IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received 17 January 2025, accepted 11 February 2025, date of publication 14 February 2025, date of current version 6 March 2025.

*Digital Object Identifier 10.1109/ACCESS.2025.3542302*

RESEARCH ARTICLE

# Smart Packet Delivery in Mobile Underwater Sensors Networks (M-CTSP)

**ABDUL MOID KHAN [1,2], MIGUEL-ÁNGEL LUQUE-NIETO [1,3], (Member, IEEE), AND ALI AKBAR SIDDIQUE [4]**

[1]Department Ingeniería de Comunicaciones, E.T.S.I. Telecommunication, University of Málaga, 29010 Málaga, Spain
[2]Electronic Engineering Department, Sir Syed University of Engineering and Technology, Karachi 75300, Pakistan
[3]Institute of Oceanic Engineering, University of Málaga, 29010 Málaga, Spain
[4]Department of Computer Science, Iqra University, EDC Campus, Karachi 75500, Pakistan

Corresponding author: Abdul Moid Khan (amkhan@uma.es)

This work was supported by Spanish Ministry of Science and Innovation through the Project ''NEMO4EX: Heterogeneous Underwater Networks for Collaborative, Dynamic, and Cost-Effective Exploration'' under Grant PID2023-146540OB-C43.

**ABSTRACT** This research is focused on leveraging the Travelling Salesman Problem (TSP) to establish the optimal route of getting from a source node to a destination node in an underwater dynamic location. In this context, all nodes are mobile and slightly adjust their positions due to water currents. Clustering is introduced (CTSP) to identify the best possible route, where cluster heads are chosen using the known criterium of center of mass of the cluster as well as residual energy. Packets travel from source to destination (nodes) in a multiloop process, visiting cluster heads nodes, with minimum energy consumption. The intention behind using TSP in this approach is thus to efficiently handle the dynamic movement of underwater nodes by conserving energy and ensuring delivery of packets can be relied upon. Additionally, to the movement-related challenges, this method also enhances network lifetime by providing a systematic and energy-efficient way of transmitting information through cluster heads. Simulations using the novel M-CTSP into the routing algorithm yields to promising results the M-CTSP protocol enhances routing execution with the use of dynamic clustering and traveling salesman optimization. This results in a 40% rise in energy use reduction, 15% improvement in packet delivery with a 30% increase in network lifetime when compared to normal clustering-based protocols: LEACH and EECBP-FOA. Experimental results prove that M-CTSP contributes extensively in terms of node mobility packet delivery and energy conservation in dynamic UWSN environments.

**INDEX TERMS** Underwater sensor networks, energy efficiency, traveling salesman problem, clustering, data routing.

## I. INTRODUCTION

Numerous wireless sensor nodes dispersed across the ocean floor make up underwater wireless sensor networks (UWSNs), providing a wide range of applications including resource discovery, surveillance, navigation, data collecting, and catastrophe avoidance. Since UWSNs use acoustic signals for communication, every sensor node is outfitted with an acoustic modem. These nodes can create networks even in the absence of infrastructure [1]. The sensor nodes' job is to keep an eye on the undersea environment, including temperature, pressure, salinity, depth, and distance, and relay the information they gather to a sink node (SN) over one or more hops. Positioned at the surface of the ocean, the SN can receive data via acoustic signals from underwater sensor nodes and transmit the obtained data via radio signals to devices in the terrestrial network [1], [2]. Radio waves encounter the issue of absorption in the undersea environment and rapidly weaken. For long-distance underwater communications, they are therefore unsuitable. This is because sound waves experience a low level of attenuation, dispersion, and absorption losses which makes them suitable for underwater communication systems [3]. Other shortcomings of the UWA channel are significant propagation delays, high error rate

The associate editor coordinating the review of this manuscript and approving it for publication was Xuebo Zhang.

and limited bandwidth. This makes energy-intensive computational models necessary to keep up with the performance of UWSNs while maintaining the efficiency of data packet delivery. Also, there are limits to recharging or redistributing sensor nodes since their energy storage is limited [4]. Therefore, in UWSNs, energy consumption and network lifetime become important issues. On one hand, terrestrial wireless sensor networks (TWSNs) use radio signals for transmission while on another hand UWSNs employ acoustic signals; this implies that some of the energy-efficient routing techniques used by TWSNs cannot be directly applied to UWSNs [5]. Furthermore, two-dimensional network models are commonly employed in TWSN while three-dimensional network models form part of UWSN which poses serious challenges to researchers involved in it. The clustering routing methods create groupings in the network [6], where every cluster is comprised of a cluster head node (CHN) and some cluster member nodes (CMNs). At the end of the creation process, those CMNs that are in one cluster send data to its CHN as per channel resource allocation made by it hence reducing collisions [7].

The Underwater Wireless Sensor Networks (UWSNs) have become crucial in oceanographic data collection, environmental monitoring, resource exploration, and military surveillance. UWSNs differ from terrestrial sensor networks, due to certain physical and operational limitations imposed by the underwater environment. The most used technique for communication is acoustic signaling; little bandwidth available, high propagation delay, and a great amount of energy consumed in the form of attenuation and absorption.

One of the critical challenges in UWSNs has been the **energy efficiency** since they often depend on battery-power source in remote conditions, while in underwater sensor networks the practical cranes to recharge or replace batteries may be highly limited. To minimize energy dissipation is thus an important goal when extending the lifetime of a network. Another problem to further complicate the task of routing is node mobility, as these sensor node entities are susceptible to displacement by currents running under water, leading to dynamic variations in the network topology. This mobility complicates route-finding processes, enhancing the probability of message delivery failure and packet losses [8].

Oceanographic data collection, environmental monitoring, resource exploration, and military surveillance have recently become important subjects of study in the field of underwater environments. To aid this activity, UWSNs are necessary to enable efficient and reliable transmission of information through huge and changing marine terrains [2]. Nevertheless, the UWSN design is a task full of difficulties because of specific challenges that include water current-induced high mobility among nodes, limited energy resources as well as tough acoustic signal propagation conditions posed by underwater settings [9]. One of the challenges in UWSNs includes developing efficient routing protocols that can adapt to the dynamic nature of underwater nodes while minimizing energy consumption. Given the distinct characteristics exhibited by the medium underwater, terrestrial routing protocols cannot be utilized and hence there must be new perspectives that should be developed for addressing these issues peculiar to such an environment. In this regard, the Travelling Salesman Problem (TSP) provides a good basis for tackling some aspects related to routing problems in UWSNs [10]. This is a well-known combinatorial optimization problem whose goal is to find the shortest possible path that goes through all the nodes and back to the initial node. Utilization of TSP principles can therefore help in designing routing protocols that can optimize the path that data should take from a source node to a destination node within dynamic underwater environments. Present work aims to use TSP as a means for finding an optimal route for data transmission while considering both mobile nodes and energy constraints in UWSNs [10], [11].

In our proposed approach, physical location and remaining energy levels provide dynamic clustering of nodes. CHNs are chosen depending on their closeness to the center of gravity of the cluster and the energy available to them so as not to waste any resources. Finally, data packets are sent from a source node to its designated CHN, which relays them via many other CHNs until they get to the destination. It aims to reduce and efficient power consumption with each transmission to increase the network's lifetime [10], [11].

When TSP is integrated into our routing algorithm [10], it offers great benefits. Initially, it brings up the issue that the current approach is not very systematic and does not adapt to the movement of underwater nodes. That means such a dynamic system as well can stop you from getting reliable data delivery since the positioning of underwater nodes can change over time or when they are on the move. Secondly, through only energy-efficient routes, our algorithm contributes to solving one of the main issues. The short lifetime of sensor nodes in UWSNs and finally our method makes the network more sustainable by using CHs as intermediaries letting packets be routed through optimal paths hence reducing latency and improving general network performance.

This paper presents the **Mobile-Clustering Traveling Salesman Protocol (M-CTSP)** to tackle the above challenges. M-CTSP implements the concepts of dynamic clustering along with the traveling salesman problem (TSP) to optimally find the routing path that lessens energy consumption while addressing node mobility. M-CTSP enhances network performance by delivering packets over the shortest and most energy-efficient route using TSP, thereby promoting extension of a network's operational period. Implementation of the proposed solution is done via detailed simulations which yield considerable improvements in energy efficiency, packet delivery ratio, and network lifetime. To sum up, a new TSP-based UWSN routing protocol introduced by this research paper contrariwise tries to meet those specific challenges that are set in UW environments. By providing an optimized route for data transmission we guarantee reliability and improve the live span of UWSNs, make sure that ocean monitoring and exploration systems get enough reliability and longest possible networking support.

The main contributions of this work can be summarized in:

- M-CTSP: A novel routing protocol using the principles of the traveling salesman problem (TSP) to UWSNs which considers the dynamic movement of underwater nodes (mobile nodes).
- Optimal finding of data transmission routes with power consumption per transmission minimized.

The remainder of this paper is structured as follows. We begin with a brief introduction to Underwater Wireless Sensor Networks (UWSN) in Section I. This is followed by an overview of the research motivation and literature review in Sections II and III. A technical overview of UWSN and the proposed Dynamic Updates method for handling variability are presented in Sections IV and V. Our proposed dynamic clustering TSP protocol (M-CTSP) and its optimized energy-efficient algorithm are discussed in Sections VI and VII.

The impact of drift on clustering and routing, including mobility models and research novelty, are examined in Sections VIII and IX. The experimental setup and key performance indicators (KPIs) are outlined in Sections X and XI. The results, discussion, and comparison of M-CTSP with other protocols, including extensions of previous work, are presented in Sections XII to XV. Finally, the conclusion of our research is summarized in Section XVI.

## II. RESEARCH MOTIVATION

The motivation for this study is that there exists an urgent need to enhance the efficiency and reliability of Underwater Wireless Sensor Networks (UWSNs). UWSNs will be critical in support of applications like underwater environmental monitoring, disaster prevention, and military operations. However, deploying UWSN has been very challenging because of energy constraints, dynamic node mobility, and the harsh underwater environment.

Energy consumption matters because in UWSNs, the nodes are powered mostly by batteries, which are often difficult to recharge or replace. With high energy consumption, the energy reserves of nodes fast become depleted, reducing the efficient lifetime of the network and the reliability of data transmission. Another major challenge is the mobility of the nodes. The underwater currents drift them, hence disrupting the already established communication paths, causing high packet loss, and increasing transmission delays.

To alleviate these problems, we propose using the Traveling Salesman Problem (TSP) as the central mechanism for routing optimization. TSP, which finds the shortest path to visit a set of points, is modified in our protocol to dynamically optimize the routing paths. The solution to TSP, if integrated with the Mobile-Clustering Traveling Salesman Protocol (M-CTSP), will minimize the total distance traveled for data transmission, with a resultant reduction in energy consumption and improvement in packet delivery efficiency. This dynamic clustering approach also caters for node mobility since optimal routes will be recalculated as nodes move, allowing for robust and energy-efficient communication in UWSNs. The study demonstrated how effectively M-CTSP mitigates network life and reliability challenges besetting critical UWSN deployment issues.

## III. LITERATURE REVIEW

UWSNs have come out as a critical technology for the exploration and monitoring of underwater environments. They are applicable in gathering oceanographic data, environmental monitoring, finding resources, and military surveillance. Consequently, the design and modelling of effective routing protocols is highly essential due to the unique characteristics of the underwater environment such as limited bandwidth, high propagation delay, and energy constraints [2].

This literature review looks into the most recent research on energy-efficient and robust clustering routing protocols for mobile UWSNs. Traditional routing protocols designed for terrestrial networks are not well-suited for UWSNs due to the challenges. Consequently, researchers have proposed various energy-efficient clustering routing protocols specifically for underwater environments [12]. A few of them summarized below are studies focused on this literature review.

To address energy constraint concerns in UWSN, Rajini Et al 2021 introduced Energy Efficient Cluster Based Routing Modified Fairly Optimization Protocol (EECBP-FOA). For clustering, it employs the K-means algorithm. The modified Firefly Optimization Algorithm (FOA) is introduced for the optimal selection of CHNs, while the AODV protocol utilizes FOA for routing that extracts the best path for saving energy [13]. Banaeizadeh and Haghighat [14] have proposed Energy Efficient Data Gathering Scheme (EEDG) which works through three stages towards achieving energy-efficient routing. The first step involves dividing nodes into smaller sets controlled by temporary forwarder nodes ensuring balanced energy consumption. Forwarder nodes collect data from their subset nodes in one-hop communication every round. Secondly, MAC protocol improves collision rates and packet loss where regular nodes transmit their data only at given time slots to their forwarder nodes. Finally, the proposed graph structure reduces data delay from the entire network, meaning the mobile sink meets forwarder nodes based on a given degree in the graph [14].

Khan et al. [11] presented EECRAP based on remaining energy with node layer and section division to resolve these hotspot problems that can significantly increase battery life and distribute power evenly throughout the network. To save energy and extend network lifetime an innovative energy-efficient routing path selection is presented in which a subsequent forwarder node is selected based on its residual energy, layer division, and received signal strength.

The E-CDBR routing protocol [15] suggested by Chenthil Et al. reduces power consumption for UWSNs. In the beginning, nodes are randomly deployed, and a surface sink is placed at surface. Then, a clustering method is utilized to identify the ideal number of clusters before CHN selection within the cluster area. In this selection process they employed two criteria: depth coordination and in-cluster position. Once the CHN is selected, sends its data to the surface sink when the cluster area is within the transmission range.

Another routing protocol called EBECRP [16] employs the concept of using mobile sinks to balance the load on all nodes, avoiding die near sink nodes (low depth nodes) because of high load. The idea for employing clustering is to minimize multi-hopping which results in increased energy consumption. Selecting CHNs makes global communication reduce into locally compressed communication through the collection of information from one-hop neighbor nodes.

Khan et al. [17] introduced cDBR (Clustering-DBR) protocol as an improvement to the existing Depth Based Routing protocol (DBR). In DBR, routing is done based on the depth of sensor nodes; forwarder nodes are selected from among those with lesser depths and require more power than others. This is why closer-to-sink nodes die due to heavier burden. On the other hand, in cDBR, it selects one node at a time. To minimize energy loss, even distribution of load over all nodes is made. The energy usage per node can be equalized since every node has an equal chance of becoming CH (Cluster Head). That extends its lifetime.

Datta and Dasgupta [18] use multiple sinks' architecture to develop virtual layers that contain several sensor nodes and reduce the number of hops based on these sensors' nodes in a particular layer to reach the surface sink for EE-LCHR routing protocol. In this case, each layer has many clusters among which the cluster head keeps changing according to the fitness value of sensor nodes. In this protocol energy balance is better achieved because of the presence of virtual

**TABLE 1.** Comparative analysis between proposed and different clustering routing protocols.

| Protocol Name | Paper Name | Author Name & Year | Limitation | Clustering Method | Cluster Head Selection Criteria | Energy Efficiency & Network Lifetime |
|---|---|---|---|---|---|---|
| EECBP-FOA | Energy Efficient Cluster Based Routing using Modified Firefly Optimization Algorithm for UWSNs | [13] (2021) | Complexity of algorithm | K-means | Modified Firefly Optimization Algorithm (FOA) | High / Long |
| EEDG | An energy-efficient data gathering scheme in UWSNs using a mobile sink | [14] (2020) | Dependence on the mobile sink | Not specified | Temporary forwarder nodes | High/ Long |
| EECRAP | An Energy Efficient Clustering Routing Protocol based on Arithmetic Progression for UASNs | [11] (2024) | Complexity in managing layers and depths | Layer and depth division | Residual energy, layer division, received signal strength | High/ Long |
| E-CDBR | Energy Efficient Clustering Based Depth Coordination Routing Protocol For UWSNs | [15] (2022) | Initial random deployment may affect efficiency | Clustering approach | Depth coordination and in-cluster position | High/ Long |
| EBECRP | An energy efficient and balanced energy consumption Cluster based routing protocol for UWSNs | [16] (2016) | May have a higher initial setup cost | Clustering | Not specified | High/ Long |
| cDBR | Clustering depth-based routing for UWSNs | [17] (2016) | May have a higher overhead due to clustering | Cluster-based approach | Equal probability for all nodes | High/ Long |
| EE-LCHR | Energy-efficient layered cluster head rotation-based routing protocol for UWSNs | [18] (2022) | Complexity in maintaining virtual layers and rotating CHs | Virtual layers | Fitness value of sensor nodes | High/ Long |
| LEACH | Low energy adaptive clustering hierarchy protocol A retrospective analysis | [19] (2017) | Random CH selection can lead to uneven energy distribution | Randomized | Probability function based on residual energy | Moderate |
| M-CTSP ** | Mobile Clustering Traveling Salesman Protocol (M-CTSP) | This work | Implementation complexity, dynamic clustering overhead | Dynamic clustering | Center of gravity and remaining energy levels | High/ Long |

** Proposed in this work.

layers and rotation of CHNs leading to an improved network lifetime.

Palan et al. [19] conducted a retrospective analysis on Low-Energy Adaptive Clustering formation and proposed Low-Energy Adaptive Clustering Hierarchy (LEACH) protocol. In LEACH cluster formation is carried out randomly whereby nodes become cluster heads using probability function considering their residual energy. Nodes relay data to their respective CHN who then transmits it to the sink node.

Mobile Clustering Traveling Salesman Protocol (M-CTSP) is our proposed protocol solution for solving the TSP problem by integrating CTSP [10] into a routing algorithm based on mobile nodes' physical location and remaining energy levels. CHNs are selected from central nodes in each cluster or those having adequate power supply. Incorporating TSP in CTSP for integration into routing algorithms, supports effective and reliable underwater monitoring and exploration systems thus minimizing power consumption per transmission minimizing latency reducing overall network performance [10].

Recent research provides great insight into the underwater communications and modeling of underwater wireless sensor networks (UWSNs). While Fattah et al. [2] offer a detailed summary of the UWSN challenges, specifically highlighting communication protocols and energy efficiency, Jouhari et al. [9] discuss the enabling technologies and Internet of Underwater Things to enhance communication reliability. Meanwhile, Qureshi et al. [23] talk about RF path loss and absorption across various water environments, calling for adaptive protocols.

## IV. TECHNICAL OVERVIEW OF ACOUSTIC UNDERWATER SENSORS NETWORK

UWSNs using acoustics waves are also known as UASNs. They are important tools in environment monitoring, marine research, and underwater exploration. Furthermore, these systems capture data from different water bodies like lakes, rivers, or oceans, hence another reason why they are significant. Nevertheless, the greatest challenge to UWSN's growth is due to the high-power consumption of sensor nodes. Consequently, this issue is still determined by energy saving as well as efficiency [20].

Sound waves are transmitted through water travel at low speed when compared to radio frequency waves. However, the advantage is that it can be used for long-distance communication in underwater environments. UWSNs require meeting performance metrics such as coverage, connectivity, and energy efficiency to operate effectively [2]. Sensors should ensure there is always reliable communication with the central sink node so that the network stays connected. Coverage concerns how far the sensors reach while considering fast-discharge batteries that cannot be recharged or replaced down under; energy efficiency becomes essential.

In underwater wireless sensor networks (UWSN), the acoustic communication channel is inherently variable due to environmental factors, such as temperature, salinity, water depth, and ocean currents. These factors cause fluctuations in sound velocity, leading to variation in signal propagation and attenuation. Consequently, nodes must not rely on the fact that their channel states will not change, since the effective communication range and quality of the signal may indeed change along the timeline and be also position dependent.

The variability of the acoustic channel has a direct impact on the distance estimation, since it is crucial for transmission power control and reliable communication between nodes. Although the assumption for our theoretical model that nodes estimate their distance to receiver accurately seems ideal, in practice, it is impossible in view of dynamic changes in the underwater environment. Thus, the protocols must act or react to updated conditions of the channel in order to maintain efficient and robust communication [21].

The complexity of UWSNs increases when nodes are not static. UWSNs are mostly comprised of many sensor nodes that are dynamic and scattered all over the sea. Any node is composed by an underwater communication module, a computer unit, and a set of sensors for collecting data underwater to transfer to either terrestrial stations or other nodes [22]. However, underwater communications experience many challenges due to physical factors like heavy signal distortion in water amplified by obstructions such as rocks, coral reefs, and marine fauna. Researchers have developed various routing protocols to solve this problem by improving data packet processing, reducing energy consumption, and prolonging network lifetime. Multi-hop routing is one widely used method whereby packets take numerous steps through intermediate nodes before reaching their intended destination [23].

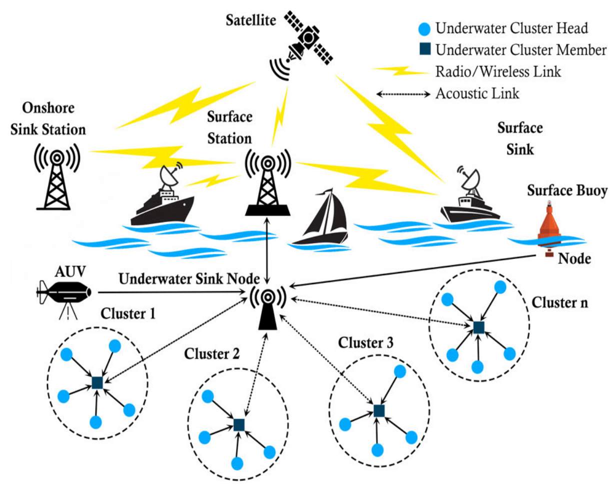



**FIGURE 1.** Smart underwater sensors network communication.

These schemes consist of graphs that represent how information flows within the network using edges (communication links) and vertices (network nodes). Hence, models consider parameters like transmission delay, energy use, and updating network topology among others to make efficient routing decisions for these networks [24]. Latest emerging algorithms have been used as optimization techniques to overcome routing issues in UWSN: genomic/genetic algorithms, particle swarm optimization, and linear/integer programming in case of mathematical programming methodologies [25], [26]. Ongoing research and development in the field of UWSN is crucial and essential for acoustic environmental monitoring and exploration [27].

## V. PROPOSED METHOD FOR HANDLIN VARIABILITY (DYNAMIC PATH UPDATES)

We propose the use of dynamic path updates. In this approach, nodes will periodically calculate the distance with respect to their neighboring nodes based on the latest signal propagation data. This periodic re-calculation is in its turn subject to dynamic adjustment of the transmit power to optimize energy use for reliable data transmission.

Four steps were included in this process: [28]

a. The nodes periodically exchange the acoustic signals among themselves and carry out time-of-flight measurements to obtain an estimate of the distance to their neighbors.
b. Using the new distance estimates, the transmit power of the nodes is adjusted to the least possible power required for reliable communication, thereby conserving energy.
c. The routing protocol uses updated distance data to re-examine the optimal paths to verify that the paths are still the most energy-efficient and reliable by present channel standards.
d. Dynamic path updates would enable the network to react effectively to changing environmental conditions, thereby alleviating the effects of channel variability in general. It also would boost overall performance and resilience of the UWSNs. In this way, the protocol is immune to channel variability in an "underwater realistic" context where static assumptions about channels become highly impractical.

## VI. MOBILE CLUSTERING TRAVEL SALESMAN PROTOCOL (M-CTSP)

The flowchart for the proposed M-CTSP protocol is shown in Fig. 2. The main difference from CTSP [10] is the ability for handling mobile nodes in the UWSN, limited to 3 meters (in x-y planes) as first approximation. By means of a smart algorithm (see Algorithm 1 below), the most efficient routing path is selected, the movement of nodes complicates the routing in the network, so it is critical to group the nodes and select the most efficient routing paths in order to keep up network effectiveness as well as extend the network's lifetime.

The process initiation starts with the system initialization phase in which all network parameters for the system are established. At this stage, the total number of nodes, the position they are initially placed at, the range of communication, and the energy each node has are determined.

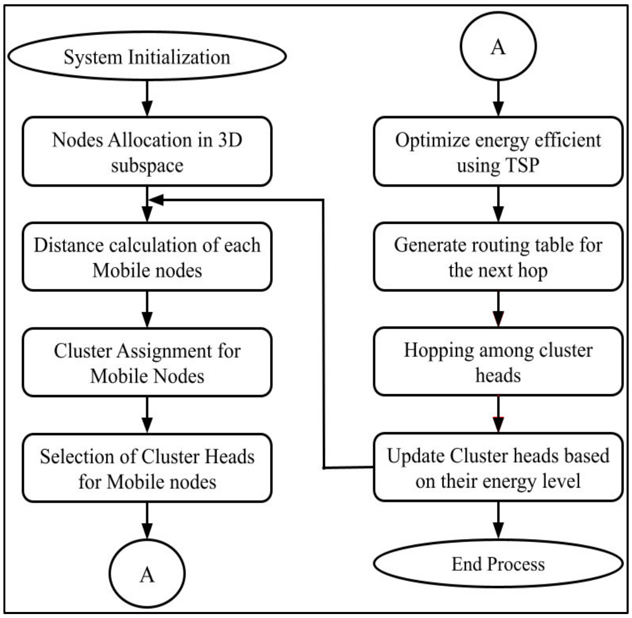



**FIGURE 2.** Flow diagram of M-CTSP.

### A. TRAVELLING SALESMAN PROBLEM

A simple optimization problem known as the traveling salesperson (TS) problem is to find the minimum cost tour through a set of cities or towns. Each city can be visited exactly once; the start and end points must coincide. NP-Hard TS problems are solved using heuristic algorithms, which are computationally intensive techniques for optimal solutions. An intense solution approach is required in this case due to the many nodes in the model, so the best way among these nodes on the net must be analyzed and identified properly. A matrix presents this problem by listing cities with distances between every pair of them. It could also be written as an optimization function.

We can view the TS Problem as a routing problem in a network where the packet plays the role of a salesman. It is seen as an efficient protocol for solving a routing problem about underwater acoustic communication as well as powerful model for optimizing routing and data collection in telecommunications and underwater sensor networks. The goal of the TSP is to minimize the total distance traveled, which can be represented mathematically by (1)-(3) [10].

$$D_{ij} = min \sum\nolimits_{i \in C} \sum\nolimits_{j \in C, j \neq 1} \bar{D}_{ij}.x_{ij} \tag{1}$$

$$\sum\nolimits_{j \in C, j \neq i} x_{ij} = 1, \forall i \in C \tag{2}$$

$$J = min \sum\nolimits_{i=1}^{N} min_k \left\| x_i - \mu_k \right\|^2, \tag{3}$$

with mathematical components defined above as:

1. **Set of nodes:** $N = \{1, 2, \ldots, n\}$, a set of cities the salesman needs to visit.
2. **Distance matrix:** $D = \left[d_{ij}\right]_{nxn}$, an $n \times n$ matrix, where $d_{ij}$ is the distance between two cities $i$ and $j$. We have $d_{ij} = d_{ij}$

for symmetric TSP instances, while for asymmetric TSP instances $d_{ij} \neq d_{ij}$ can occur.

3. **Decision variables:** $x_{ij}$, binary variables equal to 1 if the path between cities $i$ and $j$ is included in the solution, and 0 otherwise.

### B. DISPLACEMENT OF UNDERWATER NODES

An example of deployment of nodes in UWSN can be seen in Fig.3a. Every dot in this plot is a node placed in 3D position (abscissa, ordinate, and depth) where they are bound in a 500 × 500 × 500 units' cube as defined by the M-CTSP settings. Each node $i$ has an initial position $\vec{r}_i = (x_i, y_i, z_i)$, where $x_i, y_i, z_i$ have randomly generated values (in [0,500]). The movement of every node after every simulation step, is set by a displacement ($\Delta$) of every coordinate from its initial position, which follows a uniform distribution given by (4).

$$\Delta x, \Delta y, \Delta z \sim U[-3, 3] \tag{4}$$

This mobility model makes every node move in a confined sphere (of radius $\sqrt{27}$ m.) from its initial position. An example is shown in Fig. 3b once nodes have moved. The purple dots are the shifted nodes. Black lines connect each pair of points, representing displacement vectors for those nodes, while red labels next to each node indicate how far it has been displaced. When a node moves from $\vec{r}_{i,0}$ position to $\vec{r}_{i,1}$ position in the first simulation step, given by (5), its displacement $d_{i,1}$ may be expressed as (6).

$$\begin{cases} \vec{r}_{i,0} = (x_0, y_0, z_0) \\ \vec{r}_{i,1} = (x_0 + \Delta x, y_0 + \Delta y, z_0 + \Delta z) \end{cases} \tag{5}$$

$$|d_i| = \sqrt{(\Delta x)^2 + (\Delta y)^2 + (\Delta z)^2} \tag{6}$$

In the second simulation step, the initial point for node $i$ is $\vec{r}_{i,0}$ again, but a new set of random increments are chosen ($\Delta x$, $\Delta y$, $\Delta z$). This fact restricts to at most $\sqrt{27}$ meters every displacement of nodes, preserving the network's topology between iterations, and ensuring that routing paths and clusters will stabilize over time. This also prevents nodes from moving too far away from their initial positions, which reflects a realistic scenario in which nodes are not capable of moving great distances.

A popular clustering approach such as the K-means algorithm minimizes distances between nodes belonging to each cluster, calculated as in (6). The visual on the right of fig. 3 depicts how much each node can move around. There is some shifting in the position of the dots with movement size annotated.

The signal energy received in a data transfer by receiver node is proportional to distance (squared) from the transmitter node. So, the amount of energy consumed during transmission of packet to a node distant $d_i$ is defined by equation 7 [29]

$$E_{transmit} = E_{elec} \times L + E_{amp} \times L \times d_i^2 \tag{7}$$

where $E_{elec}$ signifies power consumed by electronics, $L$ refers to packet size in bits, $E_{\text{amp}}$ denotes power consumed by transmitter's amplifier. However, as nodes constantly change their positions the distances become different and so are the energy requirements for transmission also change. Mobility restriction of nodes allows a network to achieve a much preferable and stable energy consumption resulting in the longer life cycle of such network. Therefore, this highlights why there is a need for figure captions to state placement/mobility issues first and then during M-CTSP functionality. Thus, flexible node mobility must be properly regulated while its consequences may be analyzed from spatial or mathematical perspectives to attain optimal overall network performance within dynamic conditions.

### C. K-MEANS CLUSTERING

K-means (KM) clustering is an unsupervised machine learning algorithm that classifies a given dataset into several groups (clusters) based on their spatial proximity. KM Clustering Algorithm is a method to calculate the distance between each point and centroid of each cluster squared, with its main aim being to reduce WCSS ("within-cluster sum of squares") which is also recognized as inertia. The goal is to identify a clustering approach that efficiently reduces data variability within clusters, resulting in a more succinct representation of underlying patterns in the dataset. KM helps to reduce the sum of squared distances between each data point and its related centroids [30], [31], represented by its coordinates

$$\vec{\mu}_k = \left(\mu_{k,x}, \mu_{k,y}, \mu_{k,z}\right), \tag{8}$$

and uses Euclidean distance to find the nearest centroid $\vec{\mu}_k$ for each data point $\vec{r}_i$, assigning the $k$-cluster for that data point by means of this expression

$$k^* = \arg min_k \|\vec{r}_i - \vec{\mu}_k\|^2 \tag{9}$$

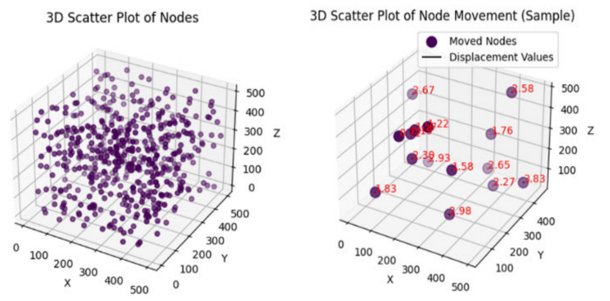
After this assignation of data points to clusters, every centroid is adjusted updating its position $\vec{\mu}_k$ for its cluster $k$(set of nodes $C_k$, composed by $N_k$ nodes) by means of

$$\vec{\mu}_k = (1/N_k) \cdot \sum_{i=1, \vec{r}_i \epsilon C_k}^{N_k} \vec{r}_i \tag{10}$$

### D. CLUSTER ASSIGNMENT TO THE NODES USING K-MEANS CLUSTERING

This will help in accurate distance calculation as it influences next clustering stage where nodes are grouped according to their nearness with one another. This is because clustering is aimed at minimizing intra-cluster distances and thus reducing the energy needed for communication among the cluster as provided in (10),

$$\min \sum_{k=1}^{K} \sum_{i=1, \vec{r}_i \epsilon C_k}^{N_k} \|\vec{r}_i - \vec{\mu}_k\|^2, \tag{11}$$




(a) (b)

**FIGURE 3.** Deployment (randomly) example for UWSN: (a) location of nodes, (b) displacement value for a subset of nodes (randomly chosen).

with $K$ the total number of clusters. However, the network remains optimized as nodes move during clustering process, assuming limited mobility of the nodes. The KM algorithm starts by initializing $K$ centroids that represent possible cluster centers. At the beginning, these centroids $(\mu_1, \mu_2 \ldots, \mu_K)$ are randomly placed within a 3D space with $K$ clusters. By means of Euclidean distance, each node $\vec{r}_i$ is assigned to the nearest centroid (e.g. cluster $k$) so that the distance function $d(\vec{r}_i, \vec{\mu}_k)$ can be minimized.

Given that network nodes are mobile, KM Clustering algorithm enables it to adapt dynamically to these movements through reassigning nodes into their most suitable clusters based on their new positions thereby keeping clusters compact and relevant even when node mobility is in place. This acts as an optimization mechanism for network operations. Data transmission within and between clusters can be efficiently managed by routing algorithms once these nodes are organized into clusters and this helps reduce computation overhead involved in handling many nodes. Energy conservation is also achieved by KM clustering since it makes sure that all nodes within one cluster are near each other, thus making communication less energy consuming especially in networks where there is limited amount of energy available from each node. Moreover, when the topology changes, the algorithm can adapt and optimize routing and energy consumption for the network. By continuously refining cluster assignments based on the current position of nodes, KM stabilizes the network so that it can function effectively in dynamic environments where node mobility is considered. This is important since MANETs are mobile ad-hoc networks where nodes often change their positions thus requiring real-time adaptation of a network upon any such event. The Fig. 4 shows how clusters were initially assigned in round 1, but clusters were reassigned after every round as a result of node movement.

### E. CLUSTER HEADS SELECTION

After clustering, cluster heads (CH) nodes are selected. The location of these nodes is nearest as possible of the center of each cluster by centroid mechanism, and aggregate data from single sources that can be sent outwards to other clusters. The process of selecting them is an optimization problem in which one should minimize power consumption across all nodes such as by choosing those with minimum residual energy while maximizing the lifetime of a network. The objective function used for choosing the CHs is given by

$$k = \arg max \sum_{i=1}^{K} \frac{E_k}{d(\vec{r}_i, \vec{r}_k)} \tag{12}$$

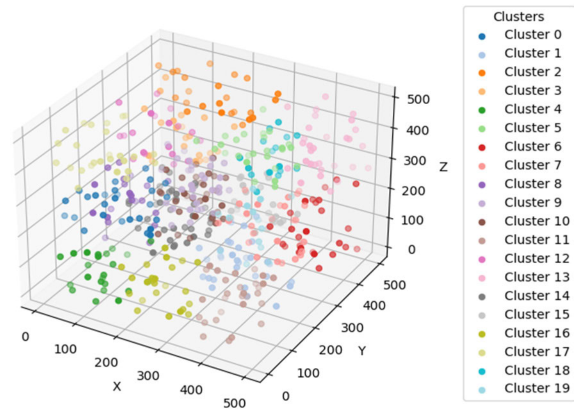



**FIGURE 4.** Cluster assignment using K-means.

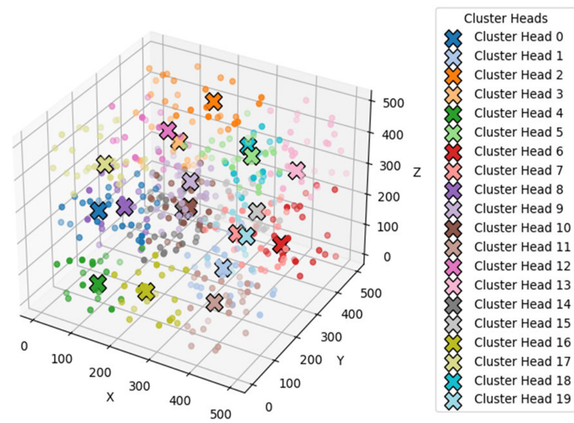



**FIGURE 5.** Selection of cluster head nodes.

With $E_k$ the residual energy of node $k$, and $d(\vec{r}_i, \vec{r}_k)$ the distance between nodes $i$, $k$. This expression indicates that more energetic and centrally located nodes will be selected as candidates for CH, thus achieving a balance between distortion and extension of operational life. An example of selected CHs is illustrated in Fig.5.

## VII. M-CTSP ENERGY EFFICIENT OPTIMIZATION

In M-CTSP selection of CHs marks the turning point, whereby energy efficiency is optimized via a modified form of the Travelling Salesman Problem (TSP). In traditional TSP, the aim is to find out the shortest path that covers all nodes exactly once; in this modification, we seek to have the route with least energy consumption while visiting only those nodes that contribute to making it shortest path to sink node. This technique works by using Dijkstra's algorithm which is used to determine the shortest path from a single source to a specific destination within a graph. However, unlike in normal cases like this one, such paths are even made for power cost besides distance as expressed by

$$\text{Minimize} \sum_{i=1}^{n} \sum_{j=1}^{n} c_{ij} x_{ij} \tag{13}$$

We introduce a cost function $c_{ij}$ which aims to evaluate the ''price'' of travelling from one node $i$ to another node $j$ while considering four criteria that must be optimized in the network. Each criterion is normalized by dividing it by the maximum among all valid nodes in set $V$ such that these are nodes that meet certain constraints like proximity to the source node and have sufficient depth. Moreover, we have introduced weights ($\alpha$, $\beta$, $\gamma$, $\epsilon$) for adjusting how important each criterion is. Finaly, the cost function is expressed as

$$\begin{aligned} c_{ij} = {} & \alpha \frac{d_{ij}}{\max_{i,j\epsilon V} d_{ij}} - \beta \frac{e_j}{\max_{k\epsilon V} e_k} \\ & - \gamma \frac{n_j}{\max_{k\epsilon V} n_k} + \varepsilon \frac{\left| r_{j,z} - r_{i,z} \right|}{\max_{i,j\epsilon V} \left| r_{j,z} - r_{i,z} \right|}, \end{aligned} \tag{14}$$

where:

- The first term includes the Euclidean distance between nodes $i$ and $j(d_{ij})$, normalized by dividing it with the maximum distance in the network,
- Second term is a cost to minimize as energy increases, considering the available at node $j$ as $e_j$ ,
- Third term takes into account the number of neighbors nodes (within 200 units) of node $j$ as $n_j$ , and
- The last term introduces the depth changes among nodes. The absolute change in depth between every pair of nodes is calculated by module function. In simple terms, it seeks to minimize the difference in depth such that cost will increase as difference in depths becomes larger. This means that paths that conserve or increase the $z$-value (depth) are preferable to those that go to lower depths.

This cost function is used by TSP solver to minimize total cost. While TSP usually concentrates on minimizing the distances between nodes in their computation, our M-CTSP also considers such other aspects as energy, neighbors and depth differences among others, expressed as the cost function $c_{ij}$ in (14), which calculates the cost of travel from node $i$ to node $j$ and encompasses the inter-node distance, energy availability in node $j$, neighborhood count of nodes within a 200-unit radius of node $j$ as well as change in depth between nodes $i$ and $j$ ($z$-value).

## VIII. IMPACT OF NODE DRIFT ON CLUSTERING AND ROUTING

In underwater wireless sensor networks (UWSNs), node mobility is among the main issues working against the performance of the network. These mobilities take effect because

**Algorithm 1** M-CTSP protocol

```
/* Parameters definition */
Max_mov = 3            /* Max. displacement x,y,z per iteration (m) */
EDR = 0.0045           /* Energy dissipation rate: 0.0045 J/100 m */
SN = (250,250,500)     /* Sink node coordinates (m) */
Radius = 300           /* Radius for node selection (m) */
NC = 20                /* Number of clusters */
n_iter = 5             /* Number of clustering runs */
α = 0.5                /* Weight for distance criterion */
β = 0.3                /* Weight for energy criterion */
γ = 0.1                /* Weight for neighbors' criterion */
ε = 0.1                /* Weight for depth criterion */
Δt = 5                 /* Time interval for dynamic updates (seconds) */
/* Functions definition */
function new_pos = rand_mov(ini_pos, Max_mov)
/* Random movement in x, y, z */
new_pos = ini_pos + uniform(−Max_mov,Max_mov)
end function
function dis = calc_dist(node1, node2)
/* Euclidean distance calculation */
dis = sqrt((node1x - node2x)^2 + (node1y - node2y)^2 + (node1z - node2z)^2)
end function
function cn = count_neighbors(node_i, V, dist)
/* Neighbors at distance ≤ dist from node_i inside set V */
   cn = sum(calc_dist(node_i, V(k)) ≤ dist for all V(k) ≠ node_i)
end function
function df = update_energy(df_ini, dist_traveled, EDR)
/* Value of energy after a movement */
energy_dissipated = EDR ∗ dist_traveled
df = df_ini - energy_dissipated
end function
/* New Function for Dynamic Path Updates */
function dynamic_update(node, Δt)
/* Periodic recalculation of distance and power adjustment */
  current_time = 0
  while current_time < simulation_time do
  updated_dist = calc_dist(node, node.nearest_neighbor)
  node.transmit_power = adjust_power(updated_dist)
   wait(Δt) /* Wait for Δt seconds */
  current_time + = Δt
  end while
end function
/* Adjust Transmit Power */
function power = adjust_power(distance)
/* Adjust transmit power based on the distance */
  power = base_power + power_factor ∗ distance
 end function
/* Network initialization: 500 nodes (random position) */
 V(1:500).pos_ini = [rand(500) rand(500) rand(500)]
/* M-CTSP Cost function */
 function c_ij = calc_cost(node_i, node_j, V, α, β, γ, ε)
  /* Euclidean distance between nodes i and j, normalized */
  d_ij = calc_dist(node_i, node_j)
  max_dist = max(calc_dist(i, j) for all i, j ∈ V)
dist_cost = α ∗ (d_ij / max_dist)
/* Energy availability at node j, normalized and weighted */
max_energy = max(node(k).df for all k ∈ V)
en_cost = -β ∗ (node_j.df / max_energy)
/* Neighbors' count (≤200 m), normalized and weighted */
n_j = count_neighbors(node_j, V, 200)
max_ne = max(count_neighbors(node_k, V, 200) for all k ∈ V)
ne_cost = -γ ∗ (n_j / max_ne)
/* Depth difference between nodes i and j, normalized */
  r_ij_z = abs(node_jz - node_iz)
max_depth_diff = max(abs(r_jz - r_iz) for all i, j ∈ V)
```

```
53:  de_cost = ε ∗ (r_ij_z / max_depth_diff)
54:            /∗ Total cost function ∗/
55:  c_ij = dist_cost + en_cost + ne_cost + de_cost
56: end function

                /∗ Best node selection within a radius ∗/

57: function best_node = select_bn(node, sel_nodes, radius, V)
58:  /∗ Filter nodes based on distance from selected nodes ∗/
59:  potential_nodes = [] /∗ Initialize list of potential nodes ∗/
60:  counter = 0
61:   for m = 1:size(sel_nodes) do
62:     potential_nodes[m] = [] /∗ Set initialization ∗/
63:     for n = 1:size(V) do
64:      dis = calc_dist(V[n], sel_nodes[m])
65:      if dis ≤ radius then
66:       counter + = 1
67:       potential_nodes[m] ≪ V[n] /∗ Add potential node ∗/
68: vec_dis_sel[counter] = dis /∗ Vector with distances ∗/
69: /∗ Search for subnodes≤radius around the current node ∗/
70:         subnode_count = 0 /∗ Counter for subnodes ∗/
71:         for p = 1:size(V) do
72:           sub_dis = calc_dist(V[p], V[n]) if p ≠ n
73:           if sub_dis ≤ radius then
74:            subnode_count + = 1
75:           end if
76:        end for
77:           potential_nodes[m].subnodes = subnode_count
78:         end if /∗ Line 65 ∗/
79:      end for /∗ n, Line 63 ∗/
80:   end for /∗ m, Line 61 ∗/
81:  /∗ Calculate cost for each potential node ∗/
82:   for i = 1:size(potential_nodes[m]) do
83:      cost[i] = calc_cost(sel_nodes[m], potential_nodes[m][i], V, α, β, γ, ε)
84 : end for
85: /∗ Select minimum cost node ∗/
86: [list, indexes] = sort(cost, ‘ascending’)
87: best_node[m] = potential_nodes[m][indexes [1]]
88: end for
89: end function /∗ best_node, Line 57 ∗/

                /∗ K-means clustering ∗/

90: for k = 1:n_iter do
91:  Cluster(1:NC) = [] /∗ Every cluster is a set of node struct ∗/
92:  Cluster ≪ K-means(V(1:500), NC) /∗ Creates clusters ∗/
93:  /∗ Cluster(m, n) is the n-th node in cluster m ∗/
94:  /∗ Cluster(m, n).df : energy of node n-th in cluster m ∗/
95:  Cluster(m, n).nc = m for all n /∗ label assigned: cluster number ∗/
96: end for

                /∗ Cluster Heads: max. energy nodes in cluster ∗/

97: for m = 1:NC do
98: [list, indexes] = sort(Cluster[m].df, ‘descending’)
99: Cluster[m].CH = Cluster[m, indexes [1]] /∗ CH: node with higher energy ∗/
100: end for

                /∗ Node movement ∗/

101:for n = 1:500 do /∗ For every node in the network ∗/
102: V[n].pos = rand_mov(V[n].pos_ini, Max_mov)
103: end for

                /∗ Path optimization ∗/

104: /∗ Select a random CH node as the starting node ∗/
105: CH_set = Cluster(1:NC).CH
106: current_node = random(CH_set) /∗ Select one of the set ∗/
107: path = [current_node] /∗ Initialize path ∗/
108: while true do
109:    /∗ Find the best node within radius from current_node ∗/
110:    best_node = select_bn(V, current_node, Radius)
111:    if not empty(best_node)
112:    dist_bn = calc_dist(current_node, best_node)
113:    /∗ Update energy of current_node based on the distance traveled ∗/
114:    current_node.df = update_energy(current_node.df, dist_bn, EDR)
115:    current_node.pos = best_node. pos /∗ Move to best node ∗/
116:    path ≪ current_node /∗ Append node to path ∗/
117:    end if

118:    if (current_node.pos == SN) or (calc_dist(current_node, SN) ≤ 200) then
119:       print “Packet reached Sink Node!”
120:       path ≪ SN /∗ Append sink node to path ∗/
121:       break
122:    end if

123: end while
```

of underwater currents working on sensor nodes, causing displacement from their original positions. The drift regulates diverse aspects of the network itself, ranging from clustering techniques to routing strategies [32].

### A. MOBILITY MODEL

Here, we consider mobile nodes for the mere reason that they are affected by underwater currents. Each node moves in a random location along the three-dimensional plane of the water, imitating real-life underwater movement. This drift is relatively small, typically within the range of few centimeters-to-meters, since we are obviously not considering the large-scale drifting mechanisms around large bodies of water. We represent the node displacement by considering that it would drift by an upper bound of 5 m at each iteration.

The mobility model allows for arbitrary motion with respect to different nodes in a region bound by their initial positions and maximum adaptation displacements (Max_movs). The movement of each node is continuously modelled with respect to their new positions determined at the end of a certain period. Accordingly, the model gives a very good input into the mobility-related issues faced by UWSNs operating in dynamic environments where nodes are not stationary [33].

### B. ANALYSIS OF MOBILITY

Drift of about 5 meters hardly alters the path loss, given the short distances involved, over which the attenuations of acoustic signals are relatively low. However, these changes cause shifts in the clustered structure of the network and consequently the routing algorithms.

#### 1) CLUSTERING IMPACT

This means the clusters are formed dynamically based on the proximity of nodes. As the nodes drift, the clusters will change, which could possibly entail frequent re-clustering. Resultant drifts might induce a node to switch from one cluster to another, thereby changing the assignments of cluster head(CH). Selection of a cluster head depends upon factors such as distance, energy levels, and the number of neighbors in contact; thus, the movement of a node will determine the selection of a cluster head [34]. Figure 6 shows that at drift of 3m cluster begin to change significantly. It helps to understand the effect of drift on cluster positions.

#### 2) ROUTING IMPACT

Routing protocols based on TSP depend on accurate node positions in order to find optimal paths. Node drift alters the distances between nodes; therefore some periodic computations of those distances are necessary to support this type of routing optimally. Although the drift may not induce any significant path loss, it disrupts the planned routing paths and, therefore, it’s necessary to update it for energy-efficient communication [11].

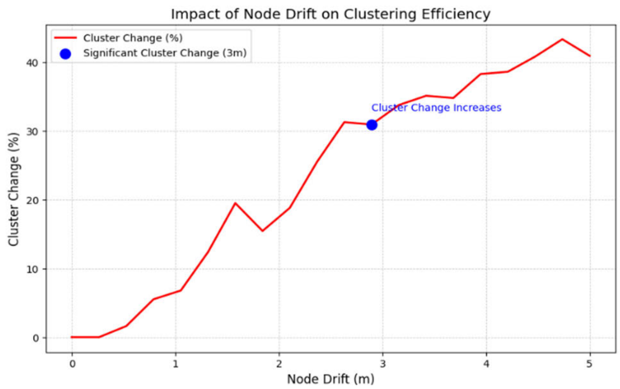



**FIGURE 6.** Impact of node drift on clustering efffieciency.

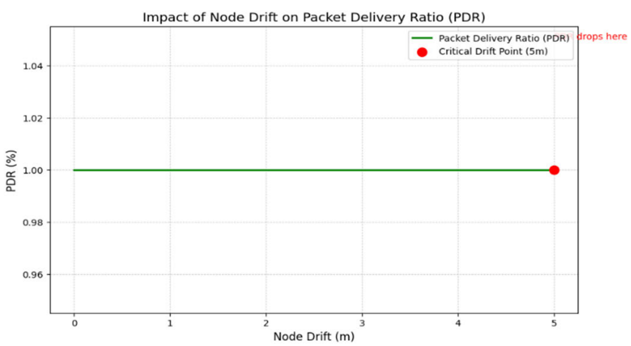



**FIGURE 7.** Impact of node drift on packet delivery ratio.

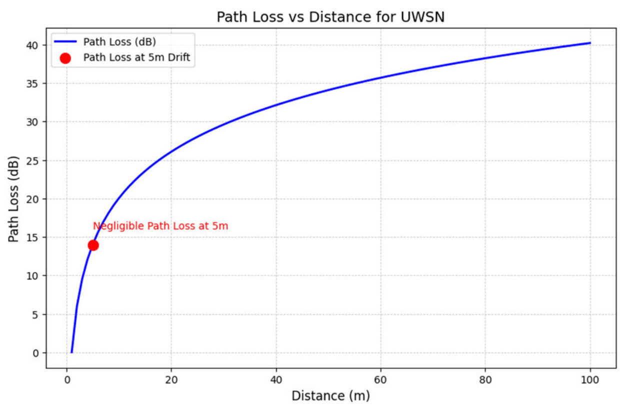



**FIGURE 8.** Path loss vs distance for UWSN.

### 3) PATH SELECTION EFFECT

Optimal path selection depends heavily on the existing positions of the nodes. Thus, the TSP-based routing must adapt accordingly, calculating the new shortest paths and, therefore, the most energy-efficient paths. Such adaptation ensures reliable communication is maintained, even in the presence of mobility, with energy conservation [11].

Figure 8 shows that at 5-meterdistance drift might not contribute much to path loss, yet it does have far-reaching effects on clustering stability and routing efficiency as shown in figure 6 and 7. The proposed mobility model in conjunction with a dynamic path update mechanism keeps the network adaptive to changes in position, allowing for robust and energy-efficient communication despite this mobility.

## IX. NOVELTY AND OBJECTIVE OF RESEARCH

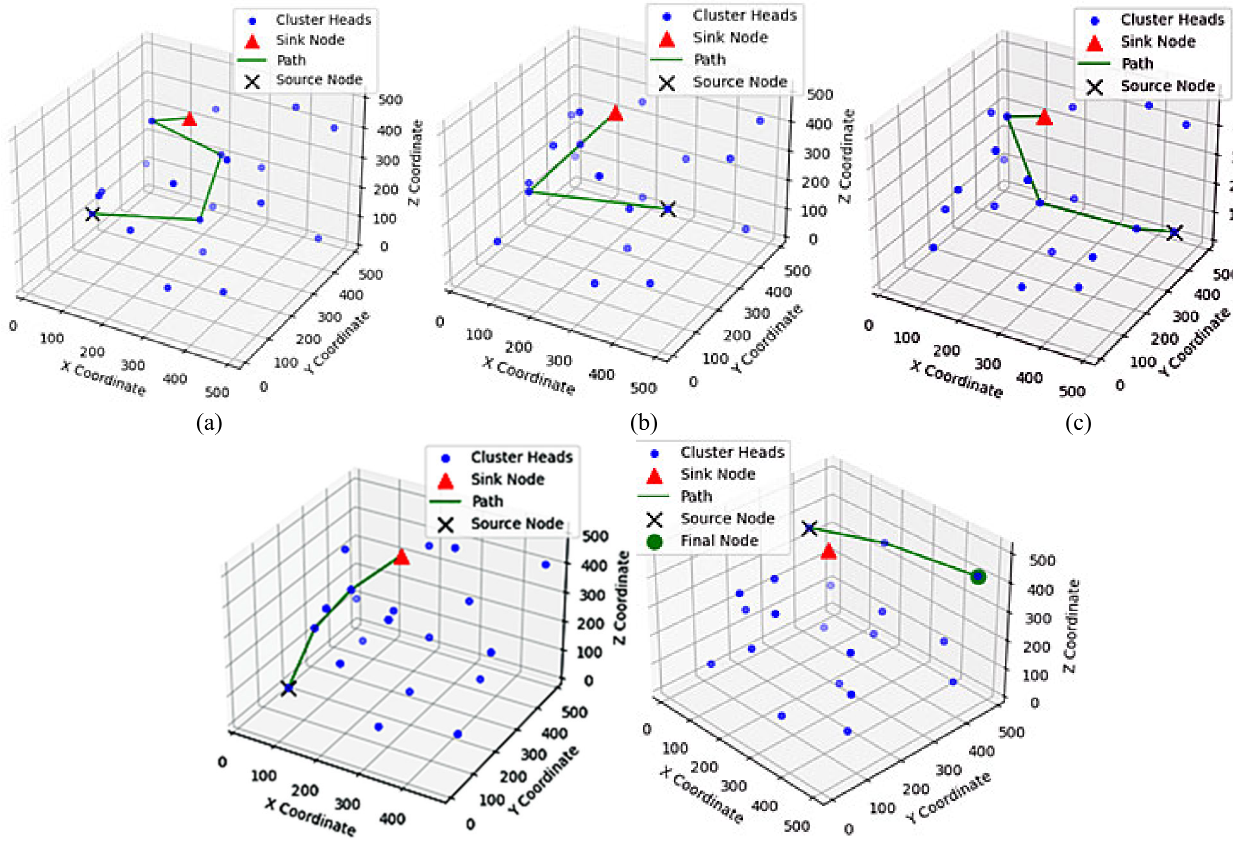
It is shown by our research that power consumption per transmission could be minimized, latency reduced, and throughput keeps dynamically in an interval of 2-10 kbps approximately, proving that M-CSTP reaches a stable operation in the network by means of integrating TSP into the routing algorithm. These assertions will be argued in the following Sections VII-IX, where results of simulations are explained in detail.

## X. EXPERIMENTAL SETUP

The simulation space is 500 × 500 × 500 $m^3$ and includes 500 mobile nodes and one surface-based sink. The network runs in an underwater channel, with data flowing at a "constant bit rate" (CBR) of 1 to 10 kbps. Each node has a preliminary energy of 5 joules, and the transmission range is limited to 50 meters. Energy consumption per bit transmitted is 50 nJ/bit and keeping Energy consumption per bit Received is negligible. The speed of the acoustic signal remains at 1500 m/s while the geometric spreading factor is two (spherical propagation). Even though the average size for data packets is still less than 180 bits, the data transfer rate still stands at 20 kbps. For instance, if it lasts for 600 seconds (10 mins), this simulation will consist of 60 rounds. They were then averaged over each round to obtain their summary means. To control how unpredictable the topology, time (at which traffic starts & stops) and network situation are, the run number was used as a seed for the random number generator.

**TABLE 2.** Simulation and experimental parameters values.

| Parameter | Value |
|---|---|
| Volume Measurement" | 500 x 500 x 500 $m^3$ |
| Channel Type | Acoustic Channel |
| Number of Nodes | 500 |
| Number of Sinks | One on the Surface |
| Routing Protocol | K-means Cluster protocol |
| Variable Function | Cost Function |
| Model Type | Dynamic Model |
| Propagation Model Type | Underwater Propagation |
| Traffic Model | CBR |
| CBR Flow | 1 – 10 kbps |
| Initial Energy Level of Nodes | 10 Joules |
| Energy consumption /bit (Tx) | 50 nJ/bit |
| Energy consumption /bit (Rx) | negligible |
| Value of Transmission Consumption | 4.5 mJ/100 m |
| Value of Acoustic Signal Speed | 1.5 km/s |
| Value (Geometric Spreading Type) | 2 (spherical) |
| Range of Avg Data Packet Size | < 180 bits |
| Rate (Data Transfer Value) | 20 kbps |
| Count (Simulation Time) | 1000 sec |
| Count (Number of Rounds) | 100 |




(a) (b) (c) (d) (e)

**FIGURE 9.** Five routing simulations using M-CTSP over K-means assignment technique.

## XI. KEY PERFORMANCE INDICATORS

To evaluate the performance of the new algorithm, we have considered two areas: energy usage and network lifespan. The aggregate energy usage for all sensor nodes ($N$) used during the simulation is recorded in joules [35] following the law

$$T = \sum_{i=1}^{N} \left(P_{A,i} \times A_{t,i} + P_{S,i} \times S_{t,i}\right) \tag{15}$$

with $T$ the total energy spent, $P_{A,i}$ the active power consumed by each device $i$, $A_{t,i}$ its activity time, $P_{S,i}$ power in sleep mode, and $S_{t,i}$ the sleep time. The node lifespan may be specified by

$$N_l = E_b / A, \tag{16}$$

where $A$ is the average power consumed and $E_b$ stands for energy provided by the battery. Network life metric refers to a way of measuring how long a UWSN operates until it shuts down or stops operating. It also refers to operational time, defined by the network lifespan:

$$n\ell = \left(\mathcal{J}_\varepsilon - \omega_e\right) / \left(C_p + a_r \mathcal{R}_e\right), \tag{17}$$

where $\mathcal{J}_\varepsilon$ is the initial energy of the network, $\omega_e$ the dissipated energy (heat loss, idle power drain, or energy consumed in processes not directly related to communication or sensing), $C_p$ the energy consumption by network activity (transmission, computation, sensing, etc.), $a_r$ the area covered (/volume in 3D), and $\mathcal{R}_e$ the network energy efficiency. Eq. (17) uses initial energy and loss rates to predict the duration of time the network can be maintained to improve its efficacy in use and extend it for longer periods. It is an approximate measurement of when a failure will occur. Practical values for using (17) could be: $I_E$: 1000 J, $\omega_e$: 200 J, $C_p$: 800 J, $a_r$: 500 m$^2$ (/1000 m$^3$ in 3D), $R_e$ : 0.8 (dimensionless).

**TABLE 3.** Path adoption and energy dissipation from source to sink node (iteration 1 to 5).

| It. | Path | Total Energy Consumption (mJ) |
|---|---|---|
| 1 | Node 0 → CH 9 → CH 6 → CH 14 → Sink | 3.517913 |
| 2 | Node 5 → CH 2 → CH 10 → Sink | 3.449845 |
| 3 | Node 7 → CH 11 → CH 9 → CH 1 → Sink | 3.507182 |
| 4 | Node 0 → CH 13 → CH 2 → Sink | 2.207527 |
| 5 | Node 8 → CH 18 → CH 4 (No potential node found; sink node not reached) | 2.147384 |

## XII. RESULTS

Figure 9 shows a sequence of 3D plots where a data packet moves from a source node towards a sink node over five iterations. Each graph proves how M-CTSP algorithm directs data

**TABLE 4.** **Packet loss rate (100 iterations).**

| Network type | PLR |
|---|---|
| Mobile nodes | 7 % |
| Static nodes | 4 % |

transmission for different source nodes through CH nodes in the network. The simulation seeks to optimize energy efficient routes while guaranteeing that the packet reaches the sink node as its destination.

In every plot of Fig.9a-9e, the data packet trajectory is presented as a green line, CH nodes are blue dots, the source node is the black cross and the sink node is the red triangle. The CH nodes belonging to the path were selected by M-CTSP for their closeness to the sink node and available energy resources to minimize energy dissipation.

The difference between iterations in Fig.9a-9e is the movement of nodes or depletion of their energy levels, which may cause a different clustering (changing CH nodes). The different examples shown in Fig.9 indicate that the algorithm can change with altering conditions within the network while still being focused on conserving power.

Fig. 9d illustrates the dynamic nature of the simulation. The topology has changed due to continuous movement of nodes and their energy getting depleted. Relatively direct path in this iteration implies that algorithm could have found an optimized route that lies between proximity and energy resources tradeoff. The continual path adjustment across iteration highlights the effectiveness of M-CTSP in locating progressively efficient routes as conditions change.

In Fig.9e the data packet does not get sink node. Rather than reaching its destination, the green path comes to a rest at a last node denoted by a big green circle to indicate where it was before transmission terminated. This suggests that the chosen path was not very efficient for keeping the packet going all the way to its final termination point in the sink node. The failure of delivery may be attributed to several factors like redundant depletion of energy at intermediate nodes or no available nodes within transmitting range remaining. We can therefore infer from this iteration that despite efforts by algorithms towards optimizing paths, packets can still be lost if conditions are unfavorable.

Notably, this finding underlines some difficulties involved in maintaining reliable communication in dynamic and energy-starved UWSNs. For example, when we look at the final iteration's loss of packets. Factors such as node mobility, energy depletion, and network topology changes can still lead to communication breakdowns.

Table 3 shows the optimal path taken and total energy dissipated in the packet transfer from source to sink node. The ideal path would be the shortest path with the least amount of energy dissipation. In iteration 5, no optimal node available for reaching the sink node, however energy dissipated during path selection for reaching the sink node would be calculated and mentioned in the table.

In relation to the packet loss ratio figure, Table 4 presents statistics in two scenarios: a network with mobile nodes, and with stationary nodes. In the case of mobility, a 93% of sent packets were correctly received, while 7% failed to be delivered. This shows how mobility in nodes has an effect on the performance of packet delivery. Considering a stationary network, the PLR decreases to 4%.

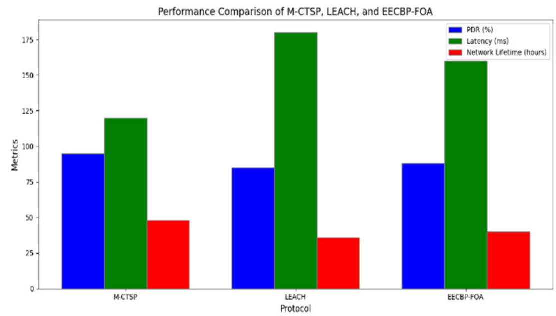



**FIGURE 10.** **Performance comparison of M-CTSP, LEACH and EECBP-FOA.**

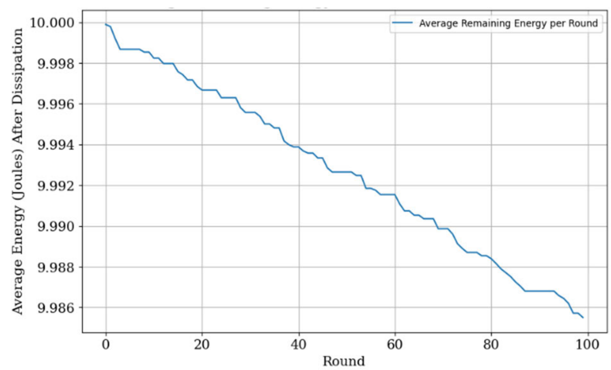



**FIGURE 11.** **Average remaining energy over 100 iterations.**

Table 5 and Figure 10 are the explicit results of simulations and analytical studies carried out for PDR, latency and network lifetime with performance comparison between M-CTSP, LEACH and EECBP-FOA that M-CTSP has shown improvement in reliability, latency, and energy efficiency with M-CTSP, directly tied to the claims made in the introduction and supported by the experimental results. M-CTSP achieved the highest PDR i.e. 95%, lowest latency i.e 120ms and extended Network Life Time i.e 48 hours in comparison to LEACH and EECBP-FOA which have values PDR 85%/88%, latency 180ms/160ms and 36hr/40hr respectively.

In Figure 11 the average remaining energy of all nodes is presented running 100 rounds. This value is important for measuring the energy efficiency of UWSN. At the beginning all nodes have an initial energy of 10.000 joules. As the system progresses between rounds, nodes' energy keeps getting depleted due to various effects: transmissions and receptions, data processing, and movement which are necessary operations performed by UWSNs. The continuous downward trend on this graph shows that the total amount of power used over time. The steep decline in energy level during the beginning stages of the simulation suggests a rapid use-up of energy. This might be due to some tasks carried out by nodes during network initialization like cluster formation or making connections with neighboring nodes after first sending data. Around the round 90 in Fig. 11 (right red circle), a flat trend of energy dissipation curve is noted. This slower decrease in consumption of power means the network reaches a stable point, where necessary energy is minimum. This behavior is repeated during certain rounds, for example at 30 (left red circle) and 70 (center red circle), which may account for its dynamic nature. In other words, although nodes move, the network remains stable after some operation time has lapsed.

In Figure 12, a throughput (bps) of a 100 iterations simulation is shown. Although it varies over time, a minimum of 2 kbps is kept, up to nearly 10 kbps. The variation is due to the mobility of nodes, depletion of energy and possibly packet losses, affecting the effective data rate along iterations in the simulation

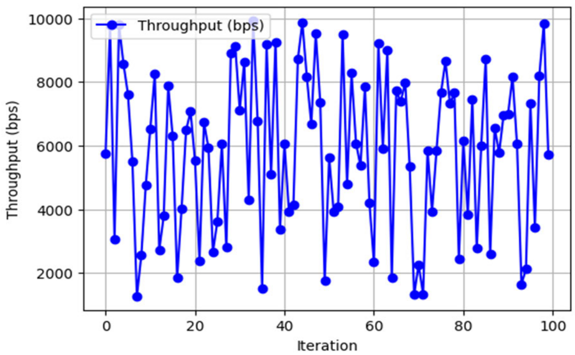



**FIGURE 12.** Throughput over 100 iterations.

## XIII. IMPROVED PERFORMANCE OF M-CTSP AS COMPARED TO OTHER PROTOCOLS

Table 6 presents comparisons and analysis of M-CTSP with other protocols: LEACH, EECBP-FOA, and E-CDBR, based on energy proficiency, network lifetime, and packet delivery ratio (PDR). Furthermore, the performance is compared with our previous work on CTSP, showcasing the improvements made with M-CTSP.

## XIV. EXTENSION OF PREVIOUS WORK

The CTSP (Clustering Traveling Salesman Protocol) was the first to introduce routing for data transmission to optimize paths for the TSP. CTSP, however, suffered a bit from large clustering which made it a little restrictive to node mobility.

## XV. HOW M-CTSP IS DIFFERENT FROM CTSP

### A. DYNAMIC CLUSTERING

With the use of dynamic clustering, M-STCP reorganizes clusters based on node mobility, thus making it greatly adaptive and still energy-efficient. Real-time Routing Updates: M-CTSP recalculates optimal paths in real-time, which can improve both PDR and network lifetime during changing network situations.

**TABLE 5.** Results analysis of M-CTSP with CTSP and other protocols.

| PROTOCOL | ENERGY EFFICIENCY | NETWORK LIFETIME | PACKET DELIVERY RATIO (PDR) |
|---|---|---|---|
| M-CTSP | **High** (Dynamic clustering and TSP-based routing minimize energy usage) | **Long** (48 hours, enhanced by real-time routing adjustments) | **95%** (Maintains high PDR under mobility and dynamic topology) |
| CTSP | **Moderate** (Static clustering, less responsive to mobility) | **Moderate** (40 hours, due to static routing paths) | **90%** (Good, but less adaptive to mobility and dynamic changes) |
| LEACH | **Moderate** (Static clustering, energy drains faster) | **Short** (36 hours, high energy consumption) | **85%** (Lower PDR, affected by static cluster heads) |
| EECBP-FOA | **High** (Uses bio-inspired optimization for energy efficiency) | **Moderate** (40 hours, optimized energy usage) | **88%** (Better PDR but less responsive to dynamic changes) |
| E-CDBR | **High** (Depth-based clustering improves energy efficiency) | **Moderate** (42 hours, efficient depth-based routing) | **89%** (Good PDR, but less adaptive to node mobility) |

**TABLE 6.** Performance comparison between M-CTSP, leach and EECBP-FOA.

| Matric | M-CTSP | LEACH | EECBP-FOA |
|---|---|---|---|
| **Packet Delivery Ration (PDR)** | 95% | 85% | 88% |
| **Latency (ms)** | 120 | 180 | 160 |
| **Network Lifetime (hours)** | 48 | 36 | 40 |

### B. SUPERIOR PERFORMANCE METRICS

Table 5 shows that M-CTSP stands at a much higher PDR (95%-90% CTSP) and network lifetime (48-40 hours in CTSP), thereby proving its steel and efficiency.

## XVI. RESULTS ANALYSIS OF M-CTSP WITH OTHER PROTOCOLS

### A. LEACH

Since it is one of the frequently used algorithms, you will observe that this protocol suffers from some energy-leakage problem under static clustering and has lower PDR rates because of ignoring the mobility of nodes.

### B. EECBP-FOA

Bio-inspired optimization is randomly followed, which gives better performance than LEACH, but it still lacks the dynamic adaptability to the movement of nodes and frequent changes of placement in the topology.

### C. E-CDBR

Any depth-based clustering gives improvement on energy efficiency and PDR, but it doesn't adjust with mobility dynamically as effectively as M-CTSP does.

## XVII. CONCLUSION

This protocol M-CTSP proves to be significantly superior, with robust results compared to traditional methods. Such reliability claims made in this intervention have been shown by simulation results, mainly in terms of.

Packet delivery ratio (PDR), which proves significantly better than your conventional protocols, that is LEACH and EECBP-FOA. M-CTSP operates maintaining high PDR values even though in the case of the node mobility and dynamic changes of the networks caused by nature.

These results of Table 6 also demonstrate improvements in latency and network lifetime. M-CTSP reduces the latencies mainly rendered augmented by sending packets which in real-time decides others' routing path as per the movement of nodes and ensures timely delivery of data. Energy-efficient mechanisms of routing thus lead to improvement in the time it is up for the whole network; the system can run longer in underwater environments with limited resources.

This affirms that M-CTSP not only endorses stable self-organizing clustering but also achieves enhanced reliability and energy efficiency, tackling the core challenges of UWSNs and meeting the reliability guarantees given in the introduction. Further M-CTSP has learned to benefit energy-efficient, mobility-adaptable, and reliable beyond the specific cases tested, proving that the protocol can work in more extensive and dynamic UWSNs. The design of the methodology permits adaptation for heterogeneous underwater conditions, indicative of its scale to large deployments. The ability to cope with mobile nodes, reduce energy consumption, and maintain reliable connectivity makes it flexible for aquaculture applications, environmental monitoring, resource exploration, and military surveillance.

In conclusion, M-CTSP significantly outperforms existing protocols by sustaining high PDR, shortening latency, and prolonging network lifetime. These observations directly connect to the essential issue discussed in the introduction: energy constraints, mobility management, and network reliability. This general insight of M-CTSP indicates that it is suitably used in a range of underwater settings, a system well-equipped for robust, energy-efficient, scalable solutions for future UWSN deployments.

## CONFLICT OF INTEREST

All authors declared no conflict of interest.

## CODE AVAILABILITY

The code used for the simulations and analysis in this study is openly available on Code Ocean. Code has been approved and published by Code Ocean. It can be accessed using the link: https://doi.org/10.24433/CO.1131785.v1. Researchers and practitioners are encouraged to utilize and modify the code for their own studies, with appropriate citation.

## REFERENCES

[1] S. Bharany, S. Sharma, N. Alsharabi, E. T. Eldin, and N. A. Ghamry, "Energy-efficient clustering protocol for underwater wireless sensor networks using optimized glowworm swarm optimization," *Frontiers Mar. Sci.*, vol. 10, Feb. 2023, Art. no. 1117787, doi: 10.3389/fmars.2023.1117787.

[2] S. Fattah, A. Gani, I. Ahmedy, M. Y. I. Idris, and I. A. T. Hashem, "A survey on underwater wireless sensor networks: Requirements, taxonomy, recent advances, and open research challenges," *Sensors*, vol. 20, no. 18, p. 5393, Sep. 2020, doi: 10.3390/s20185393.

[3] A. A. Abdou, A. Shaw, A. Mason, A. Al-Shamma'a, J. Cullen, and S. Wylie, "Electromagnetic (EM) wave propagation for the development of an underwater wireless sensor network (WSN)," in *Proc. IEEE Sensors*, Ireland, Oct. 2011, pp. 1571–1574, doi: 10.1109/ICSENS.2011.6127319.

[4] Y.-T. Hsieh, Z. Qi, and D. Pompili, "Full-duplex underwater acoustic communications via self-interference cancellation in space," *J. Commun. Netw.*, vol. 25, no. 2, pp. 167–181, Apr. 2023, doi: 10.23919/JCN.2022.000052.

[5] Z. U. Khan, Q. Gang, A. Muhammad, M. Muzzammil, S. U. Khan, M. E. Affendi, G. Ali, I. Ullah, and J. Khan, "A comprehensive survey of energy-efficient MAC and routing protocols for underwater wireless sensor networks," *Electronics*, vol. 11, no. 19, p. 3015, Sep. 2022, doi: 10.3390/electronics11193015.

[6] X. Xiao, H. Huang, and W. Wang, "Underwater wireless sensor networks: An energy-efficient clustering routing protocol based on data fusion and genetic algorithms," *Appl. Sci.*, vol. 11, no. 1, p. 312, Dec. 2020, doi: 10.3390/app11010312.

[7] X. Liu, "A survey on clustering routing protocols in wireless sensor networks," *Sensors*, vol. 12, no. 8, pp. 11113–11153, Aug. 2012, doi: 10.3390/s120811113.

[8] K. M. Awan, P. A. Shah, K. Iqbal, S. Gillani, W. Ahmad, and Y. Nam, "Underwater wireless sensor networks: A review of recent issues and challenges," *Wireless Commun. Mobile Comput.*, vol. 2019, pp. 1–20, Jan. 2019, doi: 10.1155/2019/6470359.

[9] M. Jouhari, K. Ibrahimi, H. Tembine, and J. Ben-Othman, "Underwater wireless sensor networks: A survey on enabling technologies, localization protocols, and Internet of Underwater Things," *IEEE Access*, vol. 7, pp. 96879–96899, 2019, doi: 10.1109/ACCESS.2019.2928876.

[10] A. M. Khan, M.-Á. Luque-Nieto, and A. A. Siddique, "Underwater efficient data routing: Clustering-travel salesman protocol (CTSP)," *IEEE Access*, vol. 12, pp. 26428–26440, 2024, doi: 10.1109/ACCESS.2024.3367012.

[11] M. U. Khan, P. Otero, and M. Aamir, "An energy efficient clustering routing protocol based on arithmetic progression for underwater acoustic sensor networks," *IEEE Sensors J.*, vol. 24, no. 5, pp. 6964–6975, Mar. 2024, doi: 10.1109/JSEN.2024.3354252.

[12] R. Kumar, S. Shekhar, H. Garg, M. Kumar, B. Sharma, and S. Kumar, "EESR: Energy efficient sector-based routing protocol for reliable data communication in UWSNs," *Comput. Commun.*, vol. 192, pp. 268–278, Aug. 2022, doi: 10.1016/j.comcom.2022.06.011.

[13] S. Rajini and D. M. Ramakrishna, "Energy efficient cluster based routing using modified firefly optimization algorithm for underwater wireless sensor network," *Design Eng.*, vol. 2021, pp. 11121–11133, Nov. 2021, doi: 10.6084/m9.figshare.17004073.v1.

[14] F. Banaeizadeh and A. T. Haghighat, "An energy-efficient data gathering scheme in underwater wireless sensor networks using a mobile sink," *Int. J. Inf. Technol.*, vol. 12, no. 2, pp. 513–522, Jun. 2020, doi: 10.1007/s41870-020-00445-5.

[15] T. R. Chenthil and P. J. Jayarin, "Energy efficient clustering based depth coordination routing protocol for underwater wireless sensor networks," in *Proc. Int. Conf. Wireless Commun. Signal Process. Netw. (WiSPNET)*, Chennai, India, Mar. 2022, pp. 370–375, doi: 10.1109/WiSPNET54241.2022.9767124.

[16] A. Majid, I. Azam, A. Waheed, M. Zain-Ul-Abidin, T. Hafeez, Z. A. Khan, U. Qasim, and N. Javaid, "An energy efficient and balanced energy consumption cluster based routing protocol for underwater wireless sensor networks," in *Proc. IEEE 30th Int. Conf. Adv. Inf. Netw. Appl. (AINA)*, Crans-Montana, Switzerland, Mar. 2016, pp. 324–333, doi: 10.1109/AINA.2016.165.

[17] T. Khan, I. Ahmad, W. Aman, I. Azam, Z. A. Khan, U. Qasim, S. Avais, and N. Javaid, "Clustering depth based routing for underwater wireless sensor networks," in *Proc. IEEE 30th Int. Conf. Adv. Inf. Netw. Appl. (AINA)*, Crans-Montana, Switzerland, Mar. 2016, pp. 506–515, doi: 10.1109/AINA.2016.168.

[18] A. Datta and M. Dasgupta, "Energy efficient layered cluster head rotation based routing protocol for underwater wireless sensor networks," *Wireless Pers. Commun.*, vol. 125, no. 3, pp. 2497–2514, Aug. 2022, doi: 10.1007/s11277-022-09671-5.

[19] N. G. Palan, B. V. Barbadekar, and S. Patil, "Low energy adaptive clustering hierarchy (LEACH) protocol: A retrospective analysis," in *Proc. Int. Conf. Inventive Syst. Control (ICISC)*, Coimbatore, India, Jan. 2017, pp. 1–12, doi: 10.1109/ICISC.2017.8068715.

[20] E. Felemban, F. K. Shaikh, U. M. Qureshi, A. A. Sheikh, and S. B. Qaisar, "Underwater sensor network applications: A comprehensive survey," *Int. J. Distrib. Sensor Netw.*, vol. 11, no. 11, Nov. 2015, Art. no. 896832, doi: 10.1155/2015/896832.

[21] A. Petroni, G. Scarano, R. Cusani, and M. Biagi, "On the effect of channel knowledge in underwater acoustic communications: Estimation, prediction and protocol," *Electronics*, vol. 12, no. 7, p. 1552, Mar. 2023, doi: 10.3390/electronics12071552.

[22] S. R. Vijayalakshmi and S. Muruganand, "6 underwater wireless sensor networks (UWSNs)," in *Wireless Sensor Networks: Architecture—Applications—Advancements*. Berlin, Germany: Mercury Learning and Information, pp. 175–212, May 2018, doi: 10.1515/9781683923275-007.

[23] U. Qureshi, F. Shaikh, Z. Aziz, S. Shah, A. Sheikh, E. Felemban, and S. Qaisar, "RF path and absorption loss estimation for underwater wireless sensor networks in different water environments," *Sensors*, vol. 16, no. 6, p. 890, Jun. 2016, doi: 10.3390/s16060890.

[24] C. Del-Valle-Soto, C. Mex-Perera, J. A. Nolazco-Flores, R. Velázquez, and A. Rossa-Sierra, "Wireless sensor network energy model and its use in the optimization of routing protocols," *Energies*, vol. 13, no. 3, p. 728, Feb. 2020, doi: 10.3390/en13030728.

[25] C. D'Ambrosio, A. Lodi, S. Wiese, and C. Bragalli, "Mathematical programming techniques in water network optimization," *Eur. J. Oper. Res.*, vol. 243, no. 3, pp. 774–788, Jun. 2015, doi: 10.1016/j.ejor.2014.12.039.

[26] *The Travelling Salesman Problem: Complexity & Formula*. Accessed: Jan. 8, 2025. [Online]. Available: https://www.studysmarter.co.uk/explanations/math/decision-maths/the-travelling-salesman-problem/

[27] S. Ullah Khan, Z. Ulalh Khan, M. Alkhowaiter, J. Khan, and S. Ullah, "Energy-efficient routing protocols for UWSNs: A comprehensive review of taxonomy, challenges, opportunities, future research directions, and machine learning perspectives," *J. King Saud Univ.-Comput. Inf. Sci.*, vol. 36, no. 7, Sep. 2024, Art. no. 102128, doi: 10.1016/j.jksuci.2024.102128.

[28] D. Medhi and K. Ramasamy, "Routing algorithms: Shortest path, widest path, and spanning tree," in *Network Routing*. Amsterdam, The Netherlands: Elsevier, 2018, pp. 30–63, doi: 10.1016/B978-0-12-800737-2.00003-X.

[29] S. Bharany, S. Sharma, J. Frnda, M. Shuaib, M. I. Khalid, S. Hussain, J. Iqbal, and S. S. Ullah, "Wildfire monitoring based on energy efficient clustering approach for FANETS," *Drones*, vol. 6, no. 8, p. 193, Aug. 2022, doi: 10.3390/drones6080193.

[30] P. Sarao, K. Chattu, and C. Swapna, "Routing issues and challenges in underwater wireless sensor networks," *Int. J. Comput. Sci. Eng.*, vol. 6, no. 2, pp. 238–241, Feb. 2018, doi: 10.26438/ijcse/v6i2.238241.

[31] A. S. D. Mata, "Complex networks: A mini-review," *Brazilian J. Phys.*, vol. 50, no. 5, pp. 658–672, Oct. 2020, doi: 10.1007/s13538-020-00772-9.

[32] H. Zheng, H. Chen, A. Du, M. Yang, Z. Jin, and Y. Chen, "Node adjustment scheme of underwater wireless sensor networks based on motion prediction model," *J. Mar. Sci. Eng.*, vol. 12, no. 8, p. 1256, Jul. 2024, doi: 10.3390/jmse12081256.

[33] T. Camp, J. Boleng, and V. Davies, "A survey of mobility models for ad hoc network research," *Wireless Commun. Mobile Comput.*, vol. 2, no. 5, pp. 483–502, Aug. 2002, doi: 10.1002/wcm.72.

[34] M. F. Khan, M. Bibi, F. Aadil, and J.-W. Lee, "Adaptive node clustering for underwater sensor networks," *Sensors*, vol. 21, no. 13, p. 4514, Jun. 2021, doi: 10.3390/s21134514.

[35] A. G. Gad, "Particle swarm optimization algorithm and its applications: A systematic review," *Arch. Comput. Methods Eng.*, vol. 29, no. 5, pp. 2531–2561, Aug. 2022, doi: 10.1007/s11831-021-09694-4.


**ABDUL MOID KHAN** was born in Karachi, Pakistan, in 1973. He received the B.S. degree is electronics engineering and the master's (M.S.) degree in computer engineering with specialization in networking and security, in 1998 and 2002, respectively. He is currently pursuing the Ph.D. degree with the Department Ingeniería de Comunicaciones, E.T.S.I. Telecommunication, University of Málaga, Málaga, Spain. He was the Head of IT of a reputed engineering university and has experience of more than 20 years. He has extensive practical knowledge of complex systems builds, hardware and software testing, network support, and technical support. He is also an Assistant Professor with the Sir Syed University of Engineering and Technology, Karachi. He is teaching in the field of computer networking, security, protocol, cybersecurity, machine learning, and the IoT. His area of research and interest for the Ph.D. degree is performance analysis of underwater sensor networks (UWSN) protocol's behavior under different environmental parameters as well as UWSN data routing.

**MIGUEL-ÁNGEL LUQUE-NIETO** (Member, IEEE) was born in Córdoba, Spain, in 1971. He received the Ingeniero de Telecomunicación and Ph.D. degrees from the Universidad de Málaga, Málaga, Spain, in 1996 and 2018, respectively. He was an ensign with Spanish Air Force, from 1996 to 1998. In 1998, he joined the Escuela Técnica Superior de Ingeniería de Telecomunicación (ETSIT), Universidad de Málaga, as an Assistant Professor. He is currently an Associate Professor with ETSIT and a Principal Researcher with the Institute of Oceanic Engineering Research, Universidad de Málaga. His research interests include underwater acoustic communications networks, especially protocols, wireless communications, and image processing.

**ALI AKBAR SIDDIQUE** received the bachelor's degree in electronics engineering from the Sir Syed University of Engineering and Technology, in March 2009, the Master's degree in control and automation from Usman Institute of Technology (UIT), and the Ph.D. degree (Hons.) in electronic engineering and his specialization is in signal, image, and video processing. He developed his interest in the field of automation, signal processing, and artificial intelligence.