Self-Deployment of Sensors for Maximized Coverage in Underwater Acoustic Sensor Networks

Kemal Akkaya* and Andrew Newell

Department of Computer Science
Southern Illinois University Carbondale
Carbondale, IL 62901
Email: kemal@cs.siu.edu, anewell@siu.edu
**Corresponding Author: Tel: 618-4536054, Fax: 618-4536044*

Abstract

While self-deployment/reconfiguration of terrestrial wireless sensor networks (WSNs) has been studied extensively, such self-organization has just started to receive attention for Underwater Acoustic Sensor Networks (UWSNs). Particularly, self-deployment of sensor nodes in UWSNs is challenging due to certain characteristics of UWSNs such as three dimensional (3-D) environment, restrictions on node movement and longer delays in communication. Given these characteristics, self-deployment of sensor nodes should not only ensure the necessary coverage but also guarantee the connectivity for data transmission as in the case of terrestrial WSNs. In this paper, we propose a distributed node deployment scheme which can increase the initial network coverage in an iterative basis. Assuming that the nodes are initially deployed at the bottom of the water and can only move in vertical direction in 3-D space, the idea is to relocate the nodes at different depths based on a local agreement in order to reduce the sensing overlaps among the neighboring nodes. The nodes continue to adjust their depths until there is no room for improving their coverage. We tune the parameters of the algorithm to also provide connectivity of the network with a surface station. We compared the coverage and connectivity performance of this distributed scheme with distributed/semi-distributed baseline schemes and centralized schemes which can provide optimal coverage/connectivity. We also provide several observations regarding the coverage/connectivity performance and message/travel/time complexity of the proposed approach.

Key words: UWSNs; underwater sensor deployment; self-organization; 3-D coverage; underwater mobility

1. Introduction

UWSNs provide the means for real-time, accurate and energy-efficient monitoring of seas/oceans [1][9][19][11][12][16]. Such networks include a large number of underwater sensors with acoustic modems on-board and limited number of autonomous surface sinks/stations which can collect data reported from those sensors and communicate with on-shore stations through radio communication. While sensors can be deployed both on the water surface and underwater to collect ambient data, autonomous sinks/stations typically stay on the surface to receive data from the sensors. Typical applications of UWSNs include but not limited to underwater tactical surveillance to detect enemy submarines, small delivery vehicles, mines and divers, detection of pollution in coastal areas, performing in-situ oceanic studies of bird/fish migration, detection of terrorist threats to ships in ports, detection of tsunamis and sending warnings, etc.

As opposed to terrestrial WSNs, UWSNs are deployed in 3-D environments which introduces new challenges in terms of connectivity, coverage and mobility. While maximizing the total network coverage is necessary for being able to sense information at every spot of the region, maintaining connectivity is crucial for continuous data gathering from the sensors. Achieving these goals is closely related to proper self-deployment of nodes in 3-D environments. While a lot of research has been done for node deployment and self-organization in terrestrial WSNs [22], there is still much to do for self-deployment of the nodes in UWSNs.

This stems from the observation that current node deployment in UWSNs is mostly manual and centralized. Typically; sensors are placed manually with tethers from the surface stations or anchors from the ground. However, such deployment scenarios may not be feasible in some applications where the deployment region is not accessible due to enemy threats or existence of mines in underwater tactical surveillance applications. In addition, such manual deployment requires additional time and cost particularly when the volume of the monitored region is larger. In such applications, a large number of underwater sensors should be placed manually to the pre-determined locations which usually requires a lot of human intervention. As a consequence, in some applications, it might be inevitable to drop the sensors from a flying vehicle or from a fixed ship to the area of interest. In such a case, the sensors need to self-organize from such random deployment in order to improve the overall coverage and provide connectivity to the surface base-station. Thus, novel node deployment/redeployment mechanisms are needed for UWSNs in order to achieve the desirable application level features with minimized human intervention in a distributed manner. While such self-organization of the nodes is possible in terrestrial WSNs through controlled mobility [20]-[21], such idea has not been employed in UWSNs.

In this paper, a fully distributed and localized technique for self re-configuration of UWSNs that will improve the initial coverage is proposed. The idea is to adjust the depths of the sensor nodes after their initial deployment at the bottom (or surface) of the ocean assuming that sensors can only be moved in vertical direction. Based on the redundancy that exists in 2-D region, the nodes within a neighborhood computes a certain depth which will minimize the coverage overlap among themselves. The redundancy can be determined by one of the nodes which will be referred as the leader within a certain neighborhood by utilizing a vertex coloring problem formulation. The process of depth adjustment continues until there is no room for improving the coverage for a sensor. We also determine the necessary radio/sensing range ratio in order to provide connectivity with the surface base-station. We show how our distributed approach for node deployment behaves under different scenarios and compare it with random, semi-distributed and optimal solutions. Assuming a fixed number of sensors, we illustrate the optimal solutions in terms of coverage based on the volume of the monitored region. The random solution was also implemented in a distributed manner. The simulation results show that with certain configurations, our distributed scheme can perform very close to optimal solution in terms of connected coverage and outperforms the random and semi-distributed approaches. Moreover, random distributed solution proved to be very effective in terms of coverage and connectivity when the sensor density is higher than a certain threshold. Finally, we assess the performance of the proposed scheme under the assumption of the mobility of the nodes and show that its performance is still acceptable.

The paper is organized as follows. In the next section, we summarize the related work. Section III describes the system model and assumptions considered throughout the paper. In Section IV, we describe the proposed approach in details. The performance study along with detailed analysis is done in Section V. Finally, the paper is concluded in Section VI.

2. Related Work

There has been a number of works which focused on node placement in UWSNs for achieving full coverage and connectivity. For instance, the main goal in [18] is to analyze the implications of the sensing and communication ranges on coverage, connectivity and network diameter. Considering a random uniform distribution of nodes, the author derives conditions for the node transmission range r required for achieving a degree of connectivity d, where every node has at least d neighbors. In addition, the average path length between two nodes in the network is formulated as a function of d and r. The same analysis is performed for coverage, basically estimating the sensing range required to achieve a certain degree of coverage in a region using a predetermined number of nodes. For instance, to achieve 1-coverage with a probability tending to 1 as the number of nodes, $n \rightarrow \infty$, the sensing range r of the sensors should be :

$$\sqrt[3]{3\frac{(ln(n) + ln(ln(n)) + \omega(n))\,V}{4\pi n}} \qquad (1)$$

where V is the volume of the region. The work strives to provide theoretical bounds that can help in preliminary design and feasibility studies of UWSNs. However, these bounds were probabilistic and derived for randomly deployed networks. Our goal on the other hand is to study the distributed algorithms that can achieve the desired connectivity and improve the coverage.

Pompili, et al. [17], have used the bounds derived in [18] to validate the effectiveness of their random node deployment scheme for UWSNs. Sensors are to be deployed at the bottom of the ocean along with a few gateway nodes. The sensors send their data to nearby gateways which forward it over vertical communication links to floating buoys at the surface. The idea is to adjust the depths of sensors after their deployment at the bottom of the ocean to provide 1-coverage. The initial deployment is done as random and based on a grid. However, again the deployment is controlled by a central station which tells each sensor where to go after their initial positioning for achieving 1-coverage. Thus, the approach is not fully distributed. While this is very similar to what we propose, as opposed to using a centralized approach, we propose a distributed approach where the nodes decide their depths after local communication. In addition, the paper does not talk about how the depths are adjusted by the surface station, particularly for bottom-grid deployment. Finally, connectivity was not studied in this work and the coverage performance of random approach has not been elaborated although it is very close to the other approaches. The goal of this paper is to observe the coverage and connectivity performance of random and distributed approaches and compare them with the optimal centralized solutions.

Alam and Haas [2] have investigated the problem of achieving maximal 3-D coverage with the least number of sensors. As opposed to [18], random deployment is not the option here. Rather, manual deployment is considered for determining the least number of sensors needed. The authors argue that space filling polyhedrons would be more suitable for 3-D applications. The idea is to fill the 3-D application space with the least number of polyhedrons in order to provide maximal coverage, ideally 100%. The paper compares the truncated octahedron, the rhombic dodecahedron, the hexagonal prism, and the cube in terms of volumetric quotient. The conclusion is that truncated octahedrons created through the use of the Voronoi tessellation of 3-D space yield the best results. In addition to coverage, the paper studies the relationship between the sensing and transmission ranges so that a connected topology is established. For truncated octahedron it has been concluded that connectivity is ensured if the transmission range of the employed nodes is at least 1.7889 times the sensing range. Note that these are the optimal solutions, when the number of sensors and the coordinates of the area are known to a central authority. Although they claim at the end of their paper that the algorithms can be run in a distributed manner by selecting a leader, it requires that each node can reach every node and the nodes can be moved to the desired locations in 3-D space. They state that this is a problem in 3-D and leave it as a future work. Our proposed algorithm in this paper does not find the optimal solution as it assumes only vertical movement in a distributed and localized manner. In addition, we are not trying to find the minimum number of nodes to provide a certain coverage/connectivity. Rather, we strive to maximize coverage with a given number of nodes. However, we will compare the coverage and connectivity of the resultant topologies with these optimal solutions by using the same number of nodes as will be explained in the experiment section.

Closest to our work is presented in [7] where a distributed algorithm for coverage improvement is proposed. As in our setup, the nodes are deployed randomly at the top which are connected to buoys with wires. After this initial deployment, the nodes adjust their depths to improve coverage. However, there are many differences with our work: First, they assume that any two nodes can communicate regardless of their transmission range through the buoys at the top. This is not the model we consider in our work. Any two nodes can communicate only if they are within their radio range in 3-D which makes it fully distributed. In this way, our model allows for a wider variety of UWSNs which may or may not have a buoy system as we will discuss in the experiment section. Second, they assume that the 3-D space is divided into cubes of a priori size and a node should be put in a particular cube. We only assume that a node knows the depth of the water. Third, their algorithm strives to leave as few gaps as possible between nodes which was particularly aimed for increasing the detection chance of enemy submarines and thus is not concerned with the connectivity of the network. Finally, allowing communication between any nodes in their algorithm would require multi-hopping which will increase the message complexity significantly. Our approach only requires single hop information for the nodes.

3. System Model and Assumptions

We assume that a large number of underwater sensors are spread in an area of interest with a surface station. Both the sensors and surface station are assumed to know their locations through ranging or range free mechanisms [8]. Since underwater sensors are expected to be self-reconfigurable, we assume that these sensor nodes have the ability to adjust their positions underwater through mobility. However with the current technology, the mobility of these sensors is limited. Current underwater sensors can only adjust their depths. In other words, they can only move in vertical directions (i.e., z axis). Horizontal movement (i.e., x and y axis) is not possible except with the effects of currents and waves. We note that adjusting the depth of an underwater sensor can be possible in various ways with the current technology. For instance, in [4], a Mica2 mote was put in a cylinder which has a piston for drowning water and forcing water out as shown in Figure 1. As the water moves in, the sensor starts sinking. As the water moves out, it starts rising in the water. In this way, a sensor can control its depth

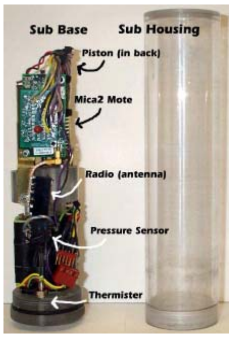


Figure 1: Underwater sensor mote. Taken from [4].

underwater. Similarly, an 'air-bladder-like device' is recommended to be used in underwater sensors for such purpose in [12]. Another possible way to adjust the depth of sensors is to use tethers from anchors which are grounded at the bottom of the ocean [17]. At each sensor there is a buoy which can be inflated by a electronically controlled pump residing on the sensor. The length of the wire can then be adjusted to control the depth of the sensor. A typical architecture for our system model is depicted in Figure 2. In this model, sensors in 3-D can communicate with each other through acoustic channels and find multi-hop paths to the surface station. We will consider both the tethered and untethered architecture throughout the paper. In the former case, the nodes will be anchored to the bottom of the water while in the latter the nodes will be floating freely. For untethered architecture, the mobility of the nodes will be an issue as will be discussed in the Experiment Section. We assume that UWSN can be modeled as a unit ball (sphere) graph of n nodes where any two nodes, u and v have an edge between them if the Euclidean distance $|uv|$ in 3-D is less than the acoustic transmission range r which is assumed to be same for all the nodes in the network.

4. Distributed Node Deployment for Improved 3-D Connected Coverage

In this section, we provide the problem definition, describe our proposed algorithm and discuss its complexity.

4.1. Problem Definition and Approach Overview

The proposed algorithm attempts to maximize the coverage of a UWSN while striving to minimize the number of messages sent/received and total movement distance of the nodes. Connectivity is also considered by deriving the necessary $\frac{r}{s}$ ratio. More formally, the problem can be defined as follows: 'Given N sensors which are randomly and uniformly deployed at the bottom of the ocean/sea as a result of dropping from the water surface, devise an algorithm which will maximize the total 3-D coverage of the nodes in addition to providing connectivity with the surface station with minimized message and travel distance complexity'.

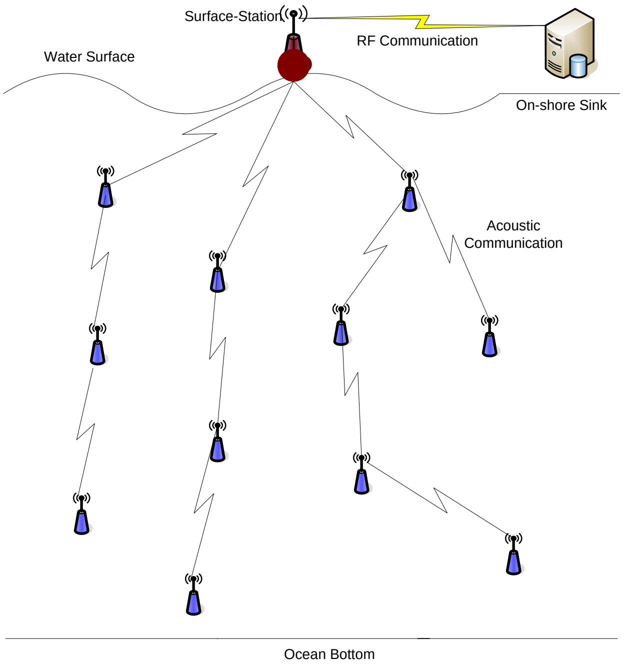


Figure 2: Considered network model in the paper.

As we mentioned before in the Introduction, we are seeking a fully distributed algorithm to be used in specific applications such as tactical surveillance and mine reconnaissance which will enable the sensor nodes to self-organize before the data collection starts. Therefore, we will utilize the relocation ability of the underwater sensor nodes in the vertical direction to adjust the initial network topology. Based on this idea, there can be three different approaches to this problem. Basically, since we assume that the depth of the water is known, one can determine the optimal locations for the nodes based on the solutions provided in [2] and [18] and let each node move to its optimal location underwater. However, there are two problems associated with this approach. First, you need to hard-code the location to each and every sensor in advance. Second, as the sensors can only move in vertical direction, you cannot guarantee that they will end up at the locations desired unless they are deterministically placed manually at the bottom of the water which is not possible for the considered applications. Another solution would be to let the nodes pick random depths once they are deployed at the bottom of the water. While this solution is simple and does not bring any major overhead, it might not provide the desired topology in terms of coverage and connectivity.

Therefore, we opt for a solution which will consider the initial coverage overlaps among the sensors and gradually stretch the topology in vertical direction in order to improve the total coverage while also caring for connectivity. The adjustment of the topology will continue until there is no room for coverage improvement. Nonetheless, this approach will require communication among the nodes in order to determine possible coverage overlaps and thus can be more costly than the random approach. This paper will assess the performance of both approaches under different configurations and determine the applicability scenarios based on certain trade-offs. We now describe the details of our proposed distributed algorithm. We note that the random approach will only be discussed in the experiment section.

4.2. Distributed Topology Adjustment

Determining the coverage overlaps among the sensors and grouping them (i.e., assigning new depths) based on such overlaps is in fact similar to graph coloring problem where each node (i.e., vertex) in a graph is assigned a different color than its 1-hop neighbors. This problem is known to be NP-Hard [10]. In our case, the graph will be formed based on the sensing ranges of the sensor nodes and the groups will correspond to colors. The goal is to assign a unique group ID to a node which will be different than its 1-hop neighbors.

As we consider a distributed approach, the coloring of the whole network can be done using one of the previously proposed distributed algorithms [14] and each node can pick a group ID (i.e., a color) for itself at the end of this process. However, after this process, each node needs to determine its new depth by using the depth of the ocean/water and the total number of colors used to color the whole network which is not available to itself. Note that such number will depend on the maximum node degree in the network and even though it is known, it needs to be forwarded to each node in the network for depth calculation. As this may require even more messaging among the nodes, we pursue a cluster-based approach where coloring is performed within each cluster by the cluster leaders in a centralized manner. Therefore, the cluster leader will know the number of total colors used and will be able to calculate the depths for each individual node within its cluster.

There are four phases in our proposed distributed algorithm: *1) Clustering*; *2) Grouping*; *3) Depth Assignment*; and *4) Additional Rounds*. In the clustering phase, we cluster the nodes in 2-D at the bottom of the ocean based on their node IDs. In grouping phase, the coloring is performed within each cluster. In the depth assignment phase, the leader of each cluster will tell the nodes within the cluster to move to a certain depth with the hope that the coverage overlaps in 2-D will be reduced in 3-D. Finally, in the additional rounds phase, the nodes strive to move further in order to reduce the coverage overlaps that exist even after the initial round of movement. We now describe each phase in details.

4.2.1. Clustering

In this phase, we cluster the nodes which were initially randomly deployed at the bottom of the water based on their node IDs. The basic idea is that each node in a neighborhood picks the highest ID in that neighborhood as its cluster ID (CID). This means that each node exchanges its ID with its 1-hop neighbors and keeps a list of its neighbors' IDs. The node with the highest ID will be the leader of that cluster. Note, however that, a node does not need to be in the cluster in order to be a leader of the cluster, which can be noticed from Figure 3a-b where node 5 is a cluster leader of nodes 2 and 4, but node 5 is not in this cluster. The initial network in Figure 3a is clustered as in Figure 3b.

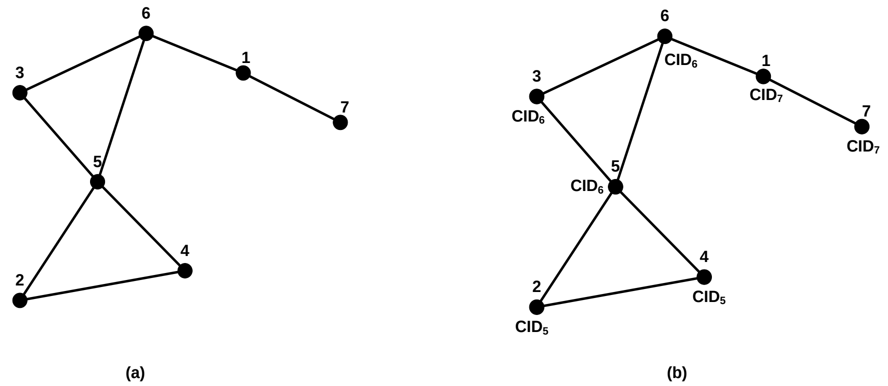


Figure 3: (a) Initial network (b) Clustering of sensor nodes based on node IDs.

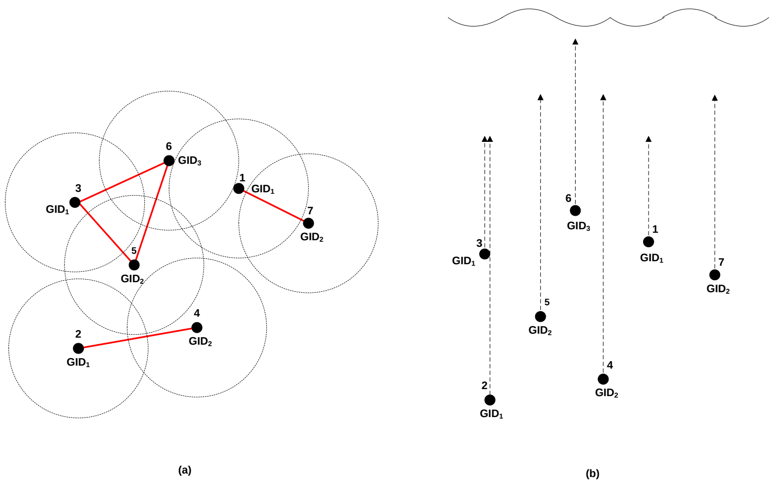


Figure 4: (a) Grouping of the nodes based on sensing coverage overlaps (b) Depth adjustment based on grouping.

4.2.2. Grouping

Once the nodes are clustered, the cluster leaders need to determine the possible coverage overlaps within a cluster. For this purpose, nodes are grouped within each cluster based on their coverage overlaps with their neighbors as seen in Figure 3c. The edge between any two nodes in this graph shows that there is a sensing coverage overlap among the two nodes and they need to be sent to different depths to eliminate this overlap. Thus, at the end of the grouping phase these two nodes need to have different group IDs (GIDs) (i.e., colors). As mentioned above, this is a graph coloring problem and is handled by the cluster leader in a centralized manner. As the cluster leader nows the IDs and locations of the nodes within the cluster, it starts the GID assignment in a sequential manner and each time it picks a new node, it checks the node's edges with the nodes which are already assigned GIDs. In this way, the cluster leader makes sure that no two nodes having an edge shares the same GID. A sample grouping is shown in Figure 4a based on the graph in Figure 3a. Once the cluster leader finishes the process, it sends a message to each node within the cluster which contains the node's GID. The GIDs are than used to determine the new location (i.e., depth) of the nodes as explained next.

4.2.3. Depth Assignment

In order to reduce the coverage overlaps and improve the overall 3-D coverage, the depths that will be assigned to each group will be computed as follows: The space between two different groups will be $\frac{D}{G+1}$ where D is the depth of the water and G is the number of groups. This spacing will make sure that each layer (i.e., xy plane with a certain depth) is $\frac{D}{G+1}$ distance from each other, and the top layer is within this distance from the surface and the bottom layer is within this distance from the ground. Furthermore, this grouping allows nodes without any coverage overlaps to be placed at the same depth layer. The calculation of depth is shown in Figure 5. Again, this computation is done by the cluster leader and the depths are sent to each node within the cluster. Figure 4b shows the configuration of the nodes after the depth assignment is done.

4.2.4. Additional Rounds

After the initial movement, it is possible that the nodes from different clusters can be assigned to the same depth and thus can still have coverage overlaps. Therefore, the nodes continue to attempt to move away from each other (i.e., repel) vertically. However, rather than repeating the first phase (i.e., grouping and depth assignment), each node determines its closest neigh-

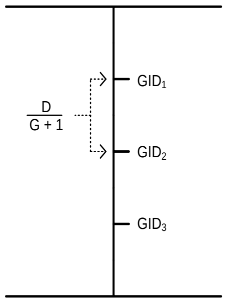


Figure 5: Calculating Initial Depths

bor in terms of distance and checks whether it has a sensing coverage overlap with that node. Further movement will occur if such closest neighbor is within the coverage range of the node. The node will move by an amount dependent on the distance to this neighbor. The closer the two nodes are, the further apart they will attempt to move from one another.

The critical question here is when to stop additional rounds. The movement will need to be stopped when there is no significant improvement for coverage or a certain number of rounds is achieved. We opt to put a cap on the number of rounds as a result of the experiments shown in Figure 6. The parameters used in these experiments match the parameters of our experiment section later on. Basically, after two rounds there is insignificant increase in the total coverage of the network. Since each additional round introduces extra message and movement overhead as will be shown in the experiment section, we set the number of additional rounds to two in the proposed algorithm.

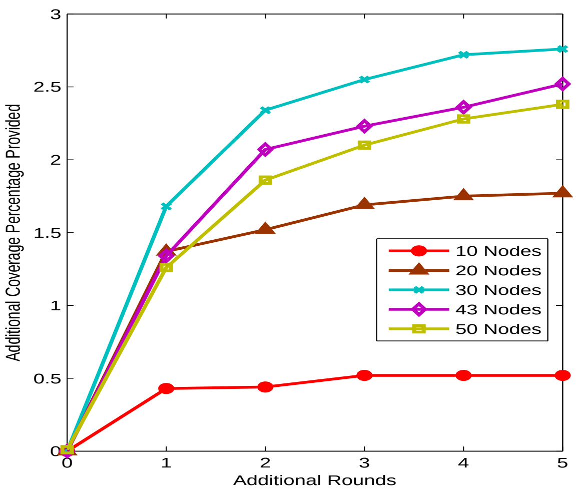


Figure 6: Additional Rounds

We now provide the pseudo-code for the proposed algorithm and discuss the message and time complexity of the algorithm.

4.3. Algorithm Pseudo-code

The proposed algorithm's pseudo-code is given below. This will basically run at each sensor node i.

In line 1, the clustering is performed. Cluster leaders will perform the task of assigning each node within their cluster (shown as N_i in the pseudo-code) a Group ID (GID) which is done in line 3 of Algorithm 1. Once the GIDs are assigned, then a depth for each GID is calculated in lines 4-5. In lines 6-11 of the algorithm, the cluster leaders send DEPTH messages to inform the nodes within their cluster the appropriate depth for their first movement. In addition, the cluster leader performs its first movement (lines 6-7). Any node receiving a DEPTH message, performs its first movement as seen in lines 13-14. After the initial movement, additional rounds are done to increase the total coverage (lines 16-21). For the additional rounds, each node determines the location of its closest neighbor (line 17) and repel from that neighbor if the distance is less than the sensing range s of the sensors. This is repeated MAXROUND times (line 16) which was set to two as explained before.

Algorithm 1: DepthAdjustment(i).

```
1  Initialize Cluster ID (CID)
2  if cluster leader of N_i then
3      assign a Group ID (GID) to each node ∈ N_i
4      forall nodes j ∈ N_i do
5          depth ← calculateDepth(GID_j, MAXDEPTH)
6          if j==i then
7              moveTo(depth)
8          else
9              unicast(j,"DEPTH, depth")
10         end
11     end
12 end
13 if 'DEPTH' message is received then
14     moveTo(depth)
15 end
16 for MAXROUNDS times do
17     j ← closestNeighbor(i)
       // s is sensing range for sensors
18     if Distance(i,j) ≤ s then
19         moveAwayFrom(j)
20     end
21 end
```

4.4. Cost Analysis

In this sub-section, we will analyze the message and time complexity of the proposed algorithm.

4.4.1. Message Complexity

For determining the message complexity of the distributed algorithm, we will consider the number of messages sent and received in the algorithm separately as the the reception's cost is much less than transmission. Assuming that there are n sensor nodes, in the clustering part, each node will send only two messages and will receive up to $2n$ messages (one for creating the neighbor list and the other for creating clusters) in the worst case where n is the total number of nodes. Note that in the worst case, all the nodes will be within the same neighborhood. After the clustering, in the grouping phase for the initial

DEPTH message, the worst case would involve each node in the same cluster and thus the cluster leadr needs to send $(n-1)$ unicast messages and each node will receive one DEPTH message. Each round after this initial movement could require up to one message to be sent and $(n-1)$ messages to be received in the worst case for closest neighbor determination.

Therefore, assuming the total number of rounds to be four, totally, in the worst case a node would be a cluster leader and needs to send $(2 + n - 1 + 2) = (n + 3)$ messages which is $O(n)$. Note that if the node is not a cluster leader, it just needs to send $2 + 2 = 4$ messages which is $O(1)$. Let us assume that the number of cluster leaders is c out of n nodes. Then the total number of sent messages by cluster leaders will be:

$$c(n+3) = cn + 3c \tag{2}$$

For non-cluster leaders the total number of sent messages will be:

$$(n-c)4 = 4n - 4c \tag{3}$$

Total # of sent messages will be (2) +(3) = $cn+4n-c$ which is $O(cn)$.

The # of messages received on the other hand will be $(2n + 1 + 2(n-1)) = (4n-1)$ in the worst case (i.e., cluster leader), which is again $O(n)$. We would like to note that in average the number of received messages will be reduced significantly since the cluster size will be much smaller and thus the message complexity will depend mostly on the number of sent messages which is constant for non-cluster leaders.

4.4.2. Time Complexity

UWSNs are characterized by longer delays compared to terrestrial WSNs. In addition to the lower propagation delays due to speed of sound, the packet drop rates are higher in UWSNs which also increases the MAC level delays. Therefore, the proposed algorithms need to take into account such possible longer delays as well. In this regard, since our distributed algorithm uses no multi-hop, and it requires little messaging overhead for the clustering and initial movement, the time required for the convergence of the algorithm (i.e., end of the deployment) will not be affected significantly. As we will show shortly, the overall time complexity of the algorithm will depend mostly on the depth of the water and the movement speed of the nodes and will be independent of the network size. We introduce the notation in Table 1 before we compute the time complexity of the algorithm:

Table 1: Notation for Computing the Convergence Time

Total Depth	d
Speed of Sound	p
Transmission Delay	t
Sensing Range	s
Transmission Range	r
Vertical Node Speed	v

According to our algorithm, two sets of messaging will occur to establish cluster leaders. Then, cluster leaders send messages to assign nodes an initial depth. The time taken so far would be $3\left(\frac{r}{p}+t\right)$. Then, the nodes will move to their initial depth, where one node might move to the surface in the worst case scenario which would be a distance of d. Thus, each node will need to wait $\frac{d}{v}$ time before moving on to the movement phase.

There are four additional rounds of movements that will occur in our distributed algorithm. For a particular round, each node will broadcast location messages, then determine a new depth based on the information from neighboring nodes. The maximum movement will not exceed the sensing range s. Therefore, $\frac{r}{p}+t+\frac{s}{v}$ time will be required for each round of movement. Now, we can represent the total time of convergence as in Equation 4.

$$3\left(\frac{r}{p}+t\right)+\frac{d}{v}+2\left(\frac{r}{p}+t+\frac{s}{v}\right) \tag{4}$$

The time complexity of the algorithm will then be $O(\frac{r}{p}+t+\frac{d+s}{v})$.

4.4.3. Actuation Cost

When the tethered sensors need to move vertically, they need to inflate the buoy by a pump as much as needed with the power available through their battery. Therefore, the actuation of the pump is one of the main sources for the consumption of the energy of the nodes and is expected to be more than the cost of messaging. However, since this will only be performed once during the lifetime of UWSNs its effect on the total lifetime of a node will not be significant.

Depending on the vertical distance to be traveled, the pump on a node needs to be working for a specific amount of time in order to inflate the buoy. Thus, in this paper, we will assume that the energy required for inflating the pump will depend on the amount of distance to be traveled by a node. In the experiments, we will assess the travel distance for each node as a metric to assess the energy usage of the nodes as a result of actuation. For the time being we note that in the worst case a node will travel a total distance of d. We strive to minimize this travel distance for each particular sensor node.

5. Experimental Evaluation

In this section, we will describe the simulation set up, performance metrics and the performance results.

5.1. Simulation Setup and Performance Metrics

The proposed distributed approach was simulated with varying network topologies and parameters. We assumed that all the messages can be transmitted/received without any error. In addition, we assumed no error in the movement of the nodes given that this can be done through tethers anchored at the bottom of the ocean. We used radio and sensing ranges, r and s respectively and changed the ratio between r and s assuming that $r \geq s$. We created random uniform networks at the bottom of a 3-D space of varying volumes. We created 100 different random topologies and got the average of the results. For a given space, we also determined the least number of sensors to provide 100% coverage and connectivity. This optimal number of sensors are then used to compare with the performance of our approach. We have used the following performance metrics:

- *Coverage:* The percentage of the 3-D volume covered.
- *Connectivity:* The percentage of the number of nodes which can reach (i.e., has a path) to the surface station.
- *Number of Messages:* The total number of message sent and received in the network during the deployment.
- *Convergence Time:* The total amount of time it takes to finish the deployment if the nodes.
- *Total Movement per Node:* The total vertical movement of each node in the network.

We have compared the distributed approach with random approach where each nodes' depth is adjusted randomly and to optimal solutions which are centralized as will be explained in the next subsections. Also, we have compared the distributed approach with an alternative distributed algorithm described in [7]. However as mentioned earlier, this approach is not fully distributed as the nodes can communicate through the assumed surface buoys. Thus, any two nodes only need to be connected in terms of two-dimensions completely ignoring depth. Therefore, the connectivity and number of messages will not be compared to our approach, but the coverage and movement costs can still be comparable even though this alternative approach has an advantage in terms of communication. We will refer it as 'Alternative Distributed Algorithm' in the graphs thereafter.

5.2. Performance Results

Coverage: Improving the coverage of the whole network was one of the main motivations of this study. Thus, we have evaluated the coverage under different conditions and compared the performance of the distributed approach to an alternative distributed approach, random, and optimal solutions.

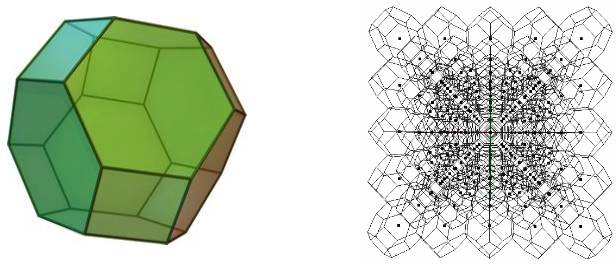
First, we introduce the optimal solutions in terms of coverage. We show how the given volume can be 100% covered with the least number of sensors. This was done in [2]. The idea is to locate the nodes at the center of the truncated octahedrons shown in Figure 7a, and then fill the space by stacking these octahedron shapes as in Figure 7b.

The following formula can be used to place the nodes [2]:

$$\left(cx + (2u_1 + w_1)\frac{2s}{\sqrt{5}}, cy + (2v_1 + w_1)\frac{2s}{\sqrt{5}}, cz + w_1\frac{2s}{\sqrt{5}}\right) \quad (5)$$

where s is the sensing range of the nodes, (u_1, v_1, w_1) is the node location and (cx, cy, cz) is the center of the x, y, z coordinate system. We first determined the number of nodes required for complete coverage in our considered 3-D space of 35.8mx35.8mx53.7m. This was turned out to be 43 nodes when s and r are 10m and 17.9m respectively. The nodes are placed by using the equation 5 as seen in Figure 8.

We would like to mention that when the number of nodes is more than 43, obviously full coverage will be preserved in this optimal solution. However, when the number of nodes is less than 43, the optimal solution with truncated octahedrons will degrade coverage since it allows overlaps among the sensing ranges of the nodes for full coverage and connectivity. Therefore, if 100% coverage is not possible (i.e., the number of sensors is low or the 3-D space is larger) then using truncated octahedrons will not be the optimal solution in terms of coverage. In that case, minimizing the overlap among the sensing ranges will improve the overall coverage. For such a placement, the optimal solution would be to use tetrahedrons to fill the space. Note that equilateral triangles were used in 2-D space for optimal coverage in [17]. Similarly, we can utilize equiliateral triangles and create tetrahedrons(i.e., prisms). The nodes are then located at the vertices of these tetrahedrons which will end up looking like Figure 9 by using the equations shown in 6.



(a) Truncated octahedron. Taken from [22].

(b) Using truncated octahedrons to build a lattice for node deployment. Taken from [2].

Figure 7: 3-D node placement with the least number of sensors for guaranteeing coverage and connectivity.

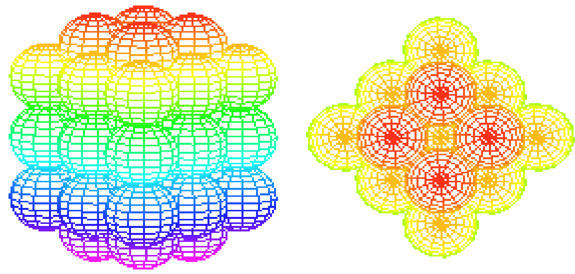


Figure 8: Truncated Octahedrons for Optimal Node Placement in 3-D Space; side and top views.

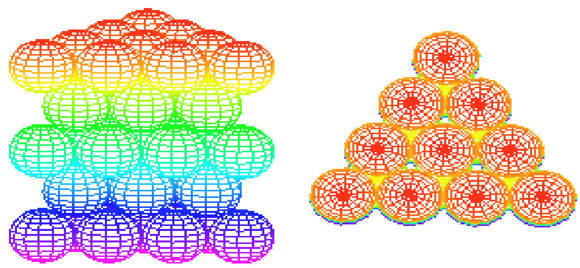


Figure 9: Optimal Strategy for minimal overlap placement using tetrahedrons; side and top views.

$$\begin{cases} x = cx + (2u_1 + v_1 + w_1)s \\ y = cy + (v_1 + w_1 + \frac{2}{3} + \frac{1}{3}(-1)^{w_1})s\sqrt{3} \\ z = cz + w_1 \frac{2s\sqrt{2}}{\sqrt{3}} \end{cases} \quad (6)$$

By using less and more than the optimal number of sensors, we have conducted experiments to assess the coverage performance of the distributed approach, alternative distributed approach, random approach and the two optimal solutions, namely centralized octahedron and centralized prism. In the implementation of the alternative distributed approach we picked $\alpha = 2.24$ and $r = 3$.

The results are depicted in Figure 10. We have observed that the distributed approach can achieve an 88.84% of coverage with 43 nodes when s and r are the same as the optimal solution with centralized octahedron. With less number of nodes (i.e., less than 30), the distributed approach performs very close to centralized tetrahedron-based placement and outperforms centralized octahedron-based placement. In fact, octahedron-based placement performs poorly with the reduced node density in the network. Obviously, the coverage in the distributed approach improves and approaches 100% when increasing the number of nodes.

Comparing to random approach, the distributed approach outperforms it by around 10% even when the network scales. This was not a significant improvement and was due to being able to move the nodes only in vertical direction. However, our algorithm did outperform the alternative distributed algorithm which was bounded by the same movement limitations. This result suggests that with dense networks random placement can also be a reasonable deployment approach. However, we will later show that random placement performs poorer in terms of connectivity.

We have also conducted an experiment by fixing the radio range (to 17.9m) and the number of nodes (to 43) and increasing the sensing range. The coverage performance of our distributed, the alternative distributed, and random approaches are shown in Figure 11. The optimal coverage which was 100% was not shown in this graph. Again, the distributed approach outperforms the random approach by around 10% in terms of coverage. The alternative approach has performed better than the random approach, but still did not perform as good as our distributed approach.

Connectivity: As connectivity of the nodes with the surface station is crucial for data communication, we have defined a metric to assess the connectivity of the network by determining the percentage of the nodes which can reach the surface station. Note that there is always a trade-off between coverage and connectivity. As connectivity is to be ensured, coverage can be reduced depending on the $\frac{r}{s}$ ratio. In the distributed approach, we have striven to place the nodes close to the surface station first so that connectivity can be improved. We have compared the connectivity of the network with random and optimal solutions. To assess the performance, we have conducted two different experiments by varying both the number of nodes and the $\frac{r}{s}$ ratio. The optimal solution is provided in [2] as described in equation 5 above. They proved that if the $\frac{r}{s} = 1.79$

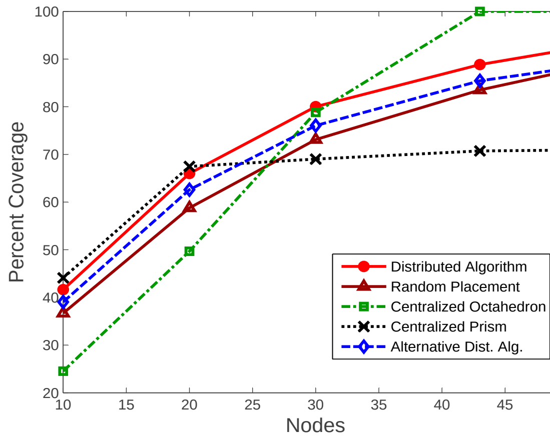


Figure 10: Coverage comparison with varying number of nodes.

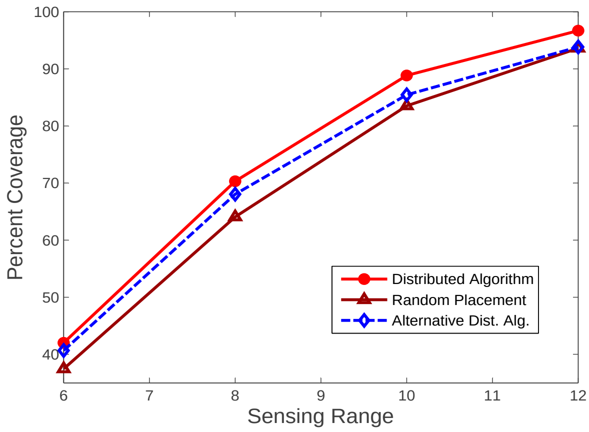


Figure 11: Coverage comparison with varying sensing ranges.

then the connectivity is ensured as long as 100% coverage is achieved. As we have determined the optimal number of nodes to guarantee 100% coverage and connectivity before, we will use the same number of nodes in the distributed and random approaches and determine the connectivity performance of the distributed approach. The results shown in Figure 12 indicate that the distributed approach can achieve much better connectivity than the random approach. In addition, with having $\frac{r}{s}$ at least 1.79, it can guarantee connectivity as in the case of the optimal approach in [2]. Similarly, with the varying number of nodes, connectivity can be guaranteed with the optimal number of nodes used in [2] as seen in Figure 13. This figure also shows that the distributed approach can almost guarantee connectivity with any number of nodes as long as $r \geq 2s$.

Number of Messages: As the approach is distributed, number of messages is an important metric for both energy and delay concerns. In the approach, most of the message exchanges are during neighborhood and coverage overlap detection. The results in Table 2 show that the number of messages per node does not increase as high as the rate of the number of nodes increases. We have tried the same experiment with different transmission ranges as well. As expected, with higher ratio of $\frac{r}{s}$, the number of messages increases since a node can reach

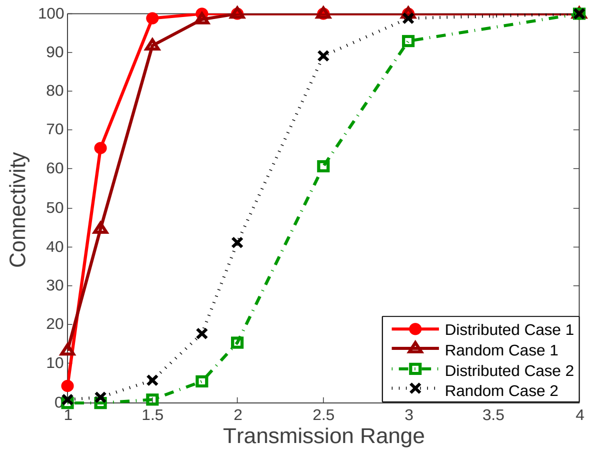


Figure 12: Connectivity with varying transmission range.

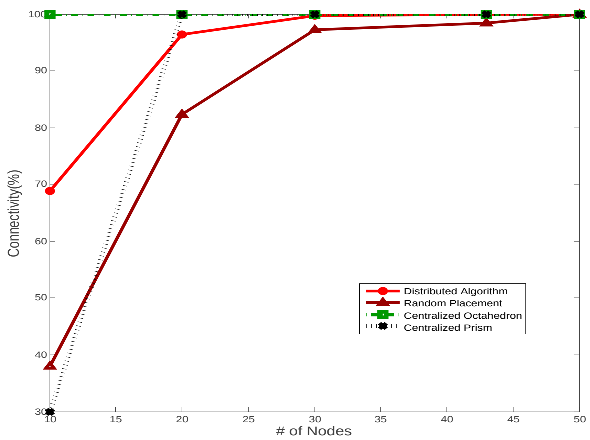


Figure 13: Connectivity with varying number of nodes.

more neighbors. This is shown in Table 3. We note that there is no messaging cost for the random approach. In the case of alternative distributed approach the messaging cost was not assessed as it is assumed that any two nodes can communicate [7]. Since no such mechanism was implemented in the paper, we have not included it in our experiments.

Table 2: Costs with varying number of nodes, r = 17.9m

# of Nodes	# of Messages per Node
10	14.1
20	23.8
30	33.1
43	44.5
50	50.0

Travel Distance: The improvement in terms of coverage and connectivity in the distributed approach is obtained through depth adjustment of the nodes. Obviously, the more the nodes travel, the greater the energy consumption will be in order to perform the depth adjustment through the wires or water displacement. We have conducted experiments with different

Table 3: Costs with varying transmission range. # of nodes = 43

r	# of Messages per Node
12.5	25.0
13.5	28.3
15	33.4
17.9	45.6

Table 4: Costs with varying sensing range

s	# of Messages per Node	Travel Distance per Node (m)
6	46.6	31.2
8	45.4	32.4
10	45.6	34.2
12	44.2	35.6

number of nodes to assess the total travel distance of the nodes and have compared it with the random approach. The results shown in Figure 14 indicate that distributed approach increases the travel distance by around 20% of what the random approach would require. The travel distance in random approach would equal approximately half the depth of the water multiplied by the number of sensors, since the random approach has an equal distribution above and below this midpoint. This is expected as the distributed approach has more rounds and strives to maximize the coverage. In addition, since the distributed approach moves the nodes close to the surface station for connectivity purposes, longer distances need to be traveled. Nonetheless, if the nodes were to start at the surface and move down to adjust their depths, this will further reduce the travel distance of the distributed algorithm. Note that the alternative distributed approach requires even more movement than our algorithm.

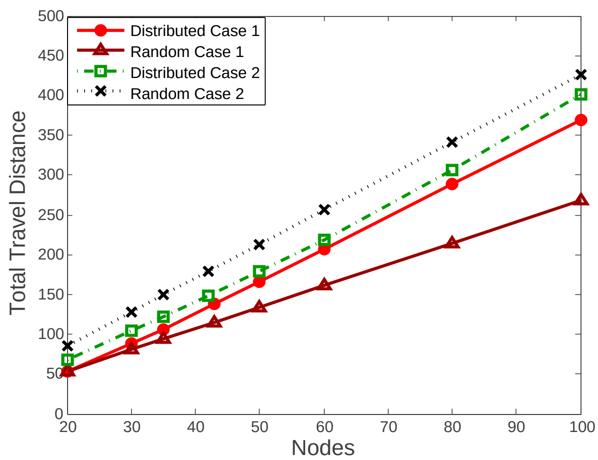


Figure 14: Travel Distance per Node with varying network size.

Similarly, when s is increased while r is fixed, total travel distance increases as seen in Table 4. With smaller s the coverage overlaps are reduced and thus the remaining rounds do not add much to the total travel distance as there will not be a significant improvement. Therefore, there will be a decrease in the total travel distance as s decreases.

5.3. Experiments with Real-life Acoustic Modems

In order to reduce the energy consumption, interference and packet loss ratio, future UWSNs promote use of smaller transmission ranges for their acoustic modems [1]. However, today's acoustic modems such as [15] have the capability of transmitting across much greater distances than we used in our experiments. Therefore in this section, we scaled the transmission range to a much larger distance of 1300 to 1700m [15] for assessing its effect on the performance of our proposed approach. This type of increase in transmission range would be used in a much larger body of water and thus the volume will be increased to 5000x5000x5000m. The number of nodes are held constant at 50, and the sensing range remained 10m as in the previous section. We run experiments with this new configuration and compared our approach with random placement.

The simulations showed that the covered volume is quite small compared to the volume of the region to be monitored. Total volume is $5000^3 = 1.25 * 10^{11}$ and the total possible volume covered by all 50 sensors is $\frac{4}{3}\pi 10^3 * 50 \approx 2.09 * 10^5$. Therefore, there was little to no coverage overlaps to deal with after the initial placement of the sensors for our algorithm. Thus, it performed on the same level as random placement in terms of coverage.

However, our algorithm outperformed the random placement in terms of connectivity. Figure 15 indicates that our algorithm is outperforming random just as it did in the previous section with respect to connectivity of the nodes.

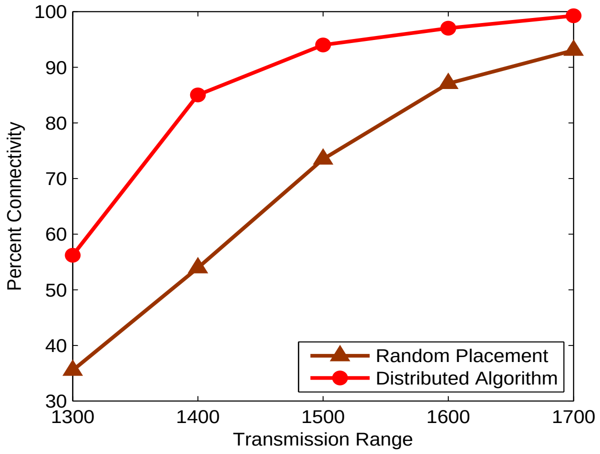


Figure 15: Connectivity with varying number of nodes in real-life scenario.

Overall these experiments showed that our algorithm is able to work under extremely large transmission ranges. Even though there is no coverage overlaps among sensors, our algorithm was able to increase connectivity over the random approach which is also very important. Nonetheless, we re-iterate the fact that in practice these long-reaching acoustic modems are quite lossy and energy inefficient and thus it may be more efficient to lower the transmission range closer to the sensing range as we did in the previous subsection.

5.4. Performance Assessment in Mobile UWSNs

The experiments up to now have been conducted under the assumption that the deployed sensors are tethered and anchored to the ground and thus there is no horizontal movement for the sensors. However, as mentioned in the Section 2, there are other UWSN architectures where the sensors may be floating freely in the water without any tethers. In such a case, there will be an inherent mobility of the nodes due to water currents and waves which can result in random horizontal movement. Even though the nodes will move similar to each other based on the conditions underwater, there will be a slight variation from sensor to sensor. Such mobility of the nodes may affect total coverage and connectivity of the network when the distributed approach is pursued. In this subsection, we will model the mobility of the underwater sensors using one of the existing UWSN mobility models and test the performance of our distributed algorithm.

Recently, modeling the mobility of the UWSNs have started to receive attention as there is a need to be used in simulations for assessing the performance of different protocols proposed for UWSNs. Since the movement of the nodes underwater has its own characteristics, widely used Random Waypoint Model [5] cannot be applied to UWSNs. Therefore, some researchers suggested to use Random Walk [13] to reflect the characteristics of movement underwater. Very recently, more realistic mobility models have been proposed for UWSNs such as [23] and [6]. We will pick the model described in [23] as the other in [6] considers a different initial deployment than the one we used in this paper.

The node mobility model used in [23] is based on the kinematic model described in [3]. According to this model, a node's speed in x and y directions will be calculated based on the following formula:

$$\begin{cases} V_x = k_1 \lambda v sin(k_2 x) cos(k_3 y) + k_1 \lambda cos(2k_1 t) + k_4 \\ V_y = \lambda v cos(k_2 x) sin(k_3 y) + k_5 \end{cases} \tag{7}$$

where k_1, k_2, k_3, λ and v are variables which are closely related to environment. A simulation using mobility was conducted in [23], and each of their sensors were given a 0.1 standard deviation of velocity difference among the nodes and the rest of the random variables pertained to the environment. Thus, in our tests we have assigned each node a random direction and a random velocity with a mean of 0 and a standard deviation of $0.1\frac{m}{s}$.

The results in Figures 16 & 17 show that our distributed algorithm does indeed perform a littler worse in terms of connectivity and coverage in this random mobility scenario compared to tethered architecture. This can be attributed to the fact that the movement of the nodes can involve some errors due to mobility and the desired topology cannot be achieved. Nonetheless, our distributed algorithm still outperforms random distribution approach which shows that it is still a valid choice when working with free floating sensors that may have mobility.

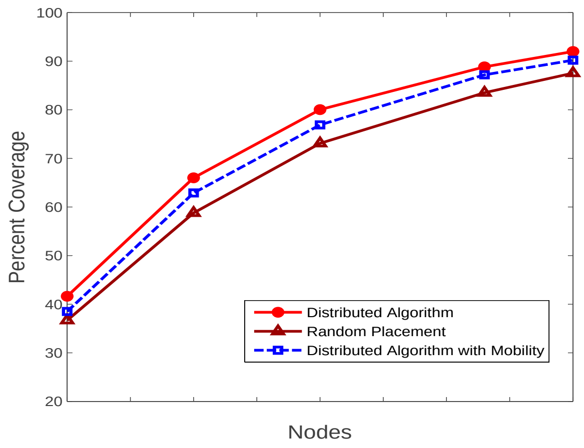


Figure 16: Total Coverage with varying transmission range.

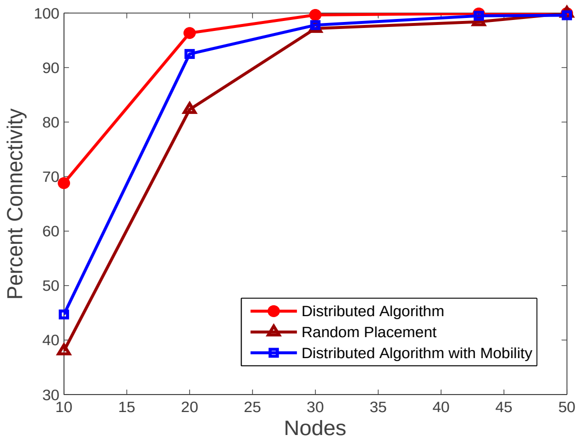


Figure 17: Connectivity with varying transmission range.

6. Conclusion

With the improvements in UWSN technology, distributed deployment mechanisms will be needed for allowing flexibility in many UWSN applications rather than depending on centralized solutions. To this end, in this paper we have presented a distributed node deployment technique for UWSNs in order to improve coverage and provide connectivity with the surface stations. The idea was to adjust the depth of the sensors nodes once they are deployed randomly at the bottom of the ocean/sea given that they can only move in vertical direction. The depth adjustment was done based on the coverage overlaps among the nodes under the water in 2-D. In order to reduce the coverage overlaps in 3-D, a solution based on the idea of graph coloring was adapted. The new locations in vertical direction were computed by cluster leaders based on the group IDs of the nodes assigned as a result of coloring algorithm. Once the nodes move, they further assess any possible coverage overlap with their neighboring nodes and continue to adjust their depth by applying repelling forces in vertical direction. The movement stops either when there is no sensing overlaps for a node or a maximum number of rounds has been achieved.

Through simulation we have shown that the distributed algorithm can improve the coverage better than the random approach while also providing a good connectivity with the surface station. We have shown how 100% coverage and connectivity can be achieved by adjusting the parameters such as the ratio among the radio and sensing ranges and the number of nodes. Further, the experiments related to scaling the network has revealed that our algorithm still outperforms random with the expense of very little extra cost to each node in terms of messaging and travel distance. Our approach has also outperformed another previous approach in terms of coverage which does not consider connectivity and is not fully distributed.

We have also compared the performance of the distributed approach in terms of coverage and connectivity with the optimal centralized solutions from the literature. The goal was to show how close can the distributed approach perform to the optimal solution under a variety of conditions. The performance results have also given us an idea of how the assets can be better utilized depending on the parameters. For instance, while the distributed approach outperforms random deployment, it comes with a little expense in terms of messages and movement complexity. Therefore one can choose to use random approach if there is enough number of sensors with respect to the volume of the space considered.

Finally, we have assessed the performance of our approach under mobility. We have observed that with respect to the used mobility model specifically designed for UWSNs, the coverage and connectivity of our distributed approach has slightly degraded while still outperforming the random approach.

As a future work, we plan to assess the effect of node movement errors in tethered architectures on the performance metrics considered in this paper. In addition, we would like to apply the same ideas to alternative UWSN architectures where clustering around underwater gateways are considered.

References

[1] I. F. Akyildiz, D. Pompili, and T. Melodia, "Underwater acoustic sensor networks: Research challenges." in *Ad Hoc Networks (Elsevier)*, vol. 3, no. 3, May 2005, pp. 257–279.

[2] S. N. Alam and Z. Haas, "Coverage and connectivity in three-dimensional networks." in *12th Annual ACM International Conference on Mobile Computing and Networking (MobiCom'06)*, Los Angeles, CA, September 2006.

[3] S. P. Beerens, H. Ridderinkhof, and J. T. F. Zimmerman, "An analytical study of chaotic stirring in tidal areas," *Chaos, Solitons & Fractals*, vol. 4, no. 6, pp. 1011–1029, Jun 1994.

[4] V. Bokser, C. Oberg, G. S. Sukhatme, and A. A. Requicha, "A small submarine robot for experiments in underwater sensor networks." in *International Federation of Automatic Control Symposium on Intelligent Autonomous Vehicles*, 2004.

[5] T. Camp, J. Boleng, and V. Davies, "A survey of mobility models for ad hoc network research," in *Wireless Communication and Mobile Computing (WCMC), Special issue on Mobile Ad Hoc Networking*, vol. 2, no. 5, 2002, pp. 483–502.

[6] A. Caruso, F. Paparella, L. F. M. Vieira, M. Erol, and M. Gerla, "The meandering current mobility model and its impact on underwater mobile sensor networks." in *Proceedings of IEEE INFOCOM'08*, Phoenix, AZ, April 2008.

[7] E. Cayirci, H. Tezcan, Y. Dogan, and V. Coskun, "Wireless sensor networks for underwater surveillance systems." in *Ad Hoc Networks (Elsevier)*, vol. 4, no. 4, 2006, pp. 431–446.

[8] V. Chandrasekhar, W. Khoon, G. Seah, Y. S. Choo, and H. V. Ee, "Localization in underwater sensor networks: survey and challenges." in *ACM International Workshop on UnderWater Networks (WUWNet'06)*, Los Angeles, CA, Sept. 2006.

[9] M. C. Domingo and R. Prior, "Energy analysis of routing protocols for underwater wireless sensor networks." in *Computer Communications (Elsevier)*, vol. 31, no. 6, April 2008, pp. 1227–1238.

[10] M. R. Garey and D. S. Johnson, *Computers and intractability : a guide to the theory of NP-completeness*. San Francisco: W. H. Freeman, January 1979.

[11] J. Heidemann, W. Ye, J. Wills, A. Syed, and Y. Li, "Research challenges and applications for underwater sensor networking." in *IEEE Wireless Communications and Networking Conference (WCNC'06)*, Las Vegas, Nevada, April 2006.

[12] M. G. J.-H. Cui, J. Kong and S. Zhou, "Challenges: Building scalable mobile underwater wireless sensor networks for aquatic applications." in *Special Issue of IEEE Network on Wireless Sensor Networking*, May 2006.

[13] U. Lee, J. Kong, M. Gerla, J.-S. Park, and E. Magistretti, "Time-critical underwater sensor diffusion with no proactive exchanges and negligible reactive floods," *Ad Hoc Networks*, vol. 5, no. 6, pp. 943–958, August 2007. [Online]. Available: http://dx.doi.org/10.1016/j.adhoc.2007.02.012

[14] N. Linial, "Locality in distributed graph algorithms." in *SIAM Journal on Computing*, vol. 21, no. 1, 1992, pp. 193–201.

[15] LinkQuest, "http://www.link-quest.com/," 2008.

[16] J. Partan, J. Kurose, and B. N. Levine, "A survey of practical issues in underwater networks." in *ACM International Workshop on UnderWater Networks (WUWNet'06)*, Los Angeles, CA, Sept. 2006.

[17] D. Pompili, T. Melodia, and I. F. Akyildiz, "Deployment analysis in underwater acoustic wireless sensor networks." in *ACM International Workshop on UnderWater Networks (WUWNet)*, Los Angeles, CA, September 2006.

[18] V. Ravelomanana, "Extremal properties of three-dimensional sensor networks with applications." in *IEEE Transactions on Mobile Computing*, vol. 3, no. 3, July/Spetember 2004, pp. 246–257.

[19] E. Sozer, M. Stojanovic, and J. Proakis, "Underwater acoustic networks." in *IEEE Journal of Oceanic Engineering*, vol. 25, no. 1, Jan. 2000, pp. 72–83.

[20] G. Wang, G. Cao, and T. L. Porta, "Movement-assisted sensor deployment." in *23rd International Annual Joint Conference of the IEEE Computer and Communications Societies (INFOCOM'04)*, Hong Kong, March 2004.

[21] J. Wu and S. Yang, "Smart: A scan-based movement assisted sensor deployment method in wireless sensor networks." in *24th International Annual Joint Conference of the IEEE Computer and Communications Societies (INFOCOM05)*, Miami, FL, March 2005.

[22] M. Younis and K. Akkaya, "Strategies and techniques for node placement in wireless sensor networks: A survey." in *Elsevier Ad Hoc Network Journal*, vol. 6, no. 4, 2008, pp. 621–655.

[23] Z. Zhou, J.-H. Cui, and A. Bagtzoglou, "Scalable localization with mobility prediction for underwater sensor networks." in *Proceedings of IEEE INFOCOM'08, Mini-Conference*, Phoenix, AZ, April 2008.