Journal of Theoretical and Applied Information Technology
15th January 2025. Vol.103. No.1

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

ADAPTIVE ROUTING PROTOCOL FOR BUOYANT WIRELESS SENSOR NETWORK (B-WSN) USING HAMSTER OPTIMIZATION-BASED MAXPROP (HOMP)

DIVYA JOSE J[1], Dr. D. VIMAL KUMAR[2]
[1]Assistant Professor, Department of Computer Science,
Nehru Arts and Science College, Coimbatore, Tamil Nadu, India.
[2]Associate Professor and Head, Department of Computer Science,
Rathinam College of Arts and Science, Coimbatore, Tamil Nadu, India.
E-mail: [1]divyajose369@gmail.com, [2]drvimalcs@gmail.com

ABSTRACT

This study presents adaptive routing protocol proposed for Buoyant Wireless Sensor Networks (B-WSNs) through the integration of Hamster Optimization and Max Prop (HOMP). Buoyant WSNs operate in challenging aquatic environments, posing unique challenges such as underwater communication, limited energy resources, and prolonged network operation. To address these challenges HOMP the routing protocol is proposed which optimizes routing decisions and resource utilization in buoyant WSNs. Through extensive experimentation, we evaluate HOMP's performance in terms of packet delivery ratio, throughput, energy consumption, and network lifetime, comparing it with existing protocols. Results demonstrate that HOMP consistently outperforms other protocols, offering superior efficiency and reliability in data transmission. These findings have significant implications for applications in oceanography, environmental monitoring, and marine exploration, where reliable underwater sensing and monitoring are essential. By leveraging the capabilities of HOMP, buoyant WSNs can achieve greater accuracy, coverage, and longevity, enabling more effective data collection and analysis in aquatic environments.

Keywords: *Wireless Sensor Networks, Buoyant WSN, Routing Protocols, Hamster Optimization, MaxProp, Underwater Communication, Energy Efficiency, Network Lifetime.*

1. INTRODUCTION

Networks serve as the backbone of modern communication systems, facilitating the exchange of information and resources across various devices and locations. Among the numerous types of networks, Wireless Sensor Networks (WSNs) represent a significant advancement, particularly in the realm of data collection and monitoring[1]. These networks comprise interconnected sensor nodes capable of detecting and relaying data wirelessly. When the application extends to underwater environments, traditional WSNs face limitations. This necessitates the development of specialized networks known as Buoyant WSNs (B-WSNs), tailored for aquatic applications[2].

WSNs have revolutionized data collection across diverse domains, ranging from environmental monitoring to industrial automation[3]. These networks consist of sensor nodes distributed throughout the target area, communicating wirelessly to collect and transmit data to a central base station[4]. In applications where deploying wired infrastructure is impractical or cost-prohibitive, WSNs offer a flexible and scalable solution. They find extensive use in agriculture, healthcare, infrastructure monitoring, and more, enabling real-time insights and informed decision-making[5].

B-WSNs, a subtype of WSNs, address the unique challenges posed by underwater environments[6]. Traditional WSNs are ill-suited for aquatic applications due to their reliance on land-based infrastructure and wireless communication protocols optimized for air or solid mediums. B-WSNs, on the other hand, are specially designed to operate underwater, incorporating buoyant materials to maintain desired depths and waterproofing to protect against water ingress[7]. These networks leverage underwater communication techniques, such as acoustic signals, to facilitate data exchange in aquatic environments.

The usage of B-WSNs spans a diverse array of applications, each harnessing the capabilities of underwater sensing and communication for specific purposes. In

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

oceanographic research, these networks play a crucial role in studying ocean currents, temperature variations, and marine biodiversity[8]. By deploying sensor nodes across vast expanses of ocean, researchers can gather data to better understand complex marine ecosystems and their dynamics. Environmental monitoring represents another key application area for B-WSNs. These networks are deployed to assess water quality, monitor pollution levels, and track the impact of human activities on aquatic environments[9]. By continuously monitoring parameters such as dissolved oxygen, pH levels, and pollutant concentrations, B-WSNs provide valuable insights into the health of aquatic ecosystems and aid in conservation efforts.

The field of marine biology also benefits from the capabilities of B-WSNs. Researchers use these networks to study marine habitats, track the movements of marine species, and monitor coral reefs[10]. By deploying sensor nodes in strategic locations, scientists can gather data on underwater flora and fauna, contributing to our understanding of marine biodiversity and ecosystem dynamics.

B-WSNs find application in the monitoring of offshore infrastructure such as oil rigs, pipelines, and underwater cables[11]. By deploying sensor nodes around these structures, operators can detect potential faults, monitor structural integrity, and ensure compliance with safety regulations. This proactive approach to infrastructure monitoring helps mitigate risks and prevent costly downtime due to equipment failures or environmental hazards[12].

1.1. Problem Statement:

B-WSNs are critical for various underwater applications, such as environmental monitoring, oceanographic data collection, and marine life observation. These networks face significant challenges, including limited energy resources, dynamic underwater conditions, intermittent connectivity, and signal attenuation. Traditional routing protocols often struggle to maintain reliable communication and efficient resource utilization in such environments. There is a need for an adaptive, energy-efficient, and resilient routing solution that can cope with the unique challenges posed by underwater communication.

This research aims to address these issues by integrating Hamster Optimization (HO) with the Max Prop routing protocol, creating a robust solution tailored for buoyant WSNs. Hamster Optimization provides dynamic adaptability and resource management, while Max Prop enhances fault tolerance and probabilistic routing. The combined HO-Max Prop approach is expected to improve network performance by optimizing energy consumption, enhancing fault tolerance, and ensuring reliable data delivery. The effectiveness of this integrated protocol will be evaluated through detailed simulations using ns-3, focusing on key performance metrics such as throughput, packet delivery ratio, energy consumption, latency, and network lifetime.

1.2. Motivation:

The increasing frequency and severity of oil spills in oceans pose a significant threat to marine ecosystems and coastal communities. Effective monitoring and rapid response are crucial to mitigating the environmental damage caused by such spills. Traditional monitoring methods often fall short due to the vast and dynamic nature of ocean environments, necessitating the deployment of advanced technologies such as B-WSNs. These networks can provide real-time data on the spread and impact of oil spills, enabling timely and informed decision-making.

The deployment of B-WSNs in underwater environments presents unique challenges, including limited energy resources, fluctuating underwater conditions, and intermittent connectivity. Addressing these challenges requires innovative solutions to ensure reliable communication and efficient resource utilization. Integrating HO with the Max Prop routing protocol offers a promising approach to enhance the performance of B-WSNs. HO's adaptive optimization capabilities, combined with Max Prop's probabilistic routing, can improve energy efficiency, fault tolerance, and data delivery reliability.

1.3. Objective:

The primary objective of this research is to develop and evaluate a robust routing protocol for B-WSNs aimed at enhancing the monitoring and management of oil spills in ocean environments. By integrating HO with the Max Prop routing protocol, this study seeks to address the critical challenges associated with underwater communication, such as limited energy resources, dynamic environmental conditions, and intermittent connectivity. The integrated HO-Max Prop (HOMP) protocol will be designed to optimize energy consumption, improve fault tolerance, and ensure reliable data delivery. This will involve detailed simulations using the ns-

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

3 network simulator to assess the performance of the proposed protocol in terms of key metrics such as throughput, packet delivery ratio, energy efficiency, latency, and network lifetime. The ultimate goal is to demonstrate that the HOMP integration can significantly enhance the operational efficiency and reliability of buoyant WSNs, thereby providing a more effective solution for real-time monitoring and rapid response to oil spills. A significant contribution can be made towards protecting marine ecosystems and supporting environmental management efforts in the face of increasing oil spill incidents is the intention of this work.

2. LITERATURE REVIEW

"UWMOT"[13]explores the fusion of WSNs and machine learning for underwater motion tracking and monitoring. By integrating sensor data with machine learning algorithms, it enables real-time detection and analysis of underwater movements, contributing to various applications such as marine life observation and security surveillance. "LoRaWAN"[14] proposes an Internet of Things (IoT) system based on the LoRaWAN technology for water quality monitoring. Through low-power, long-range communication, the system enables remote monitoring of water parameters, facilitating timely intervention and resource management in rural regions.

"Self-Powered WSN"[15] presents a self-powered wireless sensor system for water monitoring, utilizing a low-frequency electromagnetic-pendulum energy harvester. The system operates autonomously by harvesting ambient energy from water movements, providing continuous data collection for water quality assessment. "Trustworthy DCS"[16] Addressing security concerns in unmanned aerial vehicle (UAV)-assisted WSNs, this scheme proposes a trustworthy data collection approach based on active spot-checking. It enhances the reliability of collected data while minimizing energy consumption by verifying data integrity through periodic spot-checks. "Wireless Power and Data Transfer System"[17] Introducing a novel system for underwater WSNs, this work enables simultaneous wireless power and data transfer with full-duplex capability. It enhances the efficiency and reliability of underwater sensor networks by integrating power and data transmission, facilitating continuous operation without the need for frequent battery replacement.

"SOAM Networks"[18] are designed for underwater environments, featuring self-organizing capabilities to adapt to dynamic conditions. SOAM networks enhance the scalability and resilience of underwater sensor networks by enabling autonomous node deployment and network formation, supporting various monitoring and exploration tasks. "Cooperative Routing Protocol Based on Q-Learning"[19]leverages reinforcement learning techniques, specifically Q-learning, to optimize routing decisions in underwater optical-acoustic hybrid WSNs. The protocol dynamically adjusts routing paths, improving network performance and reliability in challenging underwater environments based on the experience from its past.

"Tech Node" [20] enabling technologies for underwater WSNs, focusing on node deployment strategies and data collection challenges. It provides insights into optimizing network deployment and enhancing data gathering capabilities in underwater environments by addressing factors such as communication range, node mobility, and energy efficiency. "OCPC"[21] Exploring the convergence of optical communication and positioning techniques, this work advances underwater WSNs' flexibility and performance. It enables precise and efficient data exchange in dynamic underwater environments by integrating optical communication for data transmission and positioning for node localization. "Sensor Injection Based Routing Protocol"[22] proposes a routing scheme based on sensor injection for effective load balancing in underwater WSNs. It optimizes network resource utilization and mitigates congestion, ensuring smooth and efficient data transmission in underwater environments by dynamically adjusting data routing paths through sensor injection. "Hierarchical Localization"[23] presents a hierarchical framework for localizing nodes in large-scale underwater WSNs, contributing to sustainable ocean health monitoring. It achieves accurate node positioning with minimal energy overhead by organizing nodes into hierarchical clusters and employing efficient localization techniques. The algorithm operates by iteratively refining node positions within clusters and aggregating location information across the network, enabling precise monitoring of underwater environments at scale. Several Bio-inspired optimizations are implanted in different network to enhance the performance of network in different means [24] – [45].

"MO-CBACORP"[46]introduces a novel approach to energy-efficient and secure routing in underwater monitoring WSNs. Leveraging a combination of energy-aware routing strategies and enhanced security measures, MO-CBACORP addresses the unique challenges of underwater communication, ensuring reliable data transmission while optimizing energy consumption. Its working mechanism involves dynamically adapting routing paths based on energy levels and network conditions, thereby prolonging network lifetime and enhancing data integrity. "HOCOR"[47]a routing protocol designed for underwater WSNs, combining optimization techniques with cooperative opportunistic routing strategies. HOCOR enhances data delivery efficiency and reliability by utilizing both network optimization and opportunistic data forwarding, improving the overall performance of underwater sensor networks.

3.ADAPTIVE ROUTING PROTOCOL FOR BUOYANT WIRELESS SENSOR NETWORKS (B-WSNs).

3.1. Maximizing Probability Routing:

Routing protocols enables communication and data exchange among sensor nodes. Acoustic-based routing protocols play a crucial role in enabling reliable and efficient communication in B-WSNs deployed for underwater exploration, environmental monitoring, marine research, and offshore surveillance. Opportunistic Routing a type of Acoustic-based routing protocol which is a dynamic and adaptive routing paradigm used in wireless communication networks, including B-WSNs operating in underwater environments. Taditional routing protocols rely on predefined paths, opportunistic routing takes advantage of opportunistic communication opportunities to forward data packets through the network.

Maximizing Probability Routing (Max Prop) is an Opportunistic Routing protocol designed for wireless communication networks, particularly Delay-Tolerant Networks (DTNs) and challenged environments where traditional routing protocols are ineffective due to intermittent connectivity, long delays, or limited resources. Max Prop aims to maximize the probability of successful data delivery by exploiting opportunistic communication opportunities and adapting routing decisions dynamically.

3.1.1. Initialization:

Max Prop begins with the phase where it initiates the routing process by distributing initial routing information throughout the network, setting the stage for further process. This information includes node states, contact opportunities, message priorities, and other parameters necessary for routing decisions. Through this process, nodes establish a common understanding of the network topology, communication opportunities, and message requirements, enabling coordinated routing operations.

The initialization phase involves defining and distributing initial parameters and states to all nodes in the network. Let N denote the total number of nodes in the network, $S_i$represent the state of node i, and P_i denote the priority assigned to messages generated by node i. The initialization process can be expressed mathematically using Eq.(1) and Eq.(2).

$$S_i^{(0)} = InitializeState(i) \quad (1)$$

$$P_i^{(0)} = InitializePriority(i) \quad (2)$$

where $S_i^{(0)}$represents the initial state of node i and $P_i^{(0)}$represents the initial priority assigned to messages generated by node i. The functions $InitializeState(i)$ and $InitializePriority(i)$ determine the initial state and priority for each node based on network-wide parameters and local characteristics.

The initialization phase ensures that all nodes are equipped with the necessary information to participate in the routing process effectively. By establishing initial states and priorities, nodes are prepared to generate messages, make forwarding decisions, and adapt to changing network conditions throughout the routing process. This phase sets the stage for subsequent steps in the Max Prop protocol. Message generation, opportunistic forwarding, utility calculation, and other routing operations rely on the initial parameters and states established during the initialization phase.

3.1.2. Message Generation:

Message Generation marks the initiation of the routing process. In this phase, nodes in the network generate messages intended for transmission to other nodes. These messages carry vital information, such as their destination, priority, and any constraints or requirements for delivery. The message generation process involves assigning priorities to messages based on their importance or urgency. Let $M_i$represent the set of messages

Journal of Theoretical and Applied Information Technology
15th January 2025. Vol.103. No.1

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

generated by node i, D_{ij} denote the destination of message j generated by node i, and C_{ij} represent any constraints or requirements associated with message j. The message generation process can be expressed mathematically is shown in Eq.(3) and Eq.(4).

$$D_{ij} = GenerateDestination(i,j) \quad (3)$$

$$C_{ij} = GenerateConstraints(i,j) \quad (4)$$

where D_{ij} represents the destination of message j generated by node $i$and C_{ij}represents any constraints or requirements associated with message$j$. The functions $GenerateDestination\ (i,j)$ and $GenerateConstraints\ (i,j)$ determine the destination and constraints for each message based on network-wide parameters and local characteristics.

Nodes prioritize the transmission of time-sensitive information by assigning priorities to messages, which ensures the efficiency and effective message delivery. The priority assigned to each message influences its likelihood of successful transmission and impacts subsequent routing decisions in the network. The message generation phase lays the groundwork for opportunistic forwarding, as messages are stored in node buffers awaiting suitable forwarding opportunities. The destination and constraints associated with each message guide subsequent routing decisions, determining the optimal relay nodes and transmission paths. The effectiveness of the message generation phase is crucial for the overall performance of the Max Prop protocol. Accurate and timely message generation ensures that critical information is transmitted efficiently, minimizing delays and maximizing the probability of successful data delivery.

3.1.3. Opportunistic Forwarding:

Third phase of Max Prop builds upon the message generation phase to facilitate the transmission of messages through the network. In Opportunistic Forwarding nodes opportunistically forward messages to neighbouring nodes based on the probability of successful delivery. It allows messages to traverse the network efficiently, leveraging intermittent communication links and dynamic node encounters.

Opportunistic forwarding involves selecting relay nodes for message transmission based on the probability of successful delivery and network conditions. Let R_{ij} denote the relay node selected for message j generated by node i, and P_{ij} represent the probability of successful delivery to relay node R_{ij}. The opportunistic forwarding process can be expressed mathematically with Eq.(5) and Eq.(6).

$$R_{ij} = SelectRelay(i,j) \quad (5)$$

$$P_{ij} = CalculateProbability\left(i,j,R_{ij}\right) \quad (6)$$

where R_{ij} represents the relay node selected for message j generated by node i and P_{ij} represents the probability of successful delivery to relay node R_{ij}. The functions $SelectRelay(i,j)$ and $CalculateProbability\left(i,j,R_{ij}\right)$ determine the optimal relay node and calculate the probability of successful delivery based on network-wide parameters and local characteristics.

Nodes optimize the transmission of messages through the network, maximizing the likelihood of reaching their destinations by selecting relay nodes and calculating the probability of successful delivery. Opportunistic forwarding allows messages to traverse multiple hops, leveraging intermittent communication links and dynamic node encounters to overcome network disruptions and delays.

3.1.4. Utility Calculation:

The Utility Calculation plays a crucial role in guiding opportunistic forwarding decisions by assessing the potential benefits of forwarding messages to neighboring nodes. In this phase, nodes calculate the utility of forwarding messages based on various factors, such as node buffer occupancy, expected transmission time, and the likelihood of successful delivery. Utility calculation enables nodes to prioritize relay nodes and optimize message transmission through the network.

Utility calculation involves evaluating the potential benefits of forwarding messages to neighboring nodes. Let U_{ij} represent the utility of forwarding message j generated by node i, and B_i denote the buffer occupancy of node i. The utility calculation process can be expressed in mathematical form with Eq.(7).

$$U_{ij} = CalculateUtility(i,j,B_i) \quad (7)$$

where U_{ij}represents the utility of forwarding message j generated by node i and B_i denotes the buffer occupancy of node i. The function $CalculateUtility(i,j,B_i)$ determines the utility of forwarding the message based on network-wide parameters and local characteristics. Nodes prioritize relay nodes based on their potential benefits, such as reducing buffer occupancy by calculating the utility of forwarding messages. This optimization process ensures efficient and effective message transmission through the network, maximizing the likelihood of reaching their destinations.

3.1.5. Forwarding Decision:

Forwarding Decision builds upon the utility calculation phase to determine the optimal relay nodes for message transmission. Nodes make informed decisions about which neighboring nodes to forward messages to based on the calculated utility values. Forwarding decisions are crucial for optimizing message transmission and maximizing the likelihood of successful delivery. The forwarding decision process involves selecting relay nodes based on the utility values calculated in the previous step. Let F_{ij} denote the forwarding decision for message j generated by node i, and U_{ij} represent the utility value associated with message j.

$$F_{ij} = MakeForwardingDecision(i, j, U_{ij}) \quad (8)$$

where in Eq.(8),F_{ij} represents the forwarding decision for message j generated by node i and U_{ij} denotes the utility value associated with message j. The function $MakeForwardingDecision(i, j, U_{ij})$ determines the optimal relay node based on the calculated utility value and network-wide parameters.Nodes prioritize relay nodes based on their utility values by by making informed forwarding decisions, ensuring efficient and effective message transmission through the network. Forwarding decisions consider factors such as buffer occupancy, expected transmission time, and the likelihood of successful delivery, optimizing message routing to maximize the probability of reaching their destinations.

3.1.6. Feedback and Adaptation:

This phase enhances the robustness and efficiency of the routing process by enabling nodes to dynamically adjust their behavior in response to changes in the network environment.

Feedback and adaptation involve updating routing parameters and strategies based on feedback information exchanged between nodes. Let F_{ij} represent the feedback information received by node i from node j, and A_i denote the adaptation parameter for node i. The feedback and adaptation process can be mathematically expressed as shown in Eq.(9).

$$A_i = AdaptRouting(i, F_{ij}) \quad (9)$$

where A_i represents the adaptation parameter for node i and F_{ij} denotes the feedback information received by node i from node j. The function $AdaptRouting(i, F_{ij})$ updates the routing parameters and strategies of node ii based on the received feedback information. By exchanging feedback information and adapting routing parameters, nodes optimize their routing strategies to better suit the current network conditions and improve overall routing performance. Feedback information may include acknowledgments (ACKs), local observations of communication links, or performance metrics such as message delivery rates or transmission delays. Feedback and adaptation enable nodes to address network dynamics and mitigate the effects of link failures, node mobility, and channel fading. By dynamically adjusting their routing strategies, nodes can respond to changes in the network environment and maintain efficient and robust message transmission.

3.1.7. Buffer Management

Buffer Management focuses on optimizing the utilization of node buffers to ensure efficient message storage and transmission. Nodes manage their buffers by prioritizing messages based on their importance, urgency, or delivery constraints. Buffer management plays a crucial role in maximizing the probability of successful message delivery while minimizing buffer overflow and resource wastage. Buffer management involves prioritizing messages in node buffers based on their characteristics and network conditions. Let M_i represent the set of messages stored in the buffer of node i, and P_{ij} denote the priority assigned to message j stored in the buffer of node i.

$$P_{ij} = AssignPriority(i, j) \quad (10)$$

where P_{ij} represents the priority assigned to message j stored in the buffer of node i. The function $AssignPriority(i, j)$ determines the priority of each message based on its characteristics, such as importance, urgency, or delivery constraints. By prioritizing messages in node buffers, buffer management ensures that critical or time-sensitive information is transmitted promptly, maximizing the likelihood of successful delivery.

Buffer management strategies may include FIFO (First-In-First-Out), LIFO (Last-In-First-Out), or priority-based scheduling algorithms, depending on the specific requirements of the network and application. Buffer management helps prevent buffer overflow and resource wastage by discarding low-priority or expired messages when buffer space is limited. This ensures efficient utilization of node resources and improves overall network performance.

3.1.8. Termination and Convergence:

In this last phase of Max Prop the routing process iterates until a termination criterion is met, signaling the convergence of the protocol. Termination and convergence are essential for determining when to halt the routing process and finalize the routing decisions made by nodes. It involve defining a termination criterion and monitoring the convergence of the routing process. Let Trepresent the termination criterion, Idenote the current iteration of the routing process, and C signify the convergence status of the protocol. The termination and convergence process is represented mathematically in Eq.(11).

$$TC = CheckConvergence(I) \quad (11)$$

where C represents the convergence status of the protocol based on the current iteration I. The function $CheckConvergence(I)$ evaluates whether the routing process has converged based on predefined convergence criteria, such as the number of iterations, message delivery rates, or network stability.Once the convergence status C indicates that the routing process has converged, the protocol terminates, and the final routing decisions are established. The termination criterion T may be a maximum number of iterations, a threshold for message delivery rates, or a predefined convergence threshold. Termination and convergence ensure that the routing process halts when the protocol reaches a stable state and the optimal routing decisions are finalized. By monitoring convergence and defining a termination criterion, the protocol prevents unnecessary iterations and conserves network resources.

3.2. Hamster Optimization:

Hamsters are nocturnal rodents known for their cheek pouches, which they use to store food. They exhibit burrowing behavior, creating intricate tunnels. Hamsters are solitary and territorial, often showing aggression. They are active and enjoy burrowing such a curious creatures. They require exercise and mental stimulation to stay healthy and happy. Its characteristics are unique which is the inspiration to constructthe phases of Hamster Optimization (HO).

3.2.1. Exploration and Nesting:

Exploration and Nesting initiates the optimization process by initializing a population of candidate solutions and encouraging exploration of the solution space. This step mimics the behavior of hamsters as they explore their environment to find suitable nesting spots and gather resources for their nests.The exploration and nesting phase involves generating an initial population of candidate solutions. Let S represent the set of candidate solutions, N denote the population size, and s_i represent an individual solution in the population. The exploration and nesting process can be expressed in Eq.(12).

$$S = \{s_1, s_2, \cdots s_N\} \quad (12)$$

The exploration and nesting phase encourages diversity among the candidate solutions to explore different regions of the solution space. Let $(si,)$ denote the dissimilarity between solutions si and sj, and $d$max represent the maximum dissimilarity threshold. The exploration process aims to ensure that solutions in the population are sufficiently different from each other. This can be expressed mathematically in Eq.(13).

$$\mathrm{D}(s_i, s_j) \geq d_{max}, \forall s_i, s_j \in S, i \neq j \quad (13)$$

where $\mathrm{D}(s_i, s_j)$represents the dissimilarity between solutions s_i and s_j, and $d_{max}$is the maximum dissimilarity threshold.

The exploration and nesting phase may involve the initialization of solution parameters within predefined ranges. Let p_k represent the k^{th} parameter of solution s_i, p_{min}^kdenote the minimum allowable value for parameter p_k, and p_{max}^k represent the maximum allowable value for parameter p_k. The initialization process ensures that solution parameters are within feasible ranges is shown in Eq.(14).

$$p_{min}^k \leq p_k \leq p_{max}^k, \forall s_i \in S, \forall k \quad (14)$$

where p_k represents the k^{th} parameter of solution s_i, and p_{min}^k and p_{max}^k are the minimum and maximum allowable values for parameter p_k, respectively.The exploration and nesting phase may involve the generation of diverse initial solutions using stochastic techniques such as random sampling or initialization based on domain-specific knowledge. Let $f(s_i)$denote the fitness of solution s_i, which represents its quality or performance with respect to the optimization problem. The exploration process aims to generate a diverse set of initial solutions with acceptable fitness values are mathematically represented in Eq.(15).

$$f(s_i) \geq f_{min}, \forall s_i \in S \quad (15)$$

where $f(s_i)$ represents the fitness of solution s_i, and $f_{min}$is the minimum acceptable fitness threshold.

3.2.2. Randomization:

Randomization is a key mechanism for escaping local optima and discovering novel regions of the solution space, akin to how hamsters explore their environment by taking random paths to uncover new territories. The randomization phase involves introducing stochastic elements into the optimization process. Let Srepresent the set of candidate solutions, s_i denote an individual solution in the population, and R represent the randomization parameter. The randomization process can be mathematically exrpressed shown in Eq.(16).

$$s_i' = Randomize(s_i, R) \quad (16)$$

where s_i' represents the modified solution after randomization, and $Randomize(s_i, R)$ is a function that applies random perturbations to solution s_i based on the randomization parameter R. This introduces variability into the solutions, allowing for exploration of different regions of the solution space.

The randomization phase may involve the introduction of noise or uncertainty into solution parameters to promote exploration. Let p_k represent the k^{th} parameter of solution s_i, p_{min}^k denote the minimum allowable value for parameter p_k, and p_{max}^k represent the maximum allowable value for parameter p_k. The randomization process can perturb solution parameters within feasible ranges is mathematically represented in Eq.(17) and Eq.(18).

$$p_k' = p_k + \epsilon \quad (17)$$

$$p_k' = Clip\left(p_k', p_{min}^k, p_{max}^k\right) \quad (18)$$

where p_k' represents the perturbed value of parameter p_k, ϵ is a random perturbation term, and $Clip\left(p_k', p_{min}^k, p_{max}^k\right)$ ensures that p_k' remains within the feasible range defined by p_{min}^k and p_{max}^k.

The randomization phase may also involve the introduction of randomness into the selection of candidate solutions for further exploration. Let N represent the population size, and $s_i$and s_j denote individual solutions in the population. The randomization process can select candidate solutions randomly from the population is expressed in Eq.(19).

$$\begin{gathered} s_i' = s_j, \\ s_j \in S, \\ i, j \in \{1,2,\cdots,N\}, \\ i \neq j \end{gathered} \quad (19)$$

where s_i' represents the modified solution after randomization, and s_j is a randomly selected solution from the population S.

The randomization phase may involve the introduction of probabilistic decision-making mechanisms to determine the extent of random perturbations applied to solutions. Let P represent the probability distribution governing the randomization process, and $p(s_i')$ denote the probability of selecting the perturbed solution s_i'. The randomization process can be probabilistic in nature is mathematically represented in Eq.(20).

$$p(s_i') = P(Randomize) \quad (20)$$

where $p(s_i')$ represents the probability of selecting the perturbed solution s_i', and $P(Randomize)$ is the probability distribution governing the randomization process.

3.2.3. Food Foraging:

This phase simulates the behavior of hamsters as they search for food and gather resources for their nests. This step focuses on evaluating the quality of candidate solutions, akin to the nutritional value of food sources for hamsters, and selecting the most promising solutions for further exploration. The food foraging phase involves evaluating the fitness or quality of candidate solutions based on their performance in solving the optimization problem. Let S represent the set of candidate solutions, (s_i) denote the fitness of solution,s_i and f_{best} represent the fitness of the best solution found so far. The food foraging process can be expressed mathematically in Eq.(21) and Eq.(22).

$$(s_i) = EvaluateFitness(s_i) \quad (21)$$

$$f_{best} = \max\left(f_{best}, f(s_i)\right), \forall s_i \in S \quad (22)$$

where $f(s_i)$represents the fitness of solution s_i, and $EvaluateFitness(s_i)$ is a function that assesses the quality of the solution. In Additionf_{best} is updated to store the fitness of the best solution found so far.

The food foraging phase may involve the exploration of neighboring solutions to assess their quality and potential for improvement. Let s_i' represent a neighboring solution obtained through perturbation or modification of solution s_i, and $\delta(s_i, s_i')$ denote the change in fitness between solutions s_i and s_i'. The food foraging process can consider neighboring solutions for evaluation is shown in Eq.(23) and Eq.(24).

$$(s_i') = EvaluateFitness(s_i') \quad (23)$$

$$\delta(s_i, s_i') = f(s_i') - f(s_i) \quad (24)$$

where $f(s_i')$ represents the fitness of neighboring solution s_i', and $\delta(s_i, s_i')$ quantifies the change in fitness between solutions s_i and s_i'.

The food foraging phase may involve the selection of candidate solutions based on their fitness values, prioritizing solutions with higher fitness for further exploration. Let $S_{selected}$ represent the subset of candidate solutions selected for further processing, and θ denote a selection threshold. The food foraging process can select promising solutions based on their fitness values:

$$S_{selected} = \{s_i | f(s_i) \geq \theta\} \quad (25)$$

where $S_{selected}$ contains candidate solutions with fitness values exceeding the selection threshold θ.The food foraging phase may involve the exploration of multiple regions of the solution space simultaneously to ensure thorough exploration and avoid premature convergence to suboptimal solutions. This can be achieved through parallel exploration or distributed evaluation of candidate solutions.

3.2.4. Burrow Construction:

This phase Burrow Construction focuses on the modification and refinement of candidate solutions to construct improved solutions iteratively. It analogous to how hamsters arrange bedding material and create tunnels and chambers in their nests to optimize their living space and ensure comfort and safety. Burrow construction involves modifying candidate solutions to improve their quality and fitness. Let S represent the set of candidate solutions, s_i denote an individual solution in the population, and s_i' represent the modified solution after burrow construction. The burrow construction process can be expressed mathematically in Eq.(26).

$$s_i' = ConstructBurrow(s_i) \quad (26)$$

where s_i' represents the modified solution obtained through burrow construction, and $ConstructBurrow(s_i)$is a function that modifies solution s_i to improve its quality and fitness.

It also involve the integration of information from neighboring solutions to guide the modification process. Let (s_i) represent the neighborhood of solution s_i, s_j denote a neighboring solution, and w_{ij} represent the weight or influence of solution s_j on the modification of s_i. The burrow construction process can consider information from neighboring solutions.

$$s_i' = \sum_{s_j} \in N(s_i)\, w_{ij} \cdot s_j \quad (27)$$

where in Eq.(27),s_i' represents the modified solution obtained through burrow construction, and the sum aggregates the contributions of neighboring solutions weighted by their influence.

Burrow construction may involve the application of local search techniques to fine-tune candidate solutions and explore nearby regions of the solution space. Let s_i represent the current solution, s_i' denote the modified solution after local search, and Δ denote the perturbation applied during the local search process. The burrow construction process can incorporate local search is shown in Eq.(28).

$$s_i' = s_i + \Delta \quad (28)$$

where s_i' represents the modified solution obtained through local search, and Δ represents the perturbation applied to the current solution s_i during the local search process.

Burrow construction may involve the application of adaptive strategies to guide the modification process based on feedback information and problem characteristics. Let (s_i) represent the fitness of solution s_i, F_{best} denote the fitness of the best solution found, and α represent an adaptation parameter. The burrow construction process can adaptively adjust the modification process is mathematically represented in Eq.(29).

$$s_i' = s_i + \alpha \cdot \left(F_{best} - F(s_i)\right) \quad (29)$$

where s_i'represents the modified solution obtained through adaptive burrow construction, and the adaptation parameter α scales the magnitude of the modification based on the difference between the fitness of the current solution and the fitness of the best solution.

3.2.5. Memory and Adaptation:

This phase mimics the behavior of hamsters, which rely on memory and past experiences to navigate their environment efficiently and adapt to changing conditions. The memory and adaptation phase involve the storage and utilization of information from past iterations to guide the optimization process. Let S represent the set of candidate solutions, si denote an individual solution in the population, and M represent the memory matrix storing information about past solutions and their performance. The memory and adaptation process is mathematically expressed in Eq.(30).

$$M = UpdateMemory(M, S) \quad (30)$$

where M represents the updated memory matrix containing information about past solutions and their performance, and $UpdateMemory(M, S)$ is a function that updates the memory matrix based on the current population of candidate solutions S.

The memory and adaptation phase may involve the retrieval of information from the memory matrix to guide the modification and selection of candidate solutions. Let s_i' represent the modified solution obtained through memory-based adaptation, and w_i denote the weight or influence of solution s_i based on its performance stored in the memory matrix. The memory-based adaptation process can be expressed with Eq.(31).

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

$$s_i' = \sum_{s_j} \in Mw_{ij} \cdot s_j \quad (31)$$

where s_i'represents the modified solution obtained through memory-based adaptation, and the sum aggregates the contributions of past solutions weighted by their influence stored in the memory matrix.

This phase may involve the adaptation of solution parameters based on past experiences and performance feedback. Let p_k represent the k^{th} parameter of solution s_i, p_{best}^k denote the best-performing value for parameter p_k stored in the memory matrix, and β represent an adaptation parameter. The parameter adaptation process can be expressed mathematically in Eq.(32).

$$p_k' = p_k + \beta \cdot (p_{best}^k - p_k) \quad (32)$$

where p_k'represents the adapted value of parameter p_k, and the adaptation parameter β scales the magnitude of the adaptation based on the difference between the current parameter value p_k and the best-performing value $p$best$k$ stored in the memory matrix.

The exploitation of promising solutions identified in past iterations to guide the optimization process. Let $S_{exploit}$ represent the subset of candidate solutions selected for exploitation, and $N_{exploit}$ denote the number of solutions to exploit from the memory matrix. The exploitation process can be expressed in Eq.(35).

$$S_{exploit} = \text{Select Exploitable Solutions } (M, N_{exploit}) \quad (33)$$

where $S_{exploit}$ contains candidate solutions selected for exploitation from the memory matrix,and $\text{Select Exploitable Solutions } (M, N_{exploit})$ is a function that selects the most promising solutions from the memory matrix based on their performance.

3.2.6. Local Search and Optimization:

This phase is analogous to how hamsters explore the nearby areas of their nests to find food and resources efficiently. Local search and optimization involve exploring neighboring solutions to improve the quality and fitness of candidate solutions. Let S represent the set of candidate solutions, s_i denote an individual solution in the population, and s_i' represent the modified solution obtained through local search. The local search and optimization process can be expressed asEq.(34).

$$s_i' = LocalSearch(s_i) \quad (34)$$

where s_i'represents the modified solution obtained through local search, and $LocalSearch(s_i)$is a function that explores the neighborhood of solution si to identify improvements.

The local search and optimization phase may involve the exploration of multiple neighboring solutions to identify the most promising modifications. Let $N(s_i)$ represent the neighborhood of solution s_i, s_j denote a neighboring solution, and $f(s_j)$ represent the fitness of solution s_j. The local search process can select the most promising neighboring solution for modification is mathematically represented in Eq.(35).

$$s_i' = argmax_{s_j \in N(s_i)} f(s_j) \quad (35)$$

where s_i' represents the modified solution obtained through local search, and $argmax_{s_j \in N(s_i)} f(s_j)$selects the neighboring solution with the highest fitness value within the neighborhood of solution s_i.

The local search and optimization phase may involve the application of gradient-based optimization techniques to guide the exploration process. Let $\nabla(s_i)$ denote the gradient of the fitness function with respect to solution s_i, and η represent the step size or learning rate. The local search process can update the solution based on the gradient direction is expressed in Eq.(36).

$$s_i' = s_i - \eta \cdot \nabla(s_i) \quad (36)$$

where s_i' represents the modified solution obtained through local search, and the gradient descent step $\eta \cdot \nabla(s_i)$ moves the solution towards the direction of steepest descent in the fitness landscape.Local search and optimization phase may involve the incorporation of problem-specific knowledge or heuristics to guide the exploration process. Let H represent the set of problem-specific heuristics, and h_k denote a heuristic function. The local search process can apply problem-specific heuristics to guide solution modifications:

$$s_i' = ApplyHeuristic(s_i, H) \quad (37)$$

where s_i' represents the modified solution obtained through local search, and $ApplyHeuristic(s_i, H)$ is a function that applies problem-specific heuristics from the set H to solution s_i.

3.2.7. Social Interaction and Communication:

In HO the concept of interaction and communication among candidate solutions to facilitate knowledge sharing and collective decision-making. This phase replicates the social behavior observed in hamsters, where individuals interact with one another to exchange information and coordinate activities for mutual benefit. Social interaction and communication involve the

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

exchange of information among candidate solutions to improve the overall performance of the optimization process. Let S represent the set of candidate solutions, s_i denote an individual solution in the population, and s_i' represent the modified solution obtained through social interaction. The social interaction and communication process can be expressed as Eq.(38).

$$s_i' = Interact(s_i, S) \quad (38)$$

where s_i' represents the modified solution obtained through social interaction, and $Interact(s_i, S)$ is a function that enables solution s_i to interact with other solutions in the population S to exchange information.

Social interaction and communication may involve the sharing of knowledge or experiences among neighboring solutions to facilitate collective learning. Let $N(s_i)$ represent the neighborhood of solution s_i, s_j denote a neighboring solution, and $f(s_j)$ represent the fitness of solution s_j. The social interaction process can leverage information from neighboring solutions is represented mathematically in Eq.(27). where s_i' represents the modified solution obtained through social interaction, and the sum aggregates the contributions of neighboring solutions weighted by their influence w_{ij}.Social interaction and communication may involve the formation of groups or communities within the population to facilitate more effective knowledge sharing and collaboration. Let G represent the set of groups or communities formed within the population, and gk denote a group containing a subset of candidate solutions. The social interaction process can promote collaboration within groups is shown in mathematical form as Eq.(39).

$$s_i' = Collaborate(s_i, G) \quad (39)$$

where s_i' represents the modified solution obtained through social interaction, and $Collaborate(s_i, G)$ is a function that enables solution s_i to collaborate with other solutions in the group G to exchange information and coordinate activities.

In Addition social interaction and communication may involve the establishment of communication channels or mechanisms for exchanging information efficiently. Let C represent the set of communication channels established among candidate solutions, and c_{ij} denote a communication channel between solutions s_i and s_j. The social interaction process can enable communication between solutions is shown in Eq.(40).

$$s_i' = Communicate(s_i, C) \quad (40)$$

where s_i' represents the modified solution obtained through social interaction, and $Communicate(s_i, C)$ is a function that enables solution si to communicate with other solutions via established communication channels C.

3.2.8. Termination and Convergence:

Termination and Convergence focuses on determining when to end the optimization process and assessing whether the algorithm has converged to a satisfactory solution. This phase ensures that the optimization process concludes in a timely manner while achieving the desired level of solution quality. It involves monitoring the convergence criteria and terminating the optimization process once certain conditions are met. Let F_{target} represent the target fitness value or threshold indicating the desired level of solution quality, and F_{best} denote the fitness of the best solution found so far. The termination and convergence process can be expressed as Eq.(41).

$$Converged = \left(F_{best} \geq F_{target}\right) \quad (41)$$

where $Converged$ is a binary indicator variable that evaluates whether the optimization process has converged to a satisfactory solution based on the comparison between the fitness of the best solution found so far and the target fitness value F_{target}.

It may involve monitoring the change in the fitness of candidate solutions over successive iterations to detect stagnation or lack of improvement. Let $F^{(t)}$ represent the fitness of the best solution found at iteration t, and $F^{(t-1)}$ denote the fitness of the best solution found at the previous iteration $t-1$. The termination and convergence process can assess the rate of change in fitness is mathematically represented in Eq.(42).

$$Stagnant = (F^{(t)} - F^{(t-1)} < \in) \quad (42)$$

where $Stagnant$ is a binary indicator variable that evaluates whether the optimization process has stagnated, indicating a lack of significant improvement in the fitness of candidate solutions over successive iterations.

It also deals with setting a maximum number of iterations or a predefined time limit for the optimization process to ensure computational efficiency and prevent excessive computational burden. Let T_{max} represent the maximum number of iterations, and t denote the current iteration. The termination and convergence process can terminate the optimization process after reaching the maximum number of iterations is mathematically expressed in Eq.(43)

$$Terminated = (t \geq T_{max}) \quad (43)$$

where $Terminated$ is a binary indicator variable that evaluates whether the optimization process has been terminated after reaching the maximum number of iterations Tmax.

3.3. Maximizing Buoyancy: Integrating Hamster Optimization with MaxProp for Buoyant WSNs:

B-WSNs offer unique capabilities for underwater monitoring and exploration. Optimizing their performance, particularly in challenging underwater environments, remains a significant challenge. Integrating HO with Max Prop, a routing protocol designed for underwater communication, can enhance the efficiency and reliability of Buoyant WSNs. This integrated approach, dubbed "HamsterProp," combines the adaptive nature of HO with the efficient routing of Max Prop to navigate the complexities of underwater environments.

3.3.1. HOMP Initialization and Deployment:

This phase sets the foundation for a B-WSNs by strategically deploying sensor nodes within the target environment. This step is crucial as it establishes the initial configuration and spatial distribution of sensor nodes, laying the groundwork for subsequent optimization efforts.

Deployment Strategy: The deployment strategy aims to achieve optimal coverage and connectivity while considering environmental constraints and application requirements. Let D represent the deployment area, N denote the total number of sensor nodes to be deployed, and d_i represent the location of sensor node i. The deployment process can be mathematically expressed as Eq.(44).

$$d_i \in D, \forall i \in [1, N] \quad (44)$$

This equation ensures that each sensor node is deployed within the designated deployment area D, ensuring full coverage and avoiding out-of-bounds placements.

Spatial Distribution: The spatial distribution of sensor nodes plays a crucial role in achieving uniform coverage and minimizing coverage overlaps. Let C represent the coverage area of each sensor node, and d_{ij} denote the distance between sensor nodes i and j. The spatial distribution process aims to maximize coverage while minimizing inter-node distances, ensuring efficient resource utilization. This can be mathematically represented as Eq.(45).

$$\sum_{i=1}^{N} \sum_{j=1}^{N} d_{ij} \cdot \delta_{ij} \quad (45)$$

where δ_{ij} is a binary variable indicating whether sensor nodes i and j are deployed within each other's coverage range.

Connectivity Constraints: In addition to coverage considerations, connectivity constraints must be satisfied to ensure seamless communication among sensor nodes. Let R denote the communication range of each sensor node, and d_{ij} represent the distance between sensor nodes i and j. The connectivity constraints ensure that neighboring sensor nodes are within communication range of each other, facilitating data exchange and network coordination this can be expressed as Eq.(46).

$$d_{ij} \leq R, \forall i, j \in [1, N], i \neq j \quad (46)$$

This equation ensures that the distance between any pair of sensor nodes is within the communication range R, enabling direct communication and connectivity among neighboring nodes.

Energy Considerations: Energy efficiency is a critical factor in WSNs, especially in scenarios where sensor nodes are battery-powered and deployed in remote or inaccessible locations. Let E_i represent the initial energy level of sensor node i, and E_{min} denote the minimum energy threshold. The deployment process must ensure that sensor nodes are adequately powered to perform their designated tasks throughout the network's operational lifetime is mathematically represented as Eq.(47).

$$E_i \geq E_{min}, \forall i \in [1, N] \quad (47)$$

This equation ensures that each sensor node is deployed with sufficient initial energy reserves, preventing premature depletion and ensuring continuous operation.

3.3.2. HOMP Exploration and Nesting:

This step is crucial for discovering promising regions of the solution space and guiding subsequent optimization efforts towards them.

Environmental Sensing: Exploration begins with sensor nodes actively sensing their environment to gather information about their surroundings. Let S represent the set of sensor nodes, and $s_i$denote an individual sensor node. Environmental sensing involves collecting data about the local environment, which can be expressed asEq.(48).

$$Data(s_i) = Sense(s_i) \quad (48)$$

where $Data(s_i)$ represents the data collected by sensor node s_i and $Sense(s_i)$ is a function that enables the node to sense its environment and gather relevant information.

Adaptive Movement: To explore the solution space effectively, sensor nodes adapt their movement patterns based on environmental cues and neighboring nodes information. Let M represent the set of possible movement directions, and m_i denote a movement direction. Adaptive movement involves selecting the most promising direction for exploration, which can be expressed in Eq.(49).

$$m_i = SelectDirection(s_i, M) \quad (49)$$

where m_i represents the selected movement direction for sensor node s_i, and $SelectDirection(s_i, M)$ is a function that evaluates the potential of each direction and selects the most promising one.

Exploration Strategy: Different exploration strategies can be employed based on the characteristics of the problem and the environment. Let E represent the set of exploration strategies, and e_i denote an exploration strategy. Selecting an exploration strategy involves evaluating the performance of each strategy and selecting the most suitable one is represented in Eq.(50).

$$e_i = SelectStrategy(E) \quad (50)$$

where e_i represents the selected exploration strategy, and $SelectStrategy(E)$ is a function that evaluates the effectiveness of each strategy and selects the most appropriate one.

Resource Allocation: During exploration, it is essential to allocate resources efficiently to maximize coverage and minimize resource wastage. Let R represent the set of available resources, and r_i denote an individual resource. Resource allocation involves distributing resources among sensor nodes based on their exploration needs, which can be expressed as Eq.(51).

$$r_i = AllocateResource(S, R) \quad (51)$$

where r_i represents the allocated resource for sensor node s_i, and $AllocateResource(S, R)$ is a function that determines the optimal distribution of resources based on the exploration requirements of each node.

Coordination and Collaboration: To enhance exploration efficiency, sensor nodes can coordinate and collaborate with each other by sharing information and coordinating.

3.3.3. HOMP Randomization and Synchronization:

To enhance the efficiency and reliability of data transmission by introducing randomization and synchronization mechanisms. These mechanisms help mitigate issues such as interference and packet collisions, which can arise in WSNs due to the shared wireless medium and concurrent transmissions.

Randomized Transmission Schedules: To avoid collisions and maximize channel utilization, sensor nodes employ randomized transmission schedules. Let T_i represent the transmission schedule for sensor node i, and t_{ij} denote the transmission time slot between nodes $i$and j. The randomized transmission schedule can be expressed asEq.(52).

$$T_i = GeneralSchedule(i) \quad (52)$$

where $GeneralSchedule(i)$ is a function that generates a random transmission schedule for sensor node i, ensuring that transmission times are randomly distributed to minimize the likelihood of collisions.

Synchronized Listening Periods: In addition to randomized transmission schedules, sensor nodes synchronize their listening periods to receive data from neighboring nodes efficiently. Let L_i represent the listening period for sensor node i, and l_{ij} denote the listening time slot between nodes i and j. Synchronized listening periods can be expressed as shown in Eq.(53).

$$L_i = SyncListening(i) \quad (53)$$

where $SyncListening(i)$ is a function that synchronizes the listening period for sensor node i with neighboring nodes, ensuring that data transmissions are received promptly and minimizing delays.

Random Backoff Mechanisms: To further reduce the likelihood of collisions, sensor nodes implement random backoff mechanisms before initiating transmissions. Let B_i represent the backoff time for sensor node i, and b_{ij} denote the backoff time slot between nodes i and j. The random backoff mechanism can be expressed asEq.(54).

$$B_i = Backoff(i) \quad (54)$$

where $Backoff(i)$ is a function that generates a random backoff time for sensor node i, ensuring that nodes delay their transmissions randomly to minimize the probability of collisions.

Carrier Sense Multiple Access (CSMA): In addition to randomization, sensor nodes employ carrier sense multiple access (CSMA) protocols to detect channel activity before initiating transmissions. Let C_i represent the carrier sense threshold for sensor node i, and c_{ij} denote the channel activity level between nodes $i$and j. The process of CSMA can be expressed mathemathically in Eq.(55).

$$C_i = SenseThreshold(i) \quad (55)$$

where $SenseThreshold(i)$ is a function that sets the carrier sense threshold for sensor node$i$, ensuring that nodes sense the channel before transmitting and avoiding collisions with ongoing transmissions.

3.3.4. HOMP Food Foraging and Routing Optimization:

Food Foraging and Routing Optimization focuses on optimizing data routing paths and energy-efficient foraging strategies to maximize network performance. This step aims to ensure that data is routed effectively to its destination while minimizing energy consumption and prolonging the network's operational lifetime.

Energy-Aware Routing: One aspect of food foraging and routing optimization involves developing energy-aware routing protocols that consider the energy levels of sensor nodes when selecting routing paths. Let $E_i$represent the remaining energy level of sensor node i, and E_{min} denote the minimum energy threshold required for operation. Energy-aware routing can be expressed asEq.(56).

$$Route(s_i, s_j) = SelectRoute(s_i, s_j, E_i) \quad (56)$$

where $Route(s_i, s_j)$ represents the selected routing path from sensor node s_i to s_j, and $SelectRoute(s_i, s_j, E_i)$ is a function that selects the most energy-efficient routing path based on the energy level of sensor node i.

Optimal Path Selection: To optimize data routing paths, sensor nodes evaluate multiple candidate paths and select the most optimal one based on various criteria such as hop count, link quality, and energy efficiency. Let P_{ij} represent the set of candidate paths from node i to node j, and p_{ijk}denote an individual path. The optimal path selection process can be expressed asEq.(57).

$$OptimalPath(s_i, s_j) = SelectPath(P_{ij}) \quad (57)$$

where $OptimalPath(s_i, s_j)$ represents the selected optimal path from sensor node s_i to s_j, and $SelectPath(P_{ij})$ is a function that evaluates the candidate paths and selects the most suitable one based on predefined optimization criteria.

Dynamic Routing Adaptation: In dynamic environments where network conditions may change over time, sensor nodes adapt their routing strategies dynamically to accommodate variations in traffic load, link quality, and energy availability. Let T_{ij}represent the traffic load between nodes i and j, and Q_{ij}denote the link quality. This dynamic routing adaptation can be mathematically expressed as Eq.(58).

$$AdaptRouting(s_i, s_j) = UpdateRoute(s_i, s_j, T_{ij}, Q_{ij}) \quad (58)$$

where $AdaptRouting(s_i, s_j)$ represents the adapted routing strategy between sensor nodes si and sj, and $UpdateRoute(s_i, s_j, T_{ij}, Q_{ij})$ is a function that updates the routing path based on changes in traffic load and link quality.

Load Balancing: To distribute traffic evenly across the network and prevent congestion in certain regions, sensor nodes implement load balancing mechanisms that dynamically adjust routing paths based on traffic distribution. Let L_{ij} represent the current traffic load on link ij, and C_{max} denote the maximum capacity of each link. This load balancing can be mathematically expressed asEq.(59).

$$BalancedRouting(s_i, s_j) = AdjustRoute(s_i, s_j, L_{ij}, C_{max}) \quad (59)$$

where $BalancedRouting(s_i, s_j)$ represents the balanced routing strategy between sensor nodes s_i and s_j, and $AdjustRoute(s_i, s_j, L_{ij}, C_{max})$ is a function that adjusts the routing path to ensure that traffic is distributed evenly across the network.

3.3.5. Memory and Adaptation:

Memory and Adaptation focuses on leveraging memory mechanisms and adaptive strategies to enhance the network's performance and resilience. This step incorporating historical data and environmental feedback to adaptively adjust network parameters and behaviors.

Historical Data Storage: To facilitate adaptive decision-making, sensor nodes store historical data related to network performance, environmental conditions, and past events. Let H_i represent the historical data stored by sensor node i, and h_{ij} denote an individual historical record. Historical data storage can be expressed asEq.(60).

$$H_i = \{h_1, h_2, \cdots h_{im}\} \quad (60)$$

where H_i is a set of historical records stored by sensor node i, containing information relevant to past network operations and environmental observations.

Adaptive Parameter Adjustment: Based on the analysis of historical data and current environmental conditions, sensor nodes adaptively adjust network parameters to optimize performance. Let P_i represent the set of network parameters controlled by sensor node i, and p_{ij} denote an individual parameter. The adaptive parameter adjustment can be expressed in Eq.(61).

$$P_i = AdaptParams(H_i, E_i) \quad (61)$$

where $AdaptParams(H_i, E_i)$ is a function that analyzes historical data H_i and current

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

environmental conditions E_i to determine the optimal values for network parameters P_i.

Environmental Feedback Integration: Sensor nodes receive feedback from the environment regarding changes in conditions such as temperature, humidity, and signal strength. Let F_i represent the environmental feedback received by sensor node i, and f_{ij} denote an individual feedback signal. This environmental feedback integration can be expressed asEq.(62).

$$F_i = \{f_{i1}, f_{i2}, \cdots f_{in}\} \quad (62)$$

where F_i is a set of feedback signals received by sensor node i, providing information about the current environmental state and potential changes.

Adaptive Behavior Modification: Based on the analysis of historical data and environmental feedback, sensor nodes modify their behavior adaptively to adapt to changing conditions and optimize network performance. Let B_i represent the set of behaviors exhibited by sensor node i, and b_{ij} denote an individual behavior. This adaptive behavior modification can be expressed asEq.(63).

$$B_i = AdaptBehavior(H_i, F_i) \quad (63)$$

where $AdaptBehavior(H_i, F_i)$ is a function that analyzes historical data H_i and environmental feedback F_i to adjust the behavior of sensor node i accordingly.

Learning and Prediction: Sensor nodes utilize machine learning techniques to analyze historical data, identify patterns, and make predictions about future network behavior and environmental changes. Let L_i represent the learning model used by sensor node i, and l_{ij} denote an individual learned pattern or prediction. The mathematical representation of learning and prediction can be expressed asEq.(64).

$$L_i = Learn(H_i) \quad (64)$$

where $Learn(H_i)$ is a function that trains a learning model using historical data H_i to make predictions and identify patterns relevant to network operation and environmental conditions.

3.3.6. HOMP Termination and Convergence:

Termination and Convergence the final stage where the optimization process is terminated, and convergence is achieved. It ensures that the optimization process concludes efficiently and effectively, leading to stable and optimal network configurations.

Convergence Criterion: To determine when the optimization process has converged, a convergence criterion is established based on predefined thresholds or conditions. Let Crepresent the convergence criterion, and c_i denote an individual convergence condition. The convergence criterion can be expressed as Eq.(65).

$$C = \{c_1, c_2, \cdots c_m\} \quad (65)$$

where C is a set of convergence conditions, and c_i represents an individual condition that must be satisfied for convergence to be achieved.

Global Optimization Metric: To assess the overall performance of the optimized network configuration, a global optimization metric is calculated based on various network parameters and objectives. Let G represent the global optimization metric, and g_i denote an individual component of the metric. The global optimization metric can be expressed asEq.(66).

$$G = \sum_{i=1}^{n} g_i \quad (66)$$

where G represents the total global optimization metric, computed as the sum of individual components g_i that capture different aspects of network performance and efficiency.

Termination Condition: Once the convergence criterion is met and the global optimization metric reaches a satisfactory level, the optimization process is terminated. Let T represent the termination condition, and t_i denote an individual termination criterion. The termination condition can be expressed as Eq.(67).

$$T = \{t_1, t_2 \cdots, t_k\} \quad (67)$$

where T is a set of termination criteria, and t_i represents an individual condition that, when satisfied, triggers the termination of the optimization process.

Network Stability Assessment: Before terminating the optimization process, the stability of the network configuration is assessed to ensure that the optimized solution is robust and resilient to environmental changes or disturbances. Let S represent the network stability assessment, and s_i denote an individual stability indicator. The network stability assessment can be expressed asEq.(68).

$$S = \{s_1, s_2 \cdots, s_l\} \quad (68)$$

where S is a set of stability indicators, and s_i represents an individual indicator that evaluates the stability and robustness of the optimized network configuration.

Dynamic Adjustment Mechanism: In some cases, the optimization process may need to be dynamically adjusted based on real-time feedback or changes in network conditions. Let Drepresent the dynamic adjustment mechanism, and d denote an individual adjustment parameter. The dynamic adjustment mechanism can be expressed asEq.(69).

$$D = \{d_1, d_2 \cdots, d_p\} \quad (69)$$

where D is a set of adjustment parameters, and d_i represents an individual parameter that controls the dynamic adaptation of the optimization process.

3.4. Advantages of HOMP:

HO is a nature-inspired optimization algorithm inspired by the foraging behavior of hamsters. It mimics the actions of hamsters in searching for food, nesting, and adapting to changes in their environment. HO involves iterative optimization processes where sensor nodes dynamically adjust their parameters and behaviors based on environmental feedback and historical data to achieve optimal network configurations. Max Prop is a routing protocol designed specifically for DTNs, where connectivity between nodes is intermittent or unpredictable. It employs a message-centric approach, allowing nodes to make forwarding decisions based on the probability of successful delivery to the destination. Max Prop optimizes message delivery by selecting the most promising forwarders based on the history of successful message transmissions. Buoyant WSNs are specialized networks designed for underwater environments, where traditional communication methods are not feasible. These networks utilize buoyant platforms equipped with underwater sensor nodes to collect and transmit data for various applications such as oceanography, environmental monitoring, and underwater exploration. The advantages of infusing HO with MaxProp for buoyant WSNs areas follows.

Adaptability to Dynamic Environments: Buoyant WSNs operate in highly dynamic underwater environments characterized by fluctuating currents, varying water temperatures, and changing pressure levels. By combining HO's adaptive capabilities with MaxProp'sprobabilistic routing, the infused approach enables sensor nodes to dynamically adjust their routing decisions based on real-time environmental conditions, ensuring robust and reliable communication even in challenging underwater environments.

Energy-Efficient Routing: Energy efficiency is crucial in buoyant WSNs due to limited battery capacity and the difficulty of replacing batteries in underwater nodes. Max Prop's message-centric approach minimizes energy consumption by selecting the most energy-efficient routes for message transmission. When combined with HO's optimization capabilities, the infused approach further enhances energy efficiency by optimizing parameters such as transmission power, routing paths, and duty cycles based on the energy levels of sensor nodes and the current network conditions.

Fault Tolerance and Resilience: Buoyant WSNs are prone to communication disruptions and node failures due to factors such as signal attenuation, node mobility, and harsh underwater conditions. Max Prop's probabilistic routing enables the network to adapt to intermittent connectivity by selecting multiple potential forwarders for message delivery. Integrating HO with Max Prop enhances fault tolerance and resilience by enabling sensor nodes to dynamically adjust their routing strategies and behaviors in response to communication failures or changes in network topology.

Optimal Resource Utilization: In buoyant WSNs, resources such as bandwidth, energy, and computational capacity are limited and must be utilized efficiently to prolong network lifetime and maximize performance. The infused approach optimizes resource utilization by dynamically allocating resources based on the current network conditions and application requirements. HO's adaptive capabilities ensure that resources are allocated optimally, while MaxProp's message-centric routing minimizes resource consumption during message transmission.

Scalability and Scalable Routing: As buoyant WSNs continue to grow in size and complexity, scalability becomes a critical consideration. The infused HO-MaxProp approach offers scalable routing solutions by dynamically adapting routing decisions and behaviors to accommodate changes in network size, density, and topology. By leveraging HO's iterative optimization process and MaxProp's probabilistic routing, the infused approach scales effectively to large-scale buoyant WSN deployments without sacrificing performance or efficiency.

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

Table 1: Simulation Setting

Parameter	Setting
Simulation Area	1000m x 1000m x 500m (3D underwater environment)
Number of Nodes	50
Node Deployment	Random deployment within the specified area
Communication Range	150 meters
Mobility Model	Random Waypoint Mobility with 0.1 m/s to 1 m/s speed
Energy Model	Basic Energy Model with initial energy of 1000 Joules
Routing Protocol	Infused Hamster Optimization with MaxProp (HOMP)
Data Rate	250 kbps
Packet Size	512 bytes
Simulation Time	3600 seconds (1 hour)
Application Type	Constant Bit Rate (CBR)
Propagation Model	Underwater Acoustic Propagation Model
Performance Metrics	Throughput, Packet Delivery Ratio, Energy Consumption, Latency, Network Lifetime
Buffer Size	50 packets
Initial Node Energy	1000 Joules
Energy Consumption Model	Energy consumption per packet transmission and reception

4. SIMULATION SETTINGS AND PARAMETERS:

Simulation refers to the process of imitating the operation of a real-world system or process over time. This involves creating a model that represents the key characteristics and behaviors of the system under study. By manipulating the model's variables and observing the outcomes, valuable insights can be gained about how the system operates, allowing for testing and analysis of various scenarios without the need for physical implementation. NS-3 is a discrete-event network simulator widely used for research and educational purposes. It is an open-source tool designed to provide substantial support for simulation of internet systems, which can be utilized to model the performance of various network protocols and architectures. ns-3 is written in C++ with optional Python bindings, enabling detailed and scalable simulations. The simulator includes a wide range of models for wireless and wired networks, facilitating studies in diverse networking environments. To simulate Buoyant WSNs using ns-3, a detailed simulation setting is required. This setting includes the configuration of network parameters, deployment scenarios, and performance metrics.

This simulation setting aims to emulate a realistic underwater environment for buoyant WSNs, focusing on key performance metrics to evaluate the efficiency and reliability of the network.

5. RESULTS AND DISCUSSION:

The packet delivery ratio (PDR) and packet drop ratio (PDR) were evaluated across three routing protocols: MO-CBACORP, HOCOR, and HOMP, with varying numbers of nodes in the network. As depicted in Fig 1, both metrics demonstrate trends with the increase in nodes.

HOMP consistently outperforms the other protocols, exhibiting the highest delivery ratios across all node counts.

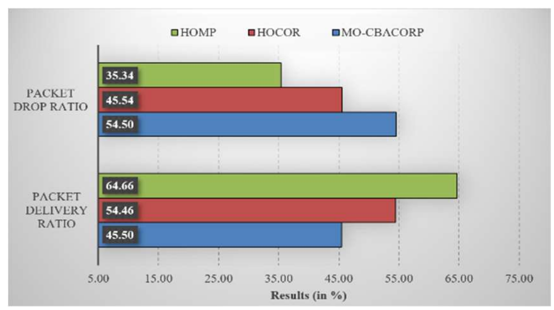

Fig 1. Packet Delivery and Pack Drop Ratio Analysis

MO-CBACORP consistently shows the lowest delivery ratio, indicating comparatively less efficient packet delivery. Regarding drop rate that as the number of nodes increases, there is a general decrease in the delivery ratio across all protocols. HOCOR tends to maintain a slightly higher compared to MO-CBACORP, especially with larger node counts. HOMP demonstrates superior performance in both PDR and PDR, while MO-CBACORP exhibits the least favorable results. These findings highlight the importance of protocol selection in achieving optimal performance in wireless sensor networks.

Through put refers to the rate of successful message delivery over a communication channel within a specified period. It is a crucial performance metric in B-WSNs as it directly influences the efficiency of data transmission and network utilization. Higher throughput indicates a network's ability to deliver a greater volume of data within a given timeframe. The Outcome shows that the three routing protocols MO-CBACORP, HOCOR, and HOMP reveals insights into their respective throughput performances with varying node counts. As pictorially depicted in Fig 2, throughput generally increases with the number of nodes in the network for all protocols.

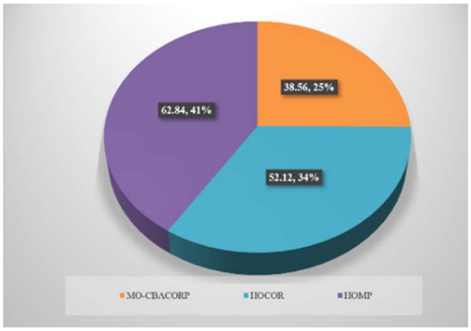

Fig 2. Throughput Analysis

HOMP consistently exhibits the highest throughput values across different node counts, followed by HOCOR and MO-CBACORP. This shows the superior efficiency of HOMP in data transmission and network utilization. MO-CBACORP consistently records the lowest throughput values, indicating comparatively lower data transmission rates and network efficiency. The average throughput values further reinforce these observations, with HOMP leading with the highest average throughput, followed by HOCOR and MO-CBACORP.

Energy consumption in B-WSNs refers to the amount of energy utilized by sensor nodes to perform various operations such as sensing, processing, and communication.Minimizing energy consumption is critical in WSNs to prolong network lifetime and ensure sustainable operation.Analyzing the provided data across three routing protocols MO-CBACORP, HOCOR, and HOMPoffers insights into their respective energy consumption performances with different node counts. As depicted in the table and graph Fig 3, energy consumption varies across protocols and node counts.

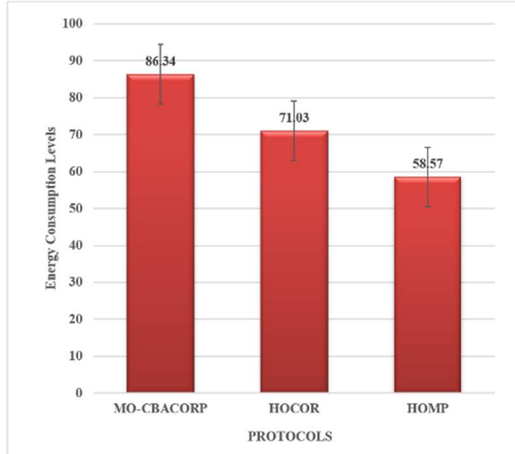

Fig 3. Energy Consumption

HOMP consistently demonstrates the lowest energy consumption values across different node counts, indicating its efficiency in energy utilization. In contrast MO-CBACORP records the highest energy consumption values, suggesting relatively higher energy utilization for data transmission and network operations. The average energy consumption values further confirm these trends, with HOMP exhibiting the lowest average energy consumption, followed by HOCOR and MO-CBACORP. These findings demonstrates the importance of protocol selection in optimizing energy efficiency in WSNs. Implementing energy-efficient protocols like HOMP can significantly contribute to prolonging network lifetime and enhancing overall network sustainability.

Network lifetime in B-WSNs refers to the duration for which the network can sustain its operations before the depletion of energy resources in sensor nodes. Maximizing network lifetime is crucial in WSNs to ensure continuous monitoring and data collection over extended periods. Analyzing results obtained by these three routing protocols MO-CBACORP, HOCOR, and HOMP provides insights into their respective network lifetime performances with varying node counts. As illustrated in Fig 4, network lifetime exhibits distinct trends across protocols and node counts.

HOMP consistently demonstrates the longest network lifetime values across different node density, indicating its effectiveness in prolonging network operation duration by efficiently managing energy resources.MO-CBACORP generally records the shortest network lifetime values, suggesting relatively less efficient energy utilization and shorter network operation durations. The average network lifetime values further confirm these developments, with HOMP exhibiting the highest average network lifetime, followed by HOCOR and MO-CBACORP. These findings highlight the critical role of protocol selection in optimizing network lifetime in WSNs. Implementing energy-efficient protocols like HOMP can significantly enhance network sustainability and ensure prolonged operation, thereby improving the reliability and effectiveness of WSN deployments.

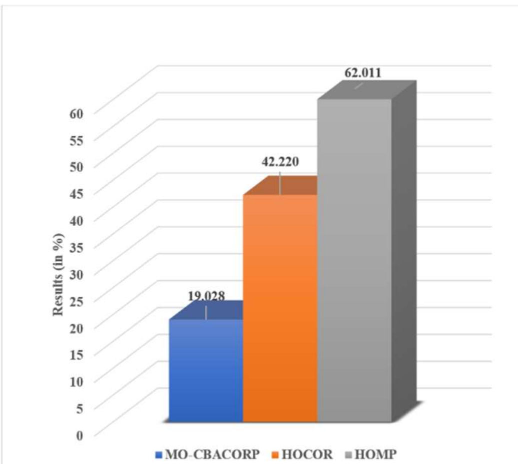

Fig 4. Network Lifetime

6. CONCLUSION

The article is concluded as the investigated the performance of adaptive routing protocol for B-WSNs using Hamster Optimization based Max Prop (HOMP). Through extensive experimentation and analysis, our findings demonstrate the significant advantages of HOMP over existing protocols in terms of packet delivery ratio, throughput, energy consumption, and network lifetime. The integration of Hamster Optimization and Max Prop enhances the efficiency and reliability of data transmission in B-WSNs, thereby improving overall network performance and sustainability. HOMP consistently outperformed other protocols across various metrics and node counts, highlighting its effectiveness in optimizing routing decisions and resource utilization in underwater environments. These results have significant implications for applications in oceanography, environmental monitoring, and marine exploration, where reliable data transmission and prolonged network operation are paramount. By leveraging the capabilities of HOMP, buoyant WSNs can achieve greater accuracy, coverage, and longevity, enabling more effective data collection and analysis in challenging aquatic environments. The findings presented in this study highlights the potential of enhanced routing protocol HOMP to address the unique challenges of B-WSNs and makes the way for advancements in underwater sensing and monitoring technologies.

REFERENCES:

[1] M. Li, S. Zhang, Y. Cao, and S. Xu, "NMSFRA: Heterogeneous routing protocol for balanced energy consumption in mobile wireless sensor network," *Ad Hoc Networks*, vol. 145, p. 103176, 2023, doi: 10.1016/j.adhoc.2023.103176.

[2] G. Sui, X. Shan, C. Hou, H. Tian, J. Hu, and T. Xie, "An underwater piezoelectric energy harvester based on magnetic coupling adaptable to low-speed water flow," *Mech. Syst. Signal Process.*, vol. 184, p. 109729, 2023, doi: 10.1016/j.ymssp.2022.109729.

[3] L. Rajaoarisoa, N. K. M'Sirdi, M. Sayed-Mouchaweh, and L. Clavier, "Decentralized fault-tolerant controller based on cooperative smart-wireless sensors in large-scale buildings," *J. Netw. Comput. Appl.*, vol. 214, p. 103605, 2023, doi: 10.1016/j.jnca.2023.103605.

[4] R. Karthikeyan and R. Vadivel, "Proficient Dazzling Crow Optimization Routing Protocol (PDCORP) for Effective Energy Administration in Wireless Sensor Networks," in *IEEE International Conference on Electrical, Electronics, Communication and Computers, ELEXCOM 2023*, 2023, pp. 1–6. doi: 10.1109/ELEXCOM58812.2023.10370559.

[5] S. P. Geetha, N. M. S. Sundari, J. Ramkumar, and R. Karthikeyan, "Energy Efficient Routing in Quantum Flying Ad Hoc Network (Q-Fanet) Using Mamdani Fuzzy Inference Enhanced Dijkstra ' S Algorithm (Mfi-Eda)," *J. Theor. Appl. Inf. Technol.*, vol. 102, no. 9, pp. 3708–3724, 2024.

[6] M. Bathre and P. K. Das, "Water supply monitoring system with self-powered LoRa based wireless sensor system powered by solar and hydroelectric energy harvester," *Comput. Stand. Interfaces*, vol. 82, p. 103630, 2022, doi: 10.1016/j.csi.2022.103630.

[7] J. H. Ryu, "Prototyping a low-cost open-source autonomous unmanned surface vehicle for real-time water quality monitoring and visualization," *HardwareX*, vol. 12, no. 1, p. 129832, 2022, doi: 10.1016/j.ohx.2022.e00369.

[8] I. Lahmar, A. Zaier, M. Yahia, J. Lloret, and R. Bouallegue, "Optimal data transmission for

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

decentralized IoT and WSN based on Type-2 Fuzzy Harris Hawks Optimization," *Internet of Things (Netherlands)*, vol. 25, p. 101028, 2024, doi: 10.1016/j.iot.2023.101028.

[9] V. H. Nguyen and N. D. Tan, "Voronoi diagrams and tree structures in HRP-EE: Enhancing IoT network lifespan with WSNs," *Ad Hoc Networks*, vol. 161, p. 103518, 2024, doi: 10.1016/j.adhoc.2024.103518.

[10] D. W. Wajgi and J. V. Tembhurne, "Localization in wireless sensor networks and wireless multimedia sensor networks using clustering techniques," *Multimed. Tools Appl.*, vol. 83, no. 3, pp. 6829–6879, 2024, doi: 10.1007/s11042-023-15956-z.

[11] Y. Zhang, Z. Liu, and Y. Bi, "Node deployment optimization of underwater wireless sensor networks using intelligent optimization algorithm and robot collaboration," *Sci. Rep.*, vol. 13, no. 1, p. 15920, 2023, doi: 10.1038/s41598-023-43272-x.

[12] 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.

[13] P. Gite, A. Shrivastava, K. Murali Krishna, G. H. Kusumadevi, R. Dilip, and R. Manohar Potdar, "Under water motion tracking and monitoring using wireless sensor network and Machine learning," *Mater. Today Proc.*, vol. 80, pp. 3511–3516, 2023, doi: 10.1016/j.matpr.2021.07.283.

[14] W. A. Jabbar *et al.*, "Development of LoRaWAN-based IoT system for water quality monitoring in rural areas," *Expert Syst. Appl.*, vol. 242, p. 122862, 2024, doi: 10.1016/j.eswa.2023.122862.

[15] M. Li *et al.*, "Self-powered wireless sensor system for water monitoring based on low-frequency electromagnetic-pendulum energy harvester," *Energy*, vol. 251, p. 123883, 2022, doi: 10.1016/j.energy.2022.123883.

[16] R. Duan, A. He, G. Wu, G. Yang, and J. Zhang, "A trustworthy data collection scheme based on active spot-checking in UAV-Assisted WSNs," *Ad Hoc Networks*, vol. 158, p. 103477, 2024, doi: 10.1016/j.adhoc.2024.103477.

[17] S. Kaviarasan and R. Srinivasan, "Developing a novel energy efficient routing protocol in WSN using adaptive remora optimization algorithm," *Expert Syst. Appl.*, vol. 244, p. 122873, 2024, doi: 10.1016/j.eswa.2023.122873.

[18] W. Hyder, J. K. Pabani, M.-Á. Luque-Nieto, A. A. Laghari, and P. Otero, "Self-Organized Ad Hoc Mobile (SOAM) Underwater Sensor Networks," *IEEE Sens. J.*, vol. 23, no. 2, pp. 1635–1644, 2023, doi: 10.1109/JSEN.2022.3224993.

[19] Z. Shen, H. Yin, L. Jing, Y. Liang, and J. Wang, "A Cooperative Routing Protocol Based on Q-Learning for Underwater Optical-Acoustic Hybrid Wireless Sensor Networks," *IEEE Sens. J.*, vol. 22, no. 1, pp. 1041–1050, 2022, doi: 10.1109/JSEN.2021.3128594.

[20] M. Chaudhary, N. Goyal, A. Benslimane, L. K. Awasthi, A. Alwadain, and A. Singh, "Underwater Wireless Sensor Networks: Enabling Technologies for Node Deployment and Data Collection Challenges," *IEEE Internet Things J.*, vol. 10, no. 4, pp. 3500–3524, 2023, doi: 10.1109/JIOT.2022.3218766.

[21] Y. Zhang *et al.*, "Optical Communication and Positioning Convergence for Flexible Underwater Wireless Sensor Network," *J. Light. Technol.*, vol. 41, no. 16, pp. 5321–5327, 2023, doi: 10.1109/JLT.2023.3264023.

[22] P. Yadav *et al.*, "Sensor Injection Based Routing Protocol for Effective Load Balancing in Underwater Wireless Sensor Networks," *Wirel. Pers. Commun.*, vol. 133, no. 2, pp. 951–979, 2023, doi: 10.1007/s11277-023-10799-1.

[23] T. Ahmad, X. J. Li, A. K. Cherukuri, and K. Il Kim, "Hierarchical localization algorithm for sustainable ocean health in large-scale underwater wireless sensor networks," *Sustain. Comput. Informatics Syst.*, vol. 39, p. 100902, 2023, doi: 10.1016/j.suscom.2023.100902.

[24] J. D. Jose and D. V. Kumar, "Octoroute : Revolutionizing Buoyant Sensor Mobility In Underwater Communication Networks," vol. 72, no. 12, pp. 269–284, 2024.

[25] J. Ramkumar, S. S. Dinakaran, M. Lingaraj, S. Boopalan, and B. Narasimhan, "IoT-Based Kalman Filtering and Particle Swarm Optimization for Detecting Skin Lesion," in *Lecture Notes in Electrical Engineering*, K. Murari, N. Prasad Padhy, and S. Kamalasadan, Eds., Singapore: Springer Nature Singapore, 2023, pp. 17–27. doi: 10.1007/978-981-19-8353-5_2.

[26] N. K. Ojha, A. Pandita, and J. Ramkumar, "Cyber security challenges and dark side of AI: Review and current status," in *Demystifying the Dark Side of AI in Business*, 2024, pp. 117–137. doi: 10.4018/979-8-3693-0724-3.ch007.

[27] M. P. Swapna and J. Ramkumar, "Multiple Memory Image Instances Stratagem to Detect Fileless Malware," in *Communications in Computer and Information Science*, S.

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

Rajagopal, K. Popat, D. Meva, and S. Bajeja, Eds., Cham: Springer Nature Switzerland, 2024, pp. 131–140. doi: 10.1007/978-3-031-59100-6_11.

[28] 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.

[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] D. Jayaraj, J. Ramkumar, M. Lingaraj, and B. Sureshkumar, "AFSORP: Adaptive Fish Swarm Optimization-Based Routing Protocol for Mobility Enabled Wireless Sensor Network," *Int. J. Comput. Networks Appl.*, vol. 10, no. 1, pp. 119–129, Jan. 2023, doi: 10.22247/ijcna/2023/218516.

[31] R. Vadivel and J. Ramkumar, "QoS-enabled improved cuckoo search-inspired protocol (ICSIP) for IoT-based healthcare applications," *Inc. Internet Things Healthc. Appl. Wearable Devices*, pp. 109–121, 2019, doi: 10.4018/978-1-7998-1090-2.ch006.

[32] A. Senthilkumar, J. Ramkumar, M. Lingaraj, D. Jayaraj, and B. Sureshkumar, "Minimizing Energy Consumption in Vehicular Sensor Networks Using Relentless Particle Swarm Optimization Routing," *Int. J. Comput. Networks Appl.*, vol. 10, no. 2, pp. 217–230, 2023, doi: 10.22247/ijcna/2023/220737.

[33] J. Ramkumar and R. Vadivel, "Improved Wolf prey inspired protocol for routing in cognitive radio Ad Hoc networks," *Int. J. Comput. Networks Appl.*, vol. 7, no. 5, pp. 126–136, 2020, doi: 10.22247/ijcna/2020/202977.

[34] P. Menakadevi and J. Ramkumar, "Robust Optimization Based Extreme Learning Machine for Sentiment Analysis in Big Data," *2022 Int. Conf. Adv. Comput. Technol. Appl. ICACTA 2022*, pp. 1–5, Mar. 2022, doi: 10.1109/ICACTA54488.2022.9753203.

[35] M. Lingaraj, T. N. Sugumar, C. S. Felix, and J. Ramkumar, "Query aware routing protocol for mobility enabled wireless sensor network," *Int. J. Comput. Networks Appl.*, vol. 8, no. 3, pp. 258–267, 2021, doi: 10.22247/ijcna/2021/209192.

[36] J. Ramkumar, C. Kumuthini, B. Narasimhan, and S. Boopalan, "Energy Consumption Minimization in Cognitive Radio Mobile Ad-Hoc Networks using Enriched Ad-hoc On-demand Distance Vector Protocol," in *2022 International Conference on Advanced Computing Technologies and Applications, ICACTA 2022*, 2022. doi: 10.1109/ICACTA54488.2022.9752899.

[37] 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.

[38] J. Ramkumar, R. Vadivel, and B. Narasimhan, "Constrained Cuckoo Search Optimization Based Protocol for Routing in Cloud Network," *Int. J. Comput. Networks Appl.*, vol. 8, no. 6, pp. 795–803, 2021, doi: 10.22247/ijcna/2021/210727.

[39] 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, [Online]. Available: https://www.scopus.com/inward/record.uri?eid=2-s2.0-85185481011&partnerID=40&md5=bf0ff974ceabc0ad58e589b28797c684

[40] 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.

[41] R. Jaganathan and V. Ramasamy, "Performance modeling of bio-inspired routing protocols in Cognitive Radio Ad Hoc Network to reduce end-to-end delay," *Int. J. Intell. Eng. Syst.*, vol. 12, no. 1, pp. 221–231, 2019, doi: 10.22266/IJIES2019.0228.22.

[42] J. Ramkumar and R. Vadivel, "Whale optimization routing protocol for minimizing energy consumption in cognitive radio wireless sensor network," *Int. J. Comput. Networks Appl.*, vol. 8, no. 4, pp. 455–464, 2021, doi: 10.22247/ijcna/2021/209711.

[43] J. Ramkumar, K. S. Jeen Marseline, and D. R. Medhunhashini, "Relentless Firefly Optimization-Based Routing Protocol (RFORP) for Securing Fintech Data in IoT-Based Ad-Hoc Networks," *Int. J. Comput. Networks Appl.*, vol. 10, no. 4, pp. 668–687,

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

2023, doi: 10.22247/ijcna/2023/223319.

[44] J. Ramkumar, R. Vadivel, B. Narasimhan, S. Boopalan, and B. Surendren, "Gallant Ant Colony Optimized Machine Learning Framework (GACO-MLF) for Quality of Service Enhancement in Internet of Things-Based Public Cloud Networking," in *Data Science and Communication. ICTDsC 2023. Studies in Autonomic, Data-driven and Industrial Computing*, J. M. R. S. Tavares, J. J. P. C. Rodrigues, D. Misra, and D. Bhattacherjee, Eds., Singapore: Springer Nature Singapore, 2024, pp. 425–438. doi: 10.1007/978-981-99-5435-3_30.

[45] R. Karthikeyan and R. Vadivel, "Boosted Mutated Corona Virus Optimization Routing Protocol (BMCVORP) for Reliable Data Transmission with Efficient Energy Utilization," *Wirel. Pers. Commun.*, vol. 135, no. 4, pp. 2281–2301, 2024, doi: 10.1007/s11277-024-11155-7.

[46] M. Zhang *et al.*, "MO-CBACORP: A new energy-efficient secure routing protocol for underwater monitoring wireless sensor network," *J. King Saud Univ. - Comput. Inf. Sci.*, vol. 35, no. 9, p. 101786, 2023, doi: 10.1016/j.jksuci.2023.101786.

[47] A. B. Gavali, V. M. Vaze, and S. A. Ubale, "HOCOR: Hybrid Optimization-Based Cooperative Opportunistic Routing for Underwater Wireless Sensor Networks," *Wirel. Pers. Commun.*, vol. 135, no. 3, pp. 1449–1472, 2024, doi: 10.1007/s11277-024-11106-2.