-Hop Cluster Routing Protocol for Wireless Sensor Networks," IEEE Internet Things J., vol. 9, no. 20, pp. 19575–19588, 2022, doi: 10.1109/JIOT.2022.3148106.
[24] F. H. El-Fouly, R. A. Ramadan, and R. A. Ramadan, "E3AF: Energy Efficient Environment-Aware Fusion Based Reliable Routing in Wireless Sensor Networks," IEEE Access, vol. 8, pp. 112145–112159, 2020, doi: 10.1109/ACCESS.2020.3003155.
[25] J. Ramkumar, A. Senthilkumar, M. Lingaraj, R. Karthikeyan, and L. Santhi, "Optimal Approach for Minimizing Delays in Iot-Based Quantum Wireless Sensor Networks Using Nm-Leach Routing Protocol," J. Theor. Appl. Inf. Technol., vol. 102, no. 3, pp. 1099–1111, 2024.
[26] R. Jaganathan, V. Ramasamy, L. Mani, and N. Balakrishnan, "Diligence Eagle Optimization Protocol for Secure Routing (DEOPSR) in Cloud-Based Wireless Sensor Network," Res. Sq., 2022, doi: 10.21203/rs.3.rs-1759040/v1.
[27] J. Ramkumar and R. Vadivel, "Improved frog leap inspired protocol (IFLIP) – for routing in cognitive radio ad hoc networks (CRAHN)," World J. Eng., vol. 15, no. 2, pp. 306–311, 2018, doi: 10.1108/WJE-08-2017-0260.
[28] L. Mani, S. Arumugam, and R. Jaganathan, "Performance Enhancement of Wireless Sensor Network Using Feisty Particle Swarm Optimization Protocol," ACM Int. Conf. Proceeding Ser., pp. 1–5, Dec. 2022, doi: 10.1145/3590837.3590907.
[29] R. Jaganathan and R. Vadivel, "Intelligent Fish Swarm Inspired Protocol (IFSIP) for Dynamic Ideal Routing in Cognitive Radio Ad-Hoc Networks," Int. J. Comput. Digit. Syst., vol. 10, no. 1, pp. 1063–1074, 2021, doi: 10.12785/ijcds/100196.
[30] K. Wang, C. M. Yu, and L. C. Wang, "DORA: A Destination-Oriented Routing Algorithm for Energy-Balanced Wireless Sensor Networks," IEEE Internet Things J., vol. 8, no. 3, pp. 2080–2081, 2021, doi: 10.1109/JIOT.2020.3025039.
[31] G. Tong, S. Zhang, W. Wang, and G. Yang, "A particle swarm optimization routing scheme for wireless sensor networks," CCF Trans. Pervasive Comput. Interact., vol. 5, no. 2, pp. 125–138, 2023, doi: 10.1007/s42486-022-00118-1.

Authors

J. Jerlin Adaikala Sundari, MCA,M.Phil,NET. has been working as an Assistant professor in the Department of Computer Science at PSG College of Arts and Science,Coimbatore since 2016, Her research interests include IOT, data mining and Artificial Intelligence . He has served as an Assistant professor in Computer Science at Sacred Heart College of Arts and Science, Dindigul from 2013-2016.Also She worked as a lecturer in Computer Science at Don Bosco Community College from 2011 to 2013.She has served a lecturer in Computer Science at AVK First Grade College, Bangalore from 2008 to 2010.

Dr. G. Preethi, M.Phil, M.Tech., Ph.D has worked as a Lecturer in Sami Arul College and T.U.K. Arts College, Thanjavur, She worked as an Assistant Professor in SASTRA University, and Annai Vailankanni Arts and Science College, Thanjavur. She is currently working as Associate Professor from the department of Computer Science in PRIST University, Thanjavur. Her research area includes Data Mining, Cloud Computing, Big Data and IoT. She has published various research articles in many journals.

RESEARCH ARTICLE

How to cite this article:

J. Jerlin Adaikala Sundari, G. Preethi, "Energizing Firefly Optimization-Inspired Routing Protocol (EFOIRP) for Performance Enhancement in IOT-Based Cloud Wireless Sensor Networks (IC-WSN)", International Journal of Computer Networks and Applications (IJCNA), 11(2), PP: 140-158, 2024, DOI: 10.22247/ijcna/2024/224441.