International Journal of Computer Networks and Applications (IJCNA)
DOI: 10.22247/ijcna/2025/10 Volume 12, Issue 1, January – February (2025)

RESEARCH ARTICLE

Intelligent Penguin Inspiration Routing Protocol (IPIRP) for Maximizing Energy Efficiency in Internet of Things-Based Cloud Wireless Sensor Networks (IC-WSN)

J. Jerlin Adaikala Sundari
Department of Computer Science, PRIST University, Thanjavur, Tamil Nadu, India.
jerlinsundari@gmail.com

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

Received: 11 December 2024 / Revised: 31 January 2025 / Accepted: 07 February 2025 / Published: 28 February 2025

Abstract – The Intelligent Penguin Inspiration Routing Protocol (IPIRP) is proposed to maximize energy efficiency in Internet of Things-based Cloud Wireless Sensor Networks (IC-WSN). The scalability of routing algorithms becomes challenging when accommodating many sensors while maintaining efficient data transmission. Existing protocols struggle with network expansion, resulting in performance degradation and reduced efficiency. To address this issue, IPIRP introduces innovative routing strategies that scale effectively with the growing number of sensors. This includes hierarchical routing architectures, geographic-based routing algorithms, and load-balancing techniques. By dividing the network into smaller sub-networks or clusters, reducing routing overhead, and dynamically adjusting routing paths based on network conditions, IPIRP enhances scalability, reduces latency, and optimizes data transmission. This research aims to enable seamless network expansion, efficient resource utilization, and improved performance in IC-WSN for various applications, including greenhouse farming. By focusing on scalable routing solutions, IPIRP empowers users to build robust and energy-efficient monitoring systems that provide reliable data for informed decision-making and enhance the overall efficiency of IoT-based networks.

Index Terms – Cloud, Energy Efficiency, Penguin, Internet of Things, Scalability, Wireless Sensor Networks.

1. INTRODUCTION

Greenhouse farming is an advanced agricultural technique designed to optimize crop production by creating controlled environments within enclosed structures such as glass or plastic greenhouses. Unlike traditional farming, which relies on external weather conditions, greenhouse farming enables precise control over essential factors such as temperature, humidity, light, and air quality. By maintaining stable environmental conditions, this approach allows year-round cultivation, ensuring a consistent food supply regardless of seasonal changes or geographic limitations[1].This is particularly crucial in regions with extreme climates or unpredictable weather patterns, where traditional farming may face disruptions. Greenhouses protect crops from droughts, heavy rains, and sudden temperature drops, significantly improving yield reliability. Additionally, by implementing automated climate control systems, farmers can monitor and adjust environmental parameters to suit different crop requirements, reducing plant stress and increasing productivity. This method also enhances food security by providing stable and predictable harvests, reducing dependency on fluctuating weather conditions[2]. With increasing concerns about climate change and land degradation, greenhouse farming offers a promising solution to maximize agricultural efficiency while ensuring sustainability. By integrating technology-driven precision farming techniques, greenhouse agriculture supports the production of high-quality crops while reducing the risks associated with environmental uncertainties. This shift towards controlled-environment agriculture paves the way for sustainable and resilient food systems worldwide[3].

Beyond improving productivity, greenhouse farming plays a vital role in resource conservation and environmental sustainability. Compared to conventional farming, it significantly reduces water consumption by integrating advanced irrigation techniques such as drip irrigation and fogging systems. The greenhouse farming reduces the need for chemical pesticides and herbicides since the enclosed structure limits pest infestations[4]. This leads to healthier,

RESEARCH ARTICLE

chemical-free crops, reducing environmental pollution caused by excessive pesticide use. Another advantage is optimized land use, as greenhouse farming allows for vertical farming and hydroponic systems, enabling higher crop yields per square meter. This approach is particularly beneficial in urban areas where arable land is limited, allowing food production closer to consumption centers, reducing transportation costs and carbon emissions[5]. The ability to grow off-season crops within a controlled environment also increases profitability and reduces dependency on imports. Furthermore, greenhouse farming supports climate resilience by mitigating the impacts of droughts, floods, and extreme weather conditions. As the world faces rising population growth and depleting natural resources, greenhouse farming presents a viable solution for sustainable food production. This modern agricultural technique promotes efficiency, ensuring maximum output with minimal environmental impact while preserving resources for future generations[6].

The Internet of Things-based Cloud Wireless Sensor Network (IC-WSN) represents a cutting-edge integration of wireless sensors and cloud computing, enabling seamless data collection, transmission, and processing for various applications. In an IC-WSN system, sensors are deployed across different environments to monitor key parameters [7]. These sensors continuously collect real-time data, transmitting it wirelessly to a cloud-based platform for storage and analysis. The cloud infrastructure allows for efficient data processing and remote accessibility, ensuring timely responses to changes in environmental conditions. One of the most significant benefits of IC-WSN is scalability, as the cloud can handle vast amounts of sensor-generated data without requiring expensive local storage. Additionally, machine learning and artificial intelligence (AI) algorithms enhance IC-WSN by identifying patterns and predicting future trends, leading to automated decision-making[8], [9]. This technology is widely applied in smart cities, healthcare, industrial automation, and precision agriculture, offering improved efficiency, optimized resource allocation, and intelligent automation. By integrating IC-WSN with existing infrastructure, businesses and organizations gain access to real-time insights, enabling them to make data-driven decisions. This interconnected system fosters intelligent monitoring and remote management, significantly enhancing productivity and operational efficiency across multiple industries[10].

IC-WSN has emerged as a transformative solution for greenhouse farming, addressing the need for continuous monitoring, automation, and data-driven decision-making. In a greenhouse, wireless sensors are strategically placed to measure critical ecological factors [11]. These sensors relay data in real-time to cloud-based platforms, where sophisticated algorithms analyze the information and provide actionable insights. This system enables farmers to detect anomalies, predict trends, and make necessary adjustments remotely, ensuring optimal growing conditions at all times. IC-WSN allows for automated irrigation and climate control, reducing manual intervention and minimizing resource wastage[12]. The ability to monitor greenhouse conditions remotely ensures greater efficiency, reduced operational costs, and improved crop quality. Additionally, predictive analytics help farmers anticipate changes in weather patterns, allowing for proactive adjustments to temperature, ventilation, and water supply. The integration of IC-WSN with artificial intelligence (AI) and machine learning (ML) further enhances greenhouse farming by optimizing resource utilization and preventing crop loss. As agriculture continues to shift towards smart and sustainable practices, IC-WSN plays a crucial role in enhancing productivity and ensuring food security. By leveraging real-time data and automation, IC-WSN empowers farmers to improve efficiency, reduce costs, and maximize crop yields in controlled-environment agriculture[13].

1.1. Problem Statement

Scaling routing algorithms to accommodate a large number of sensors while maintaining efficient data transmission becomes a challenging task. Existing routing protocols may struggle to handle the increasing network size, resulting in degraded performance, longer latency, and reduced overall network efficiency. To address this problem, innovative routing strategies need to be developed that can scale effectively with the growing number of sensors. This could involve designing hierarchical routing architectures that divide the network into smaller sub-networks or clusters, reducing the routing overhead and improving scalability. Alternatively, routing algorithms based on geographic or virtual coordinates can be explored, enabling efficient routing without relying on individual sensor addresses. The load balancing techniques can be incorporated into the routing protocols to distribute the traffic among multiple paths and prevent congestion evenly.

1.2. Motivation

Scaling routing algorithms in IC-WSN for greenhouse farming is essential to accommodate the increasing number of sensors and ensure efficient data transmission. We can achieve several vital motivations for greenhouse farming by addressing this challenge. Firstly, developing innovative routing strategies that scale effectively enables seamless network expansion without sacrificing performance. This ensures that greenhouse farmers can easily add more sensors to their monitoring systems as needed without compromising data transmission quality. Secondly, hierarchical routing architectures and geographic-based routing algorithms offer efficient alternatives to handle large-scale networks, reducing latency and improving overall network efficiency. Additionally, incorporating load-balancing techniques minimizes congestion and maximizes resource utilization, ensuring optimal data transmission across the network. By

RESEARCH ARTICLE

focusing on scalable routing solutions, we empower greenhouse farmers to build robust and scalable IC-WSN systems that provide reliable and timely data, facilitating informed decision-making and improving the efficiency of greenhouse operations.

1.3. Research Objective

This study focuses on overcoming the challenges associated with expanding routing mechanisms in cloud-based wireless sensor networks designed for greenhouse agriculture. The goal is to develop highly efficient solutions that can support a growing number of sensor nodes while maintaining seamless network performance. The specific aims include:

- Examining Current Routing Constraints: Identifying the bottlenecks and inefficiencies in existing network protocols, particularly in handling large-scale deployments, increased sensor density, and stable data transmission.
- Designing Advanced Routing Mechanisms: Developing adaptive communication strategies that can accommodate an expanding sensor network, ensuring minimal delays, reliable data transfer, and enhanced system responsiveness.
- Investigating Optimization Strategies: Exploring hierarchical network structures, intelligent data handling methods, and decentralized routing models to enhance network adaptability, energy conservation, and overall operational efficiency.
- Validating Proposed Techniques: Conducting extensive trials and performance assessments through simulations and real-world experiments, evaluating critical metrics such as data delivery efficiency, transmission latency, and power consumption.
- Assessing Practical Implementation: Evaluating the real-world applicability of the proposed routing models in diverse greenhouse settings, considering environmental variability and system scalability.

By addressing these objectives, this research aims to establish robust, scalable, and energy-conscious routing solutions that improve network longevity, resource allocation, and agricultural productivity in sensor-assisted greenhouse systems.

1.4. Organization of the Paper

The paper is organized as follows: Section 1 provides an introduction to the study, highlighting the significance of IC-WSN in greenhouse farming. Section 2 presents a literature review, discussing existing routing algorithms, their limitations, and the need for scalable and energy-efficient solutions. Section 3 introduces the proposed Intelligent Penguin Inspiration Routing Protocol (IPIRP), detailing its biological inspiration, algorithm, mathematical modeling, and optimization strategies. Section 4 describes the simulation setup, including network configurations, parameter settings, and experimental design using GNS3. Section 5 presents the performance evaluation and results, comparing IPIRP with existing routing protocols (DORA, PSORS) across significant key metrics. Finally, Section 6 concludes the study with findings, the impact of IPIRP in maximizing energy efficiency, and potential future research directions.

2. LITERATURE REVIEW

"Energy Efficient Routing Scheme" [14] leverages the combined power of neural networks and fuzzy logic to make intelligent routing decisions based on node energy levels, network traffic, and distance. By training the neural network using historical data, the scheme learns to make informed routing choices that minimize energy consumption while maintaining acceptable levels of network performance. The fuzzy logic component helps handle uncertainty and vagueness in decision-making. "Improved African Buffalo Optimization-based Routing" [15] optimizes cluster formation and routing decisions to minimize energy consumption. In this, sensor nodes form clusters based on their energy levels and proximity to a cluster head. The ABO algorithm dynamically adjusts the cluster formation process, ensuring balanced energy consumption among nodes. It also optimizes the routing paths by considering transmission distance and energy constraints. The technique effectively balances energy efficiency and network performance by incorporating metaheuristic optimization. It improves the overall lifetime of the WSN by prolonging the network operation through efficient clustering and routing strategies.

"Flamingo Search Algorithm-based Cluster Head Selection" [16] is inspired by the flocking behavior of flamingos and is employed to select cluster heads dynamically. The algorithm mimics the movement patterns of flamingos, allowing sensor nodes to collectively identify optimal cluster heads based on criteria such as energy levels, communication proximity, and network connectivity. It achieves efficient cluster formation, reducing the energy consumption of long-distance data transmission and facilitating localized data aggregation. Cluster heads are selected based on their suitability for leading and coordinating data collection and communication within their respective clusters. "E-Sigma Routing Method" [17] utilizes a combination of energy awareness and statistical analysis to make routing decisions. E-Sigma considers parameters such as node energy levels, link quality, and data traffic to calculate a routing metric. This metric helps in selecting the most suitable next hop for data forwarding. By incorporating E-Sigma into RPL, the enhanced protocol improves network efficiency and reliability. It dynamically adapts to changing network conditions and balances energy

RESEARCH ARTICLE

consumption among sensor nodes. E-Sigma's statistical analysis component enhances the routing decisions' robustness by considering historical data and trends. “Intelligent Fish Swarm Routing” [18] was introduced to enhance network adaptability. Decision-making relied on leader-follower dynamics, where node movements mimicked cooperative fish schooling behavior to optimize packet flow. This approach minimized energy depletion but exhibited unstable performance when nodes experienced sudden mobility surges. An inherent lack of predictive modeling limited its ability to handle traffic spikes, necessitating an improved congestion-aware mechanism for sustained performance.

"The collective behavior of rat colonies inspires modified Rat Swarm Optimization" [22] and utilizes a modified version of the core Rat Swarm Optimization algorithm. In this, sensor nodes emulate the behavior of rats to determine their positions in the network cooperatively. The algorithm incorporates various techniques such as distance measurement, angle of arrival estimation, and trilateration to calculate the location coordinates of each sensor node. It leverages the collective intelligence of the swarm to optimize the localization process, minimizing localization errors and ensuring accurate positioning of sensor nodes. "Energy-Aware and QoS-Based Routing" [19] employs reinforcement learning techniques to optimize routing decisions by considering both energy efficiency and Quality of Service (QoS) requirements. By training a routing agent through trial and error, this protocol selects energy-efficient and QoS-compliant routes for data transmission in IoMT networks. This schema dynamically adapts to changing network conditions, minimizing energy consumption while ensuring reliable and timely delivery of medical data.

"Reinforcement learning-based dynamic routing" [20] determines the best routing paths for data collection using a mobile sink, such as a drone or mobile device. The algorithm learns from environmental interactions, considering factors like node energy levels, data traffic, and network topology to make informed decisions on the sink's movement. This approach adapts to changing network conditions, minimizing energy consumption and maximizing data collection efficiency. It offers improved energy efficiency, enhanced data collection coverage, and reduced communication overhead. It is a valuable solution for WSNs and IoT applications where efficient data collection and resource optimization are crucial. "Improved Gateway-Based Multi-Hop Routing Protocol" [21] employs a gateway-based approach to facilitate multi-hop communication and optimize energy consumption. This protocol considers energy levels and distances to select routing paths that balance energy usage and prevent network partitioning. By considering the heterogeneity of WSNs, it aims to improve the overall network performance.

“Multi-Adaptive Routing Protocol”[22] was formulated to dynamically adjust path selection based on network congestion, node mobility, and residual energy levels. Reinforcement-based decision-making allowed the network to self-optimize under fluctuating conditions. The strategy enhanced data transmission reliability but introduced computational complexity due to continuous learning updates. Security challenges related to route misdirection attacks persisted, requiring additional mechanisms for attack mitigation. While energy-efficient path selection was prioritized, further refinements were necessary to balance network load distribution effectively. “Bio-Inspired Routing Performance” [23]analyzed the performance of bio-inspired routing models focused on swarm intelligence-based decision-making for improved data forwarding in wireless networks. Techniques incorporating ant colony optimization, artificial bee colony, and genetic principles enabled adaptive route selection. The study revealed that while these models exhibited resilience in dynamic conditions, they struggled with latency issues when handling highly dense network topologies. Adaptive clustering mechanisms mitigated congestion but increased routing overhead, necessitating optimization for scalability and computational efficiency.

“Improved Frog Leap Routing” [24]was proposed to optimize data forwarding paths. A leap-based multi-hop routing metric enabled faster convergence to optimal paths, reducing transmission latency. By incorporating mutation-based learning, the protocol refined route selection over time, improving resilience in dynamically changing environments. Despite these advantages, jitter variability remained a challenge, and the model lacked real-time security countermeasures, exposing vulnerabilities to routing disruptions. “Cuckoo Search Routing”[25] introduced Levy flight-based probabilistic exploration for efficient path discovery. By balancing exploration and exploitation, the model improved route stability while minimizing spectrum contention. Adaptability to real-time link failures was limited due to computationally intensive update procedures. The lack of robust misbehavior detection mechanisms exposed routing paths to selective forwarding threats, necessitating an integrated security-aware enhancement to ensure reliability in adversarial network scenarios.

"Dynamic Multi-Hop Energy Efficient Routing Protocol" [26] aims to optimize energy consumption and extend the network lifetime by dynamically selecting efficient multi-hop routes for data transmission. It incorporates various techniques to achieve energy efficiency. It considers factors such as node energy levels, link quality, and data traffic to adapt routing decisions dynamically. By leveraging these parameters, the protocol intelligently selects routes that minimize energy consumption and evenly distribute the energy load among sensor nodes. It allows it to adapt to changing network conditions, such as node failures or energy depletions, by

RESEARCH ARTICLE

rerouting data through alternative paths. This enhances the robustness and reliability of the network. "Survivable path routing" [27] focuses on establishing reliable and resilient communication paths between sensor nodes to ensure continuous operation in the presence of failures or disruptions. This routing scheme aims to maintain connectivity and data transmission capabilities even when nodes or links experience failures due to various factors such as node mobility, environmental conditions, or malicious attacks. By dynamically selecting alternate paths and utilizing redundancy, survivable path routing enhances the fault tolerance and reliability of WSNs in IoT applications.

"Destination-Oriented Routing Algorithm (DORA)" [28] is designed to ensure balanced energy consumption across sensor nodes, thereby extending the operational lifespan of the network. Unlike conventional routing strategies that prioritize shortest paths or minimal hop counts, this approach emphasizes energy-aware decision-making. It dynamically selects the most efficient relay node by evaluating both the available energy of each sensor and its proximity to the final destination. To maintain uniform energy utilization, DORA integrates an adaptive load distribution strategy, which prevents excessive energy drain on specific nodes. By evenly spreading data transmission tasks across the network, this mechanism reduces node failures caused by energy exhaustion, leading to enhanced network stability and prolonged functionality. Through its intelligent selection process, DORA optimizes data forwarding, ensuring efficient resource usage and improved energy sustainability within wireless sensor networks.

"Particle Swarm Optimization Routing Scheme (PSORS)" [29]enhances network routing efficiency by utilizing the collective intelligence of particles. Drawing inspiration from the swarming behavior observed in nature, this method dynamically modifies routing pathways to conserve energy and improve network performance. Each wireless sensor node is modeled as a particle, which navigates the search space by adjusting its velocity and position based on both local interactions and global awareness. Through repeated position updates, particles collaboratively determine optimal data transmission routes, reducing energy consumption while maximizing communication efficiency.

The approach integrates both exploratory and exploitative mechanisms. During the exploration phase, particles move randomly across the search domain to identify potential routing solutions. In contrast, the exploitation phase fine-tunes these paths based on their evaluated efficiency. A fitness function assesses each route by considering key parameters such as power consumption, signal integrity, and network stability. Based on this evaluation, particles adjust their movements to adapt to dynamic network variations, ensuring optimized and energy-conscious routing decisions. Through this self-adaptive mechanism, PSORS continuously refines data transmission paths, making it a highly effective strategy for sustained performance in wireless sensor networks.

3. INTELLIGENT PENGUIN INSPIRATION ROUTING PROTOCOL (IPIRP)

3.1. Motivation of IPIRP

The Intelligent Penguin ($Int - Pen$), also called as emperor penguin scientifically known as Aptenodytes forsteri, is a remarkable species that stands out among penguins due to its height and weight. Both male and female Int-Pens exhibit similar physical characteristics in size and feathering. Their distinctive coloration includes a black back and head, a white belly, pale yellow breasts, and brilliant yellow ear patches. These features contribute to their ability to blend into their icy surroundings. Emperor penguins are well adapted to the harsh Antarctic environment, where they reside and breed on open ice throughout the year.

During the mating season in the austral winter, thousands of Int-Pen congregate on land to form massive breeding colonies known as rookeries. Female Int-Pen undertakes long foraging journeys, swimming up to 50 kilometers in search of prey in the ocean. This behavior allows them to sustain themselves and provide nourishment for their offspring. The Int-Pen's cooperative foraging and hunting strategies exemplify their social nature.

One of the remarkable feats of the Int-Pen is its diving ability. They have been observed diving to extraordinary depths of up to 1,900 feet (580 meters) and can remain submerged for nearly 25 minutes. These prolonged dives enable them to access food sources beneath the icy Antarctic waters. However, such impressive diving capabilities come at a cost. Over time, the wings of Int-Pen have evolved to become rigid and flattened, rendering them flightless. This adaptation is well-suited for their predominantly aquatic lifestyle, as they use their wings as efficient flippers to navigate the water.

During the extreme Antarctic winter, Int-Pen face frigid temperatures and harsh winds. They engage in a unique behavior known as huddling to combat these challenges. Huddling involves gathering penguins in tightly packed groups, where they rely on each other's body heat for warmth and survival. This huddling behavior progresses through four distinct stages.

- The first stage involves the creation and evaluation of huddle boundaries. Int-Pen work collectively to establish the limits of the huddle, ensuring that it remains compact and efficient in retaining heat. This evaluation process helps optimize the arrangement of individuals within the huddle.

- In the second stage, the penguins assess temperature variations within the huddle. They actively seek out warm and cold spots, redistributing themselves to achieve an even heat distribution throughout the group. This thermoregulation ensures that no penguin is subjected to extreme temperatures.
- The third stage revolves around determining the spacing between individuals within the huddle. Int-Pen gauge the distance between themselves and their neighboring companions, arranging themselves at appropriate intervals. By optimizing spacing, they maximize heat conservation while minimizing heat loss.
- The final stage involves identifying and relocating influential movers within the huddle. Specific individuals within the group generate more body heat than others, and their strategic positioning can benefit the entire huddle. Through subtle movements and adjustments, the penguins ensure that these influential movers are appropriately positioned, allowing the heat to be evenly shared among all individuals.

Maintaining equal time sharing within the cosy huddle is critical to this behavior. To achieve this, each Int-Pen's position $(\hat{P}, \hat{Q})$ within the huddle can be updated by moving a variable distance, depending on its proximity to the locations of other penguins. This continual adjustment allows optimal heat distribution and ensures the group's survival in harsh Antarctic conditions.

3.2. Algorithm for Optimization and Mathematical Model

This research delves into the mathematical modelling of the huddling behavior exhibited by Int-Pen. The foundation of this approach lies in formulating mathematical equations and algorithms to describe and analyze the various aspects of the huddle. Let's explore the mathematical procedures involved in this modelling process.

- Huddle Border Creation: The huddle border was created by chance among the Int-Pen. This can be represented mathematically by randomly assigning positions to the penguins within a defined region or by using a stochastic process to determine the initial huddle boundaries.
- Heat Map Calculation: A heat map of the surrounding area is calculated to optimize the distribution of warmth within and around the huddle. This involves analyzing the temperature variations and gradients near the penguin group. Mathematical techniques such as interpolation or spatial analysis can be employed to generate the heat map.
- Average Distance Determination: The average distance between Int-Pen within the huddle is essential for optimizing heat retention. This can be computed mathematically by measuring the distances between each pair of penguins within the huddle and taking the average of these distances. Techniques from graph theory or distance metrics can be utilized for this calculation.
- Effective Mover Identification: The identification of the effective mover, i.e., the penguin generating the most body heat or contributing significantly to the collective warmth, involves mathematical analysis. This can be achieved by evaluating the heat output of each penguin based on factors such as body size, metabolic rate, and position within the huddle. Mathematical optimization techniques can be employed to identify the effective mover, such as maximizing a heat function or solving an optimization problem.
- Huddle Perimeter Recalculation: The huddle's perimeter is recalculated using the information obtained from the effective mover. This involves adjusting the positions of the Int-Pen within the huddle based on the new locations of the effective mover. Mathematical algorithms, such as geometric transformations or iterative optimization methods, can update the huddle boundaries and redistribute the penguins accordingly.

3.2.1. Identify and Create the Huddle Border

In identifying and creating the huddle border, Int-Pen exhibit a characteristic behavior of congregating along the edges of a polygon-shaped structure. Within the huddle, each Int-Pen randomly selects a set of neighboring individuals to establish social connections. The formation of the huddle border around the polygon is determined by evaluating the wind flow patterns in the surrounding environment. However, it should be noted that the wind speed typically exceeds the movement capabilities of an individual Int-Pen. The random and complex nature of huddle border formation among Int-Pen can be described using sophisticated mathematical concepts.

To quantify the wind speed denoted as Φ, the complex potential function Ψ gradient is calculated. The complex potential function Ψ represents the wind flow properties and is obtained through the vector operation of taking the gradient of Φ. The same is mathematically expressed as Eq.(1)

$$\Psi = \nabla\Phi \tag{1}$$

In Eq.(2)., the complex potential function G is derived by combining the wind speed vector Φ with the vector Ω, representing the flow characteristics within the huddle. This combination is achieved by multiplying the wind speed vector Φ by a fictitious variable s:

$$G = \Phi + s\Omega \tag{2}$$

Where variable s acts as a parameter determining the influence of the flow vector Ω on the wind speed. The resulting function G is an analytical representation of the polygon-shaped huddle region in a two-dimensional setting. It captures the complex interplay between the wind flow and the Int-Pen' movements within the huddle.

The visualization of Eq.(2) in a two-dimensional context provides insights into the spatial arrangement of Int-Pen within the huddle. Within this illustration, each Int-Pen can randomly adjust its position based on the combination of the wind speed and the flow vector Ω. Through these random adjustments, the Int-Pen gradually converge toward the ideal location, corresponding to the centre of the Z-shaped polygon area exhibiting the highest effective fitness rate.

The sophisticated mathematical concepts in this modelling process allow a quantitative understanding of the huddle border formation among Int-Pen. By considering the wind flow patterns and the collective behavior of the penguins, researchers can gain valuable insights into the dynamics and optimization of huddling behaviors in these remarkable Antarctic creatures.

3.2.2. Temperature Distribution Around the Group

The huddling behavior of Int-Pen contributes to energy conservation and increases the ambient temperature within the huddle. To mathematically model this scenario, we can create a mathematical representation that considers the temperature profile based on the radius of the polygon-shaped huddle. Assuming that the temperature is $F = 0$ when the radius of the polygon (B) is more significant than one and $F = 1$ when the radius is less than 1, we can define the temperature profile using Eq.(3).

$$F = \begin{cases} 0, & if\ B > 1 \\ 1, & if\ B < 1 \end{cases} \tag{3}$$

Where F represents the temperature at a given point, and it takes on the value 0 if the polygon's radius is greater than 1, indicating a lower temperature. Conversely, if the radius is less than 1, F is set to 1, representing a higher temperature. This mathematical model allows us to differentiate between distinct temperature states based on the huddle size.

As defined in Eq.(3), the temperature profile significantly drives Int-Pen' exploration and exploitation behavior across various sites. It is a guiding factor that influences their decision-making process and directs them towards areas with the most favorable conditions or resources. Furthermore, the computer-generated temperature distribution around the group, denoted as F', can be expressed using Eq.(4).

$$F' = \left(F - \frac{Max_{iteration}}{p - Max_{iteration}}\right) \tag{4}$$

Where F represents the temperature defined by Eq.(3) and is either 0 or 1 based on the value of B, and the variable p represents the current iteration in the search process. At the same time, $Max_{iteration}$ denotes the maximum number of iterations. The temperature distribution F' is calculated by considering the ratio of the current iteration to the maximum iteration value. This distribution quantitatively represents how the temperature changes over time during the search process.

3.2.3. Distance Calculation

Once the huddle border has been established, the proximity of an Int-Pen to an optimal solution can be quantified by considering its fitness value, with values closer to 1 indicating a more optimal solution. In response, the other search agents, also known as Int-Pen, adjust their positions to align with the mathematically determined optimal solution. The distance between an Int-Pen and the search agent with the lowest fitness score, representing the fittest Int-Pen, can be calculated using Eq.(5).

$$\vec{Y}_{hm} = Dve\left(E(\vec{D}).M(\vec{p}) - \vec{U}.\vec{M}_{hm}(p)\right) \tag{5}$$

Where p represents the current iteration, and $\vec{Y}_{hm}$ represents the distance in meters between the Int-Pen and the fittest search agent. To avoid collisions with neighboring Int-Pen or obstacles, the vectors $\vec{D}$ and $\vec{U}$ are employed. $\vec{M}$ represents the most optimal solution, or the position vector of the fittest Int-Pen, while $\vec{M}_{hm}$ represents the position vector of the Int-Pen under consideration. The social factors guiding Int-Pen toward the most optimal search agent are defined by function E ( ).The vectors $\vec{D}$ and $\vec{U}$ can be calculated using Eq.(6) to Eq.(8).

$$\vec{D} = \left(C \times \left(F' + M_{grid}(Accuracy)\right) \times Rand\ () - F'\right) \tag{6}$$

$$M_{grid}(Accuracy) = Dve\left(\vec{C} - \vec{C}_{hm}\right) \tag{7}$$

$$\vec{U} = Rand\ () \tag{8}$$

The movement parameter C ensures a safe distance between search agents to prevent collisions. The range of temperatures surrounding the huddle is determined by F', with parameter C set to 2. By contrasting the differences between Int-Pen and a random function, $Rand(\ )$, within the range of [0, 1], the term M_{grid} (Accuracy) determines the accuracy of the polygon grid. The function $E(\ )$ is calculated as follows:

$$E(\vec{D}) = \left(\sqrt{g.h^{-p/z}} - h^{-p}\right)^2 \tag{9}$$

Eq.(9) defines the function h within the expression. Control parameters g and z are used to balance exploration and exploitation. The suggested ranges for g and z are [2.2, 3.7]

RESEARCH ARTICLE

and [1.1, 2.4], respectively. It is worth noting that this method has shown promising outcomes within these specified parameter ranges.

3.2.4. Transfer the Mover

Eq.(10) determines the next position of an Int-Pen in the search area.

$$\vec{M}_{hm}(p+1) = \vec{M}(p) - \vec{D}.Y_{hm} \tag{10}$$

Where $\vec{M}_{hm}(p+1)$ represents the new position of the Int-Pen in the next iteration $(p+1)$, $\vec{M}(p)$ denotes the current position of the Int-Pen, $\vec{D}$ is a vector, and Y_{hm}represents the distance between the Int-Pen and the fittest search agent. By subtracting the product of $\vec{D}$ and $Y_{hm}$from the current position, the Int-Pen determines its new location. It is anticipated that the Int-Pen will be located at the coordinates $\vec{M}_{hm}(p+1)$ in future iterations. After the mover's relocation, the Int-Pen' huddling behavior is recalculated throughout the iteration phase. The proposed IPIRP incorporates several intriguing aspects:

- Memorizing Optimal Solutions: The algorithm keeps track of the optimal solutions discovered during the iterative process. This allows the Int-Pen to remember and utilize previously found promising solutions, leading to potentially faster convergence towards the global optimum.
- Polygon Grid Technique: The suggested polygon grid technique can create a grid in higher-dimensional search spaces. This grid structure aids in organizing and exploring the search area more effectively, enabling efficient movement and placement of the Int-Pen.
- Balancing Randomness and Collision Avoidance: The vectors $\vec{D}$ and $\vec{U}$ play a crucial role in promoting randomness and avoiding collisions among search agents. By incorporating these vectors into the movement calculations, potential solutions are encouraged to explore the search area more randomly while ensuring safe distances between the Int-Pen.
- Pinpointing Potential Nesting Sites: The suggested distance approach assists in identifying potential nesting sites for Int-Pen. By considering the distances between the Int-Pen and the fittest search agent, the algorithm can guide the penguins towards locations that exhibit favorable fitness values, indicating potential optimal solutions.
- Enhanced Exploration and Exploitation: Unlike standard optimization algorithms, which often exhibit trade-offs between exploration and exploitation, the modified values of vectors $\vec{D}$ and $\vec{U}$ in the IPIRP algorithm allow for improved exploration and exploitation potential simultaneously. This balanced approach enhances the algorithm's ability to explore the search space effectively while exploiting promising solutions, leading to potentially better convergence rates.

Algorithm 1 provides the overall working of IPIRP

Input:

- Maximum number of iterations ($Max_{iterations}$)
- Population size (N)
- Search space boundaries

Output:

- Optimal solution

Procedure:

Step 1: Initialize:

- Generate an initial population of N Int-Pen randomly within the search space boundaries.
- Set the iteration count (p) to 0.

Step 2: Calculate the fitness of each Int-Pen in the population based on their positions.

Step 3: Repeat the until the maximum number of iterations ($Max_{iterations}$) is reached:

- Increment the iteration count (p) by 1.
- Update the huddle border based on the positions of the Int-Pen.
- Calculate the fitness of each Int-Pen in the population.

Step 4: For each Int-Pen (i) in the population:

- Select at least two neighbouring Int-Pen as social references.
- Calculate the distance (Y_{hm})between the Int-Pen (i) and the fittest search agent.
- Update the position of the Int-Pen using Eq.(13)
- where $M_i(p+1)$ is the new position of the Int-Pen (i), $\vec{D}_i$is a vector, and Y_{hm} is the distance.

Step 5: Return the best solution among the Int-Pen based on their fitness scores.

Algorithm 1 IPIRP

3.3. Complexity in Computation

This section analyzes the time and space requirements of the proposed IPIRP. Below is a breakdown of the algorithm's needs in terms of time and space:

RESEARCH ARTICLE

3.3.1. Time complexity

The time complexity of the IPIRP depends on various factors, including the population size (N), the complexity of the fitness function (f), and the number of iterations (Max_iterations). The overall time complexity can be expressed as $O\left(N + Max_{iteration} * (N * f)\right)$. This considers the time required for initialization, fitness calculation, and the iterative steps of updating the huddle border, recalculating fitness, and adjusting Int-Pen positions.

- Initialization: The time complexity of the initialization step is directly dependent on the population size (N). N random positions can be generated within the search space boundaries in $O(N)$ time. This involves generating random numbers and mapping them to the search space dimensions.
- Fitness Calculation: The time complexity of calculating the fitness for each Int-Pen in the population depends on the complexity of the fitness function. Let's assume the fitness function has a time complexity of $O(f)$, where f represents the operations involved in evaluating fitness based on the Int-Pen's position.
- Iterations: The number of iterations (Max_iterations) determines the number of times the algorithm repeats the iterative steps. Within each iteration, the following operations are performed:
  - Huddle Border Update: Calculating the huddle border involves considering the Int-Pen' positions and determining the huddle's boundaries. This step typically requires comparing the positions and evaluating the proximity of Int-Pen, which can be done in $O(N)$ time.
  - Int-Pen Movement: Each Int-Pen adjusts its position based on the $\vec{D}$ and Y_{hm} calculations. This operation has a time complexity of $O(1)$ for each Int-Pen.

3.3.2. Space Complexity

The space complexity of the IPIRP primarily depends on the storage requirements for the population of Int-Pen, as well as any additional variables or data structures used during the algorithm's execution. The space complexity is typically $O(N)$, where N represents the population size. Additional space may be required for the fitness calculation, temporary variables, and any data structures specific to the algorithm's implementation.

- Population: The population's space complexity depends on the population size (N). Storing the positions of N Int-Pen requires $O(N)$ space. Additionally, if other attributes or variables associated with each Int-Pen are stored, their space requirements must also be considered.
- Fitness Calculation: The space complexity for calculating fitness depends on the storage requirements of the fitness function. This typically includes the memory needed for intermediate calculations, data structures, and variables specific to the fitness evaluation. The specific implementation of the fitness function generally determines the space complexity for this step.
- Iterations: The space complexity during iterations primarily depends on the temporary variables and data structures used within each iteration. This includes variables for position updates, fitness evaluations, and additional calculations performed during iteration.

4. SIMULATION SETTINGS

Table 1 Simulation Settings

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

In network exploration, an incredible tool emerges, reshaping the landscape of network simulation: GNS3 (Graphical Network Simulator-3). Like an artist's brush, GNS3 empowers network enthusiasts to craft intricate digital landscapes, delving into the depths of virtual networks with boundless creativity. Within GNS3's realm, users become architects of their virtual domains, weaving complex network topologies with meticulous precision and ingenuity. It

RESEARCH ARTICLE

breathes life into virtual routers, switches, and devices, offering a canvas for network configuration and protocol experimentation. Immersed in the tapestry of GNS3's virtual world, users uncover the secrets of network behavior, analyzing traffic patterns and capturing data with a keen eye for detail. They gain profound insights into the complexities of network operations, unveiling hidden connections and enhancing their expertise. GNS3's versatility transcends traditional limits, seamlessly integrating with external platforms to unlock new dimensions of network simulation. Users harness the power of automation, orchestrating deployments and configurations with seamless coordination. Step into the realm of GNS3, where imagination and technical prowess intertwine, allowing network enthusiasts to rewrite the narrative of network simulation. They embark on a remarkable journey, exploring uncharted territories and discovering innovative solutions in the vast realm of virtual networks. Table 1 provide the setting used to simulate the proposed protocol against the state-of-the-art.

5. RESULTS AND DISCUSSION

5.1. Packet Delivery Ratio

Figure 1 compares the Packet Delivery Ratios (PDR) achieved by three routing algorithms: DORA, PSORS, and IPIRP. The average PDR values obtained for each algorithm are DORA with 41.16%, PSORS with 50.43%, and IPIRP with an impressive 87.10%. Table 2 provides the corresponding result values of packet delivery ratio metric.

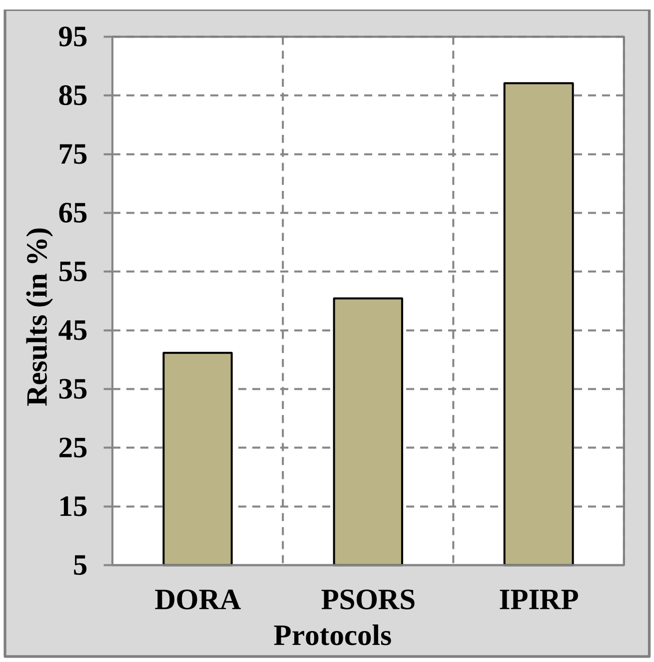

Figure 1 Packet Delivery Ratio Analysis

The results show that DORA achieved the lowest average PDR among the three algorithms, with a value of 41.16%. This indicates that DORA had difficulties delivering packets to their intended destinations. It suggests that the routing decisions made by DORA might not have been optimal, leading to a higher packet loss rate. Further analysis is required to identify the reasons for DORA's lower PDR. PSORS performed better than DORA, with an average PDR of 50.43%. This indicates that PSORS delivered a higher percentage of packets to their destinations than DORA. PSORS leverages the particle swarm optimization technique, which optimizes the routing decisions based on the collective behavior of particles. The results suggest that PSORS' optimization approach improved packet delivery, although there is room for further enhancement.

The highest average PDR of 87.10% was achieved by IPIRP, making it the most effective routing algorithm in terms of packet delivery. IPIRP is inspired by the intelligent behavior of penguins, leveraging their collective decision-making strategies. The significant difference in performance between IPIRP and the other two algorithms indicates the effectiveness of the penguin-inspired approach in routing packets efficiently. The high PDR value suggests that IPIRP delivered most packets to their intended destinations successfully.

Table 2 Packet Delivery Ratio Result Values

Nodes	DORA	PSORS	IPIRP
150	49.79	56.48	94.30
300	51.81	58.67	94.83
450	46.80	53.06	89.66
600	47.48	53.69	91.78
750	39.91	50.31	86.10
900	45.12	52.17	87.82
1050	34.01	45.68	83.09
1200	35.53	48.33	84.16
1350	29.16	42.02	78.77
1500	31.99	43.82	80.48
Average	41.16	50.43	87.10

Figure 1 compares the Packet Delivery Ratios achieved by DORA, PSORS, and IPIRP. DORA had the lowest average PDR, indicating room for improvement in its routing decisions. PSORS performed better than DORA but still had a relatively lower PDR. In contrast, IPIRP demonstrated the highest average PDR, indicating its effectiveness in delivering

RESEARCH ARTICLE

packets to their destinations. These results highlight the varying performances of different routing algorithms and emphasize the success of the penguin-inspired approach in achieving a high packet delivery ratio.

5.2. Throughput

Figure 2 compares throughput among three routing algorithms: DORA, PSORS, and IPIRP. Throughput refers to the amount of data or information transmitted successfully over a network within a given time frame. Table 2 provides the corresponding result values of packet delivery ratio metric. Table 3 provides the corresponding result values of throughput metric.

According to Figure 2, the average throughput achieved by DORA is 37.52%. DORA is a routing algorithm that focuses on optimizing routing decisions based on the destination of the data packets. The relatively lower throughput achieved by DORA suggests that it may have limitations in efficiently delivering data packets to their intended destinations. This could be due to suboptimal routing decisions or inefficient resource allocation. The PSORS routing scheme demonstrates a higher average throughput of 50.48%. PSORS employs a particle swarm optimization approach, which mimics the behavior of a swarm of particles to find optimal routing paths. The improved throughput achieved by PSORS indicates that this scheme is more effective in selecting efficient routes for data packets, resulting in a higher successful transmission rate. This may be attributed to the intelligent nature of the particle swarm optimization technique, which enables the algorithm to optimize routing decisions based on network conditions adaptively.

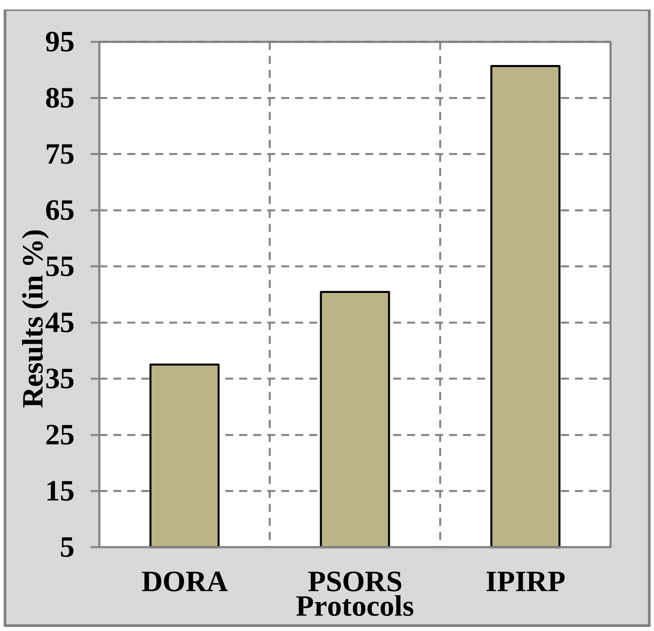

Figure 2 Throughput Analysis

IPIRP, the Intelligent Penguin Inspiration Routing Protocol, achieves the highest average throughput of 90.71%. IPIRP is a routing protocol inspired by the efficient communication and collaboration observed among penguins in their natural habitats. This protocol leverages principles from penguin behavior to create an intelligent routing mechanism. The significantly higher throughput achieved by IPIRP suggests it excels in selecting optimal routing paths and efficiently delivering data packets. The inspiration drawn from penguin behavior likely enables IPIRP to adapt to changing network conditions and dynamically optimize routing decisions.

Figure 2 compares the average throughput of three routing algorithms: DORA, PSORS, and IPIRP. DORA exhibits the lowest throughput at 37.52%, indicating potential inefficiencies in its routing decisions. PSORS demonstrates a moderate improvement with an average throughput of 50.48%. However, the highest throughput of 90.71% is achieved by IPIRP, which draws inspiration from the efficient communication observed among penguins. The results highlight the significance of intelligent routing mechanisms in achieving higher throughput and efficient data packet delivery in network environments.

Table 3 Throughput Result Values

Nodes	DORA	PSORS	IPIRP
150	33.49	46.81	85.44
300	33.05	46.27	84.78
450	35.35	48.64	86.24
600	34.23	48.36	85.50
750	36.72	50.91	90.76
900	36.02	50.10	89.38
1050	41.29	52.78	95.72
1200	40.69	52.68	93.67
1350	42.51	54.51	98.43
1500	41.90	53.81	97.17
Average	37.52	50.48	90.71

5.3. Packet Delay

Figure 3 compares packet delay across three routing algorithms: DORA, PSORS, and IPIRP. Figure 3 reveals the average packet delay experienced by each algorithm, with DORA recording an average delay of 12884.4ms, PSORS exhibiting an average delay of 10703.2ms, and IPIRP

RESEARCH ARTICLE

demonstrating the lowest average delay at 2041.4ms. The findings from Figure 3 are provided in Table 4 and it underscore the superior performance of IPIRP in terms of packet delay reduction. With an average delay of just 2041.4ms, IPIRP proves highly effective in ensuring swift packet delivery within the network. This outcome suggests that IPIRP excels in selecting optimal routes and minimizing delays compared to DORA and PSORS.

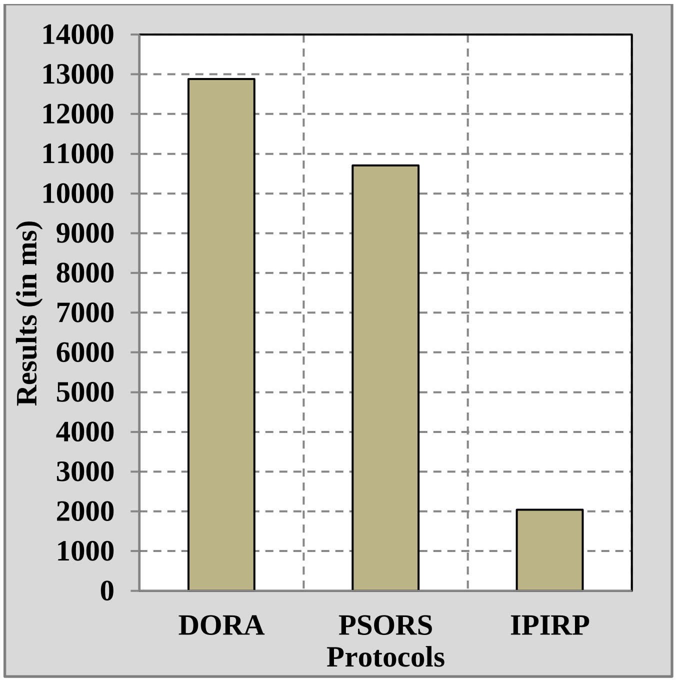

Figure 3 Packet Delay Analysis

DORA exhibits the highest average delay of 12884.4ms, implying that this routing algorithm struggles to efficiently navigate packets through the network, leading to considerable transmission delays. Such prolonged delays may significantly impact real-time applications or services that rely on low-latency communication. PSORS performs better than DORA but still falls short when compared to IPIRP regarding average packet delay. With an average delay of 10703.2ms, PSORS resides between DORA and IPIRP in terms of performance. Although it presents an improvement over DORA, PSORS does not match the efficiency of IPIRP in mitigating packet delay.

The packet delay comparison depicted in Figure 3 highlights the outstanding performance of IPIRP as a routing algorithm when it comes to minimizing delays. With an average delay of 2041.4ms, IPIRP demonstrates its capacity to deliver packets more efficiently and promptly within the network. Conversely, DORA and PSORS exhibit higher delays, with DORA registering the highest delay among the three algorithms. These outcomes emphasize the criticality of selecting an appropriate routing algorithm based on the specific requirements and priorities of the network to ensure optimal performance. Table 4 details the simulation result of the protocols obtained for the metric Packet Delay.

Table 4 Packet Delay Result Values

Nodes	DORA	PSORS	IPIRP
150	12490	9966	1403
300	12455	9903	110
450	12549	10341	1448
600	12512	10284	1423
750	12982	10552	2310
900	12767	10492	2053
1050	13206	11079	2892
1200	13090	10675	2489
1350	13540	12392	3201
1500	13253	11348	3085
Average	12884.4	10703.2	2041.4

5.4. Energy Consumption

Figure 4 compares energy consumption among three routing algorithms: DORA, PSORS, and IPIRP. Table 5 represents the average energy consumption for each routing protocol. Average values of each are represented in Figure 4.

DORA has the highest average energy consumption at 82.98%. DORA is a destination-oriented routing algorithm that finds the shortest path from the source node to the destination. Although it is efficient in finding the optimal path, it consumes more energy than the other two algorithms. This higher energy consumption could be attributed to the complex calculations determining the best path. PSORS has an average energy consumption of 65.74%. PSORS is a routing scheme based on particle swarm optimization, which mimics the behavior of particles in search of the best solution. While PSORS performs better than DORA in energy consumption, it still requires significant energy. The energy consumption reduction compared to DORA can be attributed to the optimization techniques used in PSORS to find an optimal path. However, it is still not as energy-efficient as IPIRP.

The most energy-efficient routing algorithm is IPIRP, with an average energy consumption of 21.81%. IPIRP is an Intelligent Penguin Inspiration Routing Protocol that takes

RESEARCH ARTICLE

inspiration from the collective behavior of penguins to find efficient routes. The deficient energy consumption of IPIRP suggests that it leverages innovative techniques and algorithms that prioritize energy efficiency over other factors. This makes it a good choice for energy-constrained environments or applications where energy conservation is crucial.

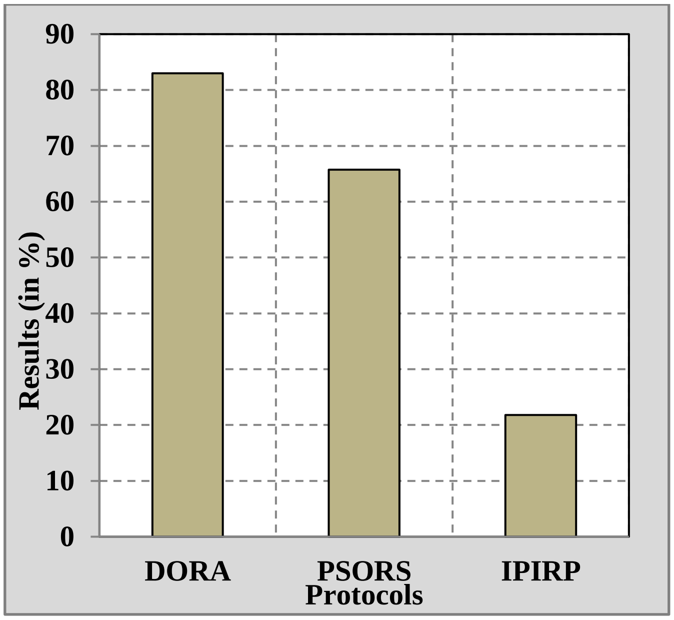

Figure 4 Energy Consumption Analysis

Table 5 Energy Consumption Result Values

Nodes	DORA	PSORS	IPIRP
150	75.61	57.94	16.88
300	74.51	56.39	14.02
450	81.21	60.98	21.08
600	77.90	60.36	19.32
750	84.47	68.90	22.72
900	82.27	61.51	22.70
1050	87.80	71.74	24.46
1200	86.63	69.51	24.34
1350	90.63	76.16	26.67
1500	88.75	73.96	25.95
Average	82.98	65.74	21.81

Figure 4 highlights the energy consumption comparison among DORA, PSORS, and IPIRP. While DORA provides optimal routing, it consumes the most energy. Thanks to its particle swarm optimization approach, PSORS reduces energy consumption compared to DORA. However, the most energy-efficient algorithm among the three is IPIRP, which leverages inspiration from penguins to achieve significant energy savings. The data presented in Figure 4 emphasizes the importance of considering energy efficiency when choosing a routing algorithm, especially in resource-constrained scenarios.

5.5. Network Lifetime

Figure 5 depicts a comprehensive analysis of network lifetimes achieved by three distinct routing algorithms: DORA, PSORS, and IPIRP. Figure 5 presents each algorithm's average network lifetimes, expressed as percentages, allowing for a comparative evaluation. Table 6 provides the result values of Figure 5.

DORA exhibits an average network lifetime of 17.19%. DORA operates on the principle of selecting paths based on the destination of data packets. While it manages to sustain network operation for a certain period, its network lifetime falls relatively short compared to the other two algorithms examined in this analysis. PSORS achieves a significantly higher average network lifetime of 37.41%. PSORS utilizes particle swarm optimization techniques, leveraging swarm intelligence to identify optimal routing paths within the network. This algorithm demonstrates superior performance in terms of network longevity, indicating its ability to optimize routing decisions and extend the overall lifespan of the network.

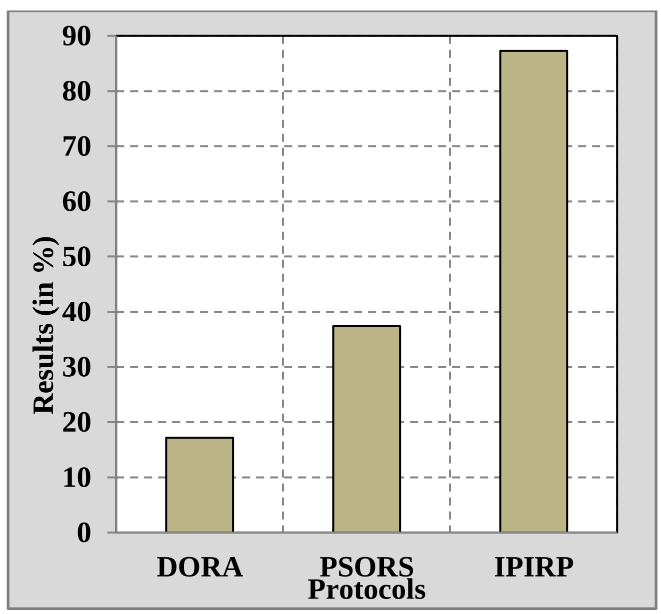

Figure 5 Network Lifetime Analysis

RESEARCH ARTICLE

The most notable performance is observed in IPIRP, which claims an impressive average network lifetime of 87.28%. IPIRP employs intelligent routing strategies inspired by the behavior of penguins. This protocol showcases remarkable enhancements over the other two algorithms, attaining the highest network lifetime. Its intelligent and nature-inspired approach enables effective and efficient routing decisions, leading to a substantially prolonged operational lifespan for the network.

Figure 5 comprehensively compares network lifetimes attained by three distinct routing algorithms: DORA, PSORS, and IPIRP. The presented data indicate that IPIRP outperforms DORA and PSORS, achieving an impressive average network lifetime of 87.28%. PSORS also performs better than DORA, attaining a network lifetime of 37.41%. These results underscore the importance of selecting and implementing efficient routing algorithms to optimize network longevity and overall performance.

Table 6 Network Lifetime Result Values

Nodes	DORA	PSORS	IPIRP
150	23.96	49.68	94.33
300	24.81	50.53	96.01
450	18.71	44.56	89.75
600	21.13	46.24	93.23
750	16.73	31.14	87.01
900	17.63	37.49	88.09
1050	12.77	29.14	81.55
1200	13.43	29.72	82.58
1350	10.51	27.79	80.04
1500	12.22	27.86	80.20
Average	17.19	37.41	87.28

6. CONCLUSION

The Intelligent Penguin Inspiration Routing Protocol (IPIRP) offers a promising solution for maximizing energy efficiency in Internet of Things-based Cloud Wireless Sensor Networks (IC-WSN). The scalability of routing algorithms is a significant challenge when accommodating many sensors while ensuring efficient data transmission. Existing protocols often struggle with network expansion, resulting in performance degradation and reduced efficiency. IPIRP addresses these challenges effectively by introducing innovative routing strategies. By dividing the network into smaller sub-networks or clusters, reducing routing overhead, and dynamically adjusting paths based on network conditions, IPIRP enhances scalability, reduces latency, and optimizes data transmission. This research aims to enable seamless network expansion, efficient resource utilization, and improved performance in various IC-WSN applications, including greenhouse farming. By focusing on scalable routing solutions, IPIRP empowers users to build robust and energy-efficient monitoring systems that provide reliable data for informed decision-making and enhance overall network efficiency. The proposed IPIRP protocol demonstrates potential in overcoming the limitations of existing routing algorithms and improving the energy efficiency of IC-WSN. Further evaluation of IPIRP in practical scenarios and exploration of its applicability in other IoT-based environments would be valuable. Advancements in routing protocols can facilitate the growth and optimization of IC-WSN, enabling enhanced data transmission and efficient resource utilization.

REFERENCES

[1] D. Chu, Y. Long, H. Yan, and H. Chen, “Research on Load Transfer of Greenhouse Based on Blockchain,” in 2022 14th International Conference on Measuring Technology and Mechatronics Automation (ICMTMA), 2022, pp. 957–961. doi: 10.1109/ICMTMA54903.2022.00194.

[2] Sonoda, Y. Takayama, A. Sugawara, and H. Nishi, “Greenhouse Heat Map Generation with Deep Neural Network Using Limited Number of Temperature Sensors,” in IECON 2022 – 48th Annual Conference of the IEEE Industrial Electronics Society, 2022, pp. 1–6. doi: 10.1109/IECON49645.2022.9968606.

[3] Solaimanian and R. Roshandel, “Effect of PV roof coverage on the lighting availability, heating, and cooling demands for a Venlo greenhouse in Tehran,” in 2023 8th International Conference on Technology and Energy Management (ICTEM), 2023, pp. 1–5. doi: 10.1109/ICTEM56862.2023.10083896.

[4] M. M. Abbassy and W. M. Ead, “Intelligent Greenhouse Management System,” in 2020 6th International Conference on Advanced Computing and Communication Systems (ICACCS), 2020, pp. 1317–1321. doi: 10.1109/ICACCS48705.2020.9074345.

[5] P. Qu et al., “Deep learning based detection of plant nutrient deficiency symptom and design of multi-layer greenhouse system,” in 2022 IEEE International Conference on Mechatronics and Automation (ICMA), 2022, pp. 641–645. doi: 10.1109/ICMA54519.2022.9856335.

[6] Castañeda-Miranda and V. M. Castaño, “Smart frost control in greenhouses by neural networks models,” Comput Electron Agric, vol. 137, pp. 102–114, 2017, doi: 10.1016/j.compag.2017.03.024.

[7] 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.

[8] S. Nandan, S. Singh, A. Malik, and R. Kumar, “A Green Data Collection & Transmission Method for IoT-Based WSN in Disaster Management,” IEEE Sens J, vol. 21, no. 22, pp. 25912–25921, 2021, doi: 10.1109/JSEN.2021.3117995.

[9] V. Agarwal, S. Tapaswi, and P. Chanak, “Intelligent Fault-Tolerance Data Routing Scheme for IoT-Enabled WSNs,” IEEE Internet Things J, vol. 9, no. 17, pp. 16332–16342, 2022, doi: 10.1109/JIOT.2022.3151501.

[10] Q. Li, R. Sun, H. Wu, and Q. Zhang, “Parallel distributed computing based wireless sensor network anomaly data detection in IoT

RESEARCH ARTICLE

framework," Cogn Syst Res, vol. 52, pp. 342–350, Dec. 2018, doi: 10.1016/j.cogsys.2018.07.007.

[11] D. Thomas, R. Shankaran, M. A. Orgun, and S. C. Mukhopadhyay, "SEC2: A Secure and Energy Efficient Barrier Coverage Scheduling for WSN-Based IoT Applications," IEEE Transactions on Green Communications and Networking, vol. 5, no. 2, pp. 622–634, 2021, doi: 10.1109/TGCN.2021.3067606.

[12] V. Rajasekar, P. Jayapaul, S. Krishnamoorthi, M. Saracevic, M. Elhoseny, and M. Al-Akaidi, "Enhanced WSN Routing Protocol for Internet of Things to Process Multimedia Big Data," Wirel Pers Commun, vol. 126, no. 3, pp. 2081–2100, 2022, doi: 10.1007/s11277-021-08760-1.

[13] Joshi, D. P. Kanungo, and R. K. Panigrahi, "WSN-Based Smart Landslide Monitoring Device," IEEE Trans Instrum Meas, vol. 72, pp. 1–12, 2023, doi: 10.1109/TIM.2023.3269746.

[14] P. Tewari and S. Tripathi, "An energy efficient routing scheme in internet of things enabled WSN: neuro-fuzzy approach," J Supercomput, vol. 79, no. 10, pp. 11134–11158, 2023, doi: 10.1007/s11227-023-05091-9.

[15] S. K. Barnwal, A. Prakash, and D. K. Yadav, "Improved African Buffalo Optimization-Based Energy Efficient Clustering Wireless Sensor Networks using Metaheuristic Routing Technique," Wirel Pers Commun, vol. 130, no. 3, pp. 1575–1596, 2023, doi: 10.1007/s11277-023-10345-z.

[16] R. Abraham and M. Vadivel, "An Energy Efficient Wireless Sensor Network with Flamingo Search Algorithm Based Cluster Head Selection," Wirel Pers Commun, vol. 130, no. 3, pp. 1503–1525, 2023, doi: 10.1007/s11277-023-10342-2.

[17] S. J. Charles, K. Palanisamy, M. Venugopal, and S. Shanmugam, "RPL enhancement with E-Sigma routing method," J Ambient Intell Humaniz Comput, vol. 14, no. 6, pp. 7813–7826, 2023, doi: 10.1007/s12652-023-04593-x.

[18] R. Jaganathan and R. Vadivel, "Intelligent Fish Swarm Inspired Protocol (IFSIP) for Dynamic Ideal Routing in Cognitive Radio Ad-Hoc Networks," International Journal of Computing and Digital Systems, vol. 10, no. 1, pp. 1063–1074, 2021, doi: 10.12785/ijcds/100196.

[19] Nazari, M. Kordabadi, R. Mohammadi, and C. Lal, "EQRSRL: an energy-aware and QoS-based routing schema using reinforcement learning in IoMT," Wireless Networks, 2023, doi: 10.1007/s11276-023-03367-9.

[20] M. Krishnan and Y. Lim, "Reinforcement learning-based dynamic routing using mobile sink for data collection in WSNs and IoT applications," Journal of Network and Computer Applications, vol. 194, p. 103223, 2021, doi: 10.1016/j.jnca.2021.103223.

[21] Benelhouri, H. Idrissi-Saba, and J. Antari, "An Improved Gateway-Based Energy-Aware Multi-Hop Routing Protocol for Enhancing Lifetime and Throughput in Heterogeneous WSNs," Simul Model Pract Theory, vol. 116, p. 102471, 2022, doi: https://doi.org/10.1016/j.simpat.2021.102471.

[22] J. Ramkumar and R. Vadivel, "Multi-Adaptive Routing Protocol for Internet of Things based Ad-hoc Networks," Wirel Pers Commun, vol. 120, no. 2, pp. 887–909, Apr. 2021, doi: 10.1007/s11277-021-08495-z.

[23] R. Jaganathan and V. Ramasamy, "Performance modeling of bio-inspired routing protocols in Cognitive Radio Ad Hoc Network to reduce end-to-end delay," International Journal of Intelligent Engineering and Systems, vol. 12, no. 1, pp. 221–231, 2019, doi: 10.22266/IJIES2019.0228.22.

[24] J. Ramkumar and R. Vadivel, "Improved frog leap inspired protocol (IFLIP) – for routing in cognitive radio ad hoc networks (CRAHN)," World Journal of Engineering, vol. 15, no. 2, pp. 306–311, 2018, doi: 10.1108/WJE-08-2017-0260.

[25] J. Ramkumar and R. Vadivel, "CSIP—cuckoo search inspired protocol for routing in cognitive radio ad hoc networks," in Advances in Intelligent Systems and Computing, Springer Verlag, 2017, pp. 145–153. doi: 10.1007/978-981-10-3874-7_14.

[26] V. Nivedhitha, A. G. Saminathan, and P. Thirumurugan, "DMEERP: A dynamic multi-hop energy efficient routing protocol for WSN," Microprocess Microsyst, vol. 79, p. 103291, 2020, doi: 10.1016/j.micpro.2020.103291.

[27] M. Elappila, S. Chinara, and D. R. Parhi, "Survivable Path Routing in WSN for IoT applications," Pervasive Mob Comput, vol. 43, pp. 49–63, Jan. 2018, doi: 10.1016/j.pmcj.2017.11.004.

[28] 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.

[29] G. Tong, S. Zhang, W. Wang, and G. Yang, "A particle swarm optimization routing scheme for wireless sensor networks," CCF Transactions on Pervasive Computing and Interaction, 2022, 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 & Science,Coimbatore.since 2016,Her research interests include IOT, data mining.She worked 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 as 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.

How to cite this article:

J. Jerlin Adaikala Sundari, G. Preethi, "Intelligent Penguin Inspiration Routing Protocol (IPIRP) for Maximizing Energy Efficiency in Internet of Things-Based Cloud Wireless Sensor Networks (IC-WSN)", International Journal of Computer Networks and Applications (IJCNA), 12(1), PP: 139-153, 2025, DOI: 10.22247/ijcna/2025/10.