Autonomous Deployment of Sensors for Maximized Coverage and Guaranteed Connectivity in Underwater Acoustic Sensor Networks

Research · August 2015

DOI: 10.13140/RG.2.1.3313.6489

1 author:

Fatih Senel

Autonomous Deployment of Sensors for Maximized Coverage and Guaranteed Connectivity in Underwater Acoustic Sensor Networks

Fatih Senel
Department of Computer Engineering
Antalya International University
07190, Antalya, Turkey
Email: fatih.senel@antalya.edu.tr

Kemal Akkaya
Department of Computer Science
Southern Illinois University
Carbondale, IL 62901 USA
Email: kemal@cs.siu.edu

Turgay Yilmaz
Department of Computer Engineering
Middle East Technical University
06531, Ankara, Turkey
Email: turgay@ceng.metu.edu.tr

***Abstract*—Self-deployment of sensors with maximized coverage in Underwater Acoustic Sensor Networks (UWASNs) is challenging due to difficulty of access to 3-D underwater environments. The problem is further compounded if the connectivity of the final network is required. One possible approach is to drop the sensors on the surface and then move them to certain depths in the water to maximize the 3-D coverage while maintaining the connectivity. In this paper, we propose a purely distributed node deployment scheme for UWASNs which only requires random dropping of sensors on the water surface. The goal is to expand the initial network to 3-D with maximized coverage and guaranteed connectivity with a surface station. The idea is based on determining the connected dominating set of the initial network and then adjust the depths of all dominatee and dominator neighbors of a particular dominator node for minimizing the coverage overlaps among them while still keeping the connectivity with the dominator. The process starts with a leader node and spans all the dominators in the network for repositioning. Simulations results indicate that connectivity can be guaranteed regardless of the transmission and sensing range ratio with a coverage very close to a coverage-aware deployment approach.**

I. Introduction

Underwater Acoustic Sensor Networks(UWASNs) can consist of a large number of sensors on and underwater which can communicate via acoustic links [1][2]. Similar to terrestrial WSNs, these networks provide numerous advantages in terms of coverage quality, labor, cost and deployment as opposed to traditional underwater sensor networks. Within the last decade a lot of studies focused on the issues related to communication underwater given that RF signals would not work underwater. The design of acoustic modem, modeling of the channel, medium access, routing and sensing issues had been the main focus of researchers [3][4].

Node deployment and mobility modeling had also been studied in UWASNs given that underwater is not readily accessible and sensor can float due to the characteristics of the environment. The main goals in these studies were to reduce the cost and time of deployment with desired objectives in terms of coverage, accuracy and communication quality. The deployment is a crucial issue especially for the applications which require rapid and remote deployment. For instance, deployment of a UWASN for measuring the water quality of a lake or river which is believed to be contaminated requires rapid deployment to measure the water quality and inform the public. Another example would be the deployment of a UWASN in a remote ocean location for environmental or habitat monitoring which is not easily accessible.

For such cases, there needs to be a mechanism to remotely deploy sensors in reasonable times. Since manual deployment is out of question due to accessibility and time constraints, one possibility would be to randomly drop sensors on the water surface from a helicopter and then expand them to 3-D with depth adjustment. This requires the ability of the sensors to adjust their depths in the water via a mechanism which can change the weight of the sensor using air or water [5]. The vertical movement of acoustic sensors resembles self-spreading of sensors in 2-D environments if they have the ability of move. While there has been studies leveraging this capability of acoustic sensors to expand their coverage [6][7], none of them can guarantee connectivity of the stretched network with a surface station so that the collected data can be send to an on-shore station. To the best of our knowledge, self depth adjustment of sensors subject to connectivity constraint is a novel problem that has not been studied in the past.

In this paper, we propose a novel remote deployment of acoustic sensors which can maximize the coverage in 3-D while guaranteeing the connectivity among the sensors and a surface sink node to collect data. The approach assumes a randomly deployed UWASN on the surface of the water in 2-D. This can be achieved with random dropping of a certain number of sensors in a targeted area [8]. The approach then determines the connected dominating set (CDS) of the whole UWASN at 2-D plane using a distributed approach [9]. The main goal here is to determine a backbone of the network consisting of dominators and then maintain the very same backbone connected underwater by exploring the third axis, namely z. In other words, any two connected dominator can still maintain connectivity while being stretched along the z axis as much as possible until their sensing ranges do not overlap (or have minimal overlapping). This idea is then applied to every dominator along with its dominator and dominatee neighbors one by one.

To pick the sensor node that will initiate the process, we determine a leader dominator by picking the lowest ID node or using a distributed leader election algorithm such as the one in [10]. Each dominator computes the depths of its dominator and dominatee neighbours based on the location of these nodes. Specifically, each dominatee node is sent to a different depth which will minimize the sensing overlaps among these dominatee nodes as well as with the dominator node. This is similar to running a graph coloring algorithm within this network of sensor nodes consisting of edges that are formed with overlapping coverage ranges. Same applies with the dominator neighbors of the node. Once a dominator is done with the computation, it transmits the new depths of these neighboring nodes to them and then passes the token to another dominator neighbor so that it will do the depth calculation in the same manner. The process continues until all the dominators are spanned and every body receives its new depth in a distributed manner. Once every sensor knows its new depth, the depth adjustment is done at the same time and a new network in 3-D is formed. The proposed approach is validated through extensive simulations and has been shown to guarantee the connectivity while achieving a 3-D coverage that is very close a baseline approach targeted for maximized coverage.

This paper is organized as follows. In the next section, we summarize the related work. Section III explains the preliminaries. In Section IV, we present the details of the proposed approach. Section V is dedicated to the evaluation of the approach. Finally, Section VI concludes the paper.

II. Related Work

Achieving full coverage and connectivity in UWASNs has been well studied in the past assuming both random and manual deployment [11][12]. For instance, the main goal in [11] is to analyze the implications of the sensing and communication ranges on coverage, connectivity and network diameter in an UWASN. The author derived conditions for the node transmission range required for achieving a degree of connectivity and coverage. In [12], the deployment is done manually to provide full coverage and connectivity. The idea is to fill the 3-D application space with the least number of polyhedrons in order to provide 100% coverage. Each sensor is deployed within these polyhedrons. The paper concludes that truncated octahedrons yield the best results for coverage. Using the same polyhedron, connectivity is guaranteed if the transmission range of the nodes is at least 1.7889 times the sensing range.

In addition to these works, coverage improvement has also been heavily studied in both distributed and centralized manners [7][13][6]. Among these works, [7] proposes a centralized approach which can tell every sensor where to go to achieve 1-coverage after initially deploying sensors on a grid or randomly. [13] and [6] propose distributed approaches to address these concerns mainly based on depth adjustment. While the communication model in [13] is via a surface buoy through multi-hop communication, this is not the case in [6] where a node can only talk to its 1-hop neighbors. In addition, [6] assumes initial deployment at the bottom of the water in a random manner and only the depth of the water is known whereas [13] assumes that the 3-D space is divided into cubes of a priori size and a node should be put in a particular cube. This makes the approach more semi-distributed since the depth computation is based on a pre-computed cubes. In addition, the approach does not concern with the connectivity since this is assumed to be achieved with the final configuration by adjusting the transmission and sensing ranges. Therefore, we describe the approach in [6] in more details to distinguish it from our proposed work.

In [6], the nodes adjust their levels based on their coverage overlaps with the neighbors. To determine the sensing coverage overlaps, nodes are clustered and links are defined between any two nodes if there is any overlap. Graph coloring is then applied within each group and for each color a new depth is calculated. After this stage, a second phase is completed to eliminate any possible overlaps among the nodes in different groups. This work improves the coverage significantly but does not guarantee any connectivity. It is only claimed that with a certain transmission range and sensing range ratio, the connectivity can also be ensured. While this has been shown to be the case with a certain number of nodes and ratio, the approach certainly cannot guarantee the connectivity for all cases. Our work in this paper fills this gap and guarantees the connectivity of the UWASN.

III. Preliminaries

A. Assumptions

Since part of our goal is to have an autonomous deployment, we assume that a large number of sensors dropped from a helicopter or boat at the surface or bottom of water randomly. Each sensor will have an acoustic modem and the ability to adjust its depth via various mechanisms. In case of a sea, it may be more efficient to drop the sensors at the surface since the depth of the see may be very long. In case of a lake, the sensors can be dropped to the bottom of the lake. In any case, the sensors are assumed to adjust their depths via any of the several mechanisms proposed in the literature.

For instance, in [14], 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 underwater. Similarly, an 'air-bladder-like device' is recommended to be used in underwater sensors for such purpose in [15]. However, since sensor will drift in the water, these mechanisms cannot fix the x and y coordinates of the sensors. Therefore, it will be better to control them via tethers from anchors which are placed at the bottom of the sea [7] or at the top with some buoys [5]. The length of the wire can then be adjusted to control the depth of the sensor as seen in Fig. 1 on AquaNode

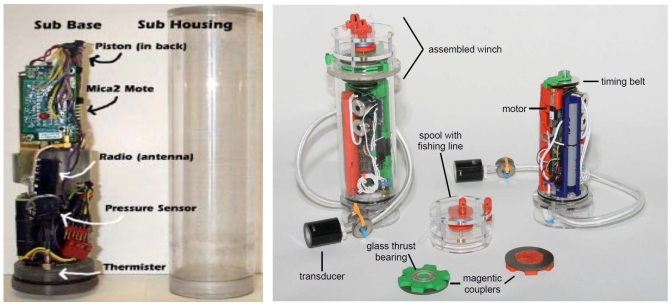

Fig. 1: Underwater sensor mote samples [14][5].

from MIT. In our work, we will assume the availability of tethered acoustic sensors and no restriction on the depth of these sensors. We also assume that these nodes can know their 3-D locations via localization techniques underwater [16].

The dropped sensors are assumed to form a connected network in 2-D (e.g., at the surface or bottom). Note that this can be achieved by uniformly randomly placing a certain number of sensors in a given area (i.e., a certain node density) with a probability of 0.99 as shown in [8]. Even if the connectivity is not achieved, the proposed work can still be applied to partitions separately. In this case, for each partition a separate surface station is needed. In our work, we assumed a single connected network which will have a single surface station deployed at the surface of the water. This surface station will communicate with an onshore station via 802.11n or WiMAX links.

An envisioned network model for our work 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 assume that UWASN can be modeled as a unit ball 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. The sensing range, s, will be the range which a sensor can sense information. This range is also assumed to be a unit ball graph as in the case of r. We will vary s with respect to r in the evaluation.

B. Problem Definition

Based on the assumptions listed, our problem can be defined as follows:"Given a connected network of n sensors each of which has a transmission and sensing range of r and s respectively and are placed at the surface of water with a given area boundaries, our goal is to compute the depths of each sensor within the water so that the coverage of the whole network in 3-D is maximized and the new network is still connected with a surface base-station that can communicate with an on-shore station". In achieving our goal, we are interested in a distributed approach which does not require any intervention from outside. The sensors will talk to each other locally and will not need any information other than their 1-hop neighbors.

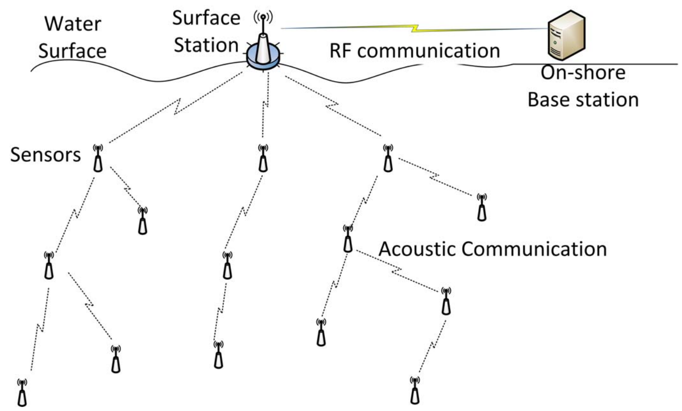

Fig. 2: Considered network model in the paper.

IV. CDS-based Depth Computation

A. Overview of the Approach

Our proposed CDS-based Depth Computation approach (CDA) for maximizing coverage is based on adjusting the depths of the sensor and sending them to 3-D locations within the water such that there will be minimal sensing overlap among the sensors. For instance, sensors which have sensing overlap in 2-D may eliminate this overlap by moving to different depths (i.e., z). Nevertheless, this may disconnect these two sensors if they move beyond their transmission range r. Therefore, we propose to adjust the depths in a restrictive manner to maintain the connectivity. For instance, for a node which has k 1-hop neighbors, the depth adjustment is done in such a way that all k neighbors still maintain their connectivity while they can go up or down as much as they can. Since this is a distributed approach, a node can determine the depths of its 1-hop neighbors and tell them to go to those depth levels locally.

Generalizing this idea, we would like to apply this to the whole network. Therefore, we determine a connected backbone of the network and assign this duty to the nodes sitting on the backbone. To this end, we decided to use the connected dominating set (CDS). In a CDS, each member, referred to as dominator, will calculate the depth for its dominatee nodes (i.e., the nodes that are not part of the backbone). In addition, it needs to do so for the neighbors who are dominators like itself. Therefore, we choose a leader node among the dominators to initiate this depth computation process. As soon as a dominator is done, it asks one of its neighbors to do the same in a recursive manner. We also make sure that this recursive computation spans all the dominator nodes. As a result, our approach can be divided into 2 phases: 1) Forming Network Backbone; and 2) Depth computation for Dominatees and Dominators. We start detailing each of these steps in the balance of this section.

B. Forming Network Backbone at 2-D

As mentioned above there is trade-off between coverage and connectivity. In order minimize coverage overlaps, nodes

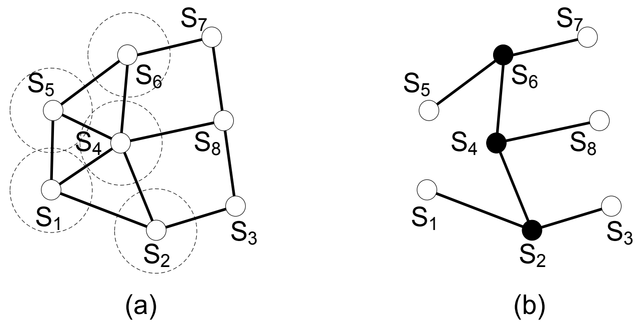

Fig. 3: (a) Initial 2-D connected UWASN. Some nodes are shown along with their sensing ranges; (b) CDS is determined as the black nodes. The dominatees are in white and each dominatee picks a dominator to connect. Our approach picks a dominator with the least sensing coverage overlaps.

have to be located far enough from each other. In that case, however, the communication link between two nodes may be broken if they are not within each others transmission range. On the other hand, if we attempt to keep all communication links of the initial network, there will be too many coverage overlaps. Hence, it is very crucial to decide which links are to be preserved while determining the depths of nodes for maximizing the coverage. The use of CDS comes into the picture at this point.

The CDS provides a connected network backbone where every other node in the network can reach this backbone in 1-hop. The elements of this backbone are called dominators and the rest of the nodes are called dominatee nodes as seen in Fig. 3. Determining a CDS in a distributed manner has been well studied in the literature. In this paper, we use the heuristic of [9] to determine the CDS with 4 message exchanges from each node. Once the dominators are determined, the rest of the nodes in the network can choose to connect to one of the dominators within its 1-hop neighborhood. Since a single link suffices, we need to determine which links to keep as part of our problem.

In our approach, we use the coverage as the criterion to make this selections. Specifically, we assign each dominatee to a dominator whose sensing coverage overlap is the minimum among other possible assignments. In this way, we strive to prevent possible overlaps when a dominatee moves to a new depth. For instance, in Fig. 3b, we picked S_6 as the dominator of S_5 as opposed to S_4 since there is some overlap among S_4 and S_5.

Once the dominators are determined, we pick a leader node among them to initiate the process of depth adjustment. This leader can be predetermined by assigning the lowest ID as the leader. If the lowest ID node is not a dominator as a result of running the CDS algorithm, then its corresponding dominator is picked as the leader. One can also prefer to run a leader election algorithm from the literature such as [10] in a distributed manner.

C. Depth Computation for Dominatees and Dominators

The proposed depth computation algorithms runs in a token-based manner. The dominator nodes having the token calculate the depths for their dominatees and neighboring dominators which have not received the token yet. Note that this is not same as token-ring based medium access control in Wireless LANs. The idea is more like a flooding-type approach where the token is broadcast to multiple dominators. The ones who had already done the computation, will ignore the message. The others will do the computation and re-broadcast the token to other neighbors and so on. This is similar to route discovery approach of Ad hoc On Demand Distance Vector Routing (AODV) [17]. At the end, all the elements of CDS will receive the token and perform the depth computation.

The nodes whose depths are calculated do not move down to their depth level immediately. Instead, they wait for the other nodes to complete depth adjustment process. After the depths for all nodes are calculated, nodes start sinking down to their own depth level.

After CDS of the network is computed, the leader node holds the token, sets its own depth level to a predefined value and initiates the depth adjustment process. As mentioned earlier, in order to maintain the connectivity while assigning depths, dominatee-dominator and dominator-dominator edges has to be preserved. Thus, the idea behind depth computation process is to keep the neighboring nodes as far as they can be, without breaking the communication link to their associated dominator as detailed below.

Let u be a dominator node holding the token (initially the leader node) where the set of dominatees of u is $V_u = \{w_1, .., w_i\}$ and the set of neighbor dominators of u is $D_u = \{w_{i+1}, .., w_n\}$. Given that the 2D coordinates (i.e., x and y coordinates) of the nodes and the transmission range (r) are known, u can calculate relative depth of w_j s.t. $w_j \in V_u \cup D_u$ using the following 3-D Pythagorean theorem:

$$w_j.z = u.z \pm \sqrt{r^2 - \left((u.x - w_j.x)^2 + (u.y - w_j.y)^2\right)} \quad (1)$$

In this formula, we call w_j as the *target node* of u, and u as the *base node* of w_j. This formula also tells us that, if we want to keep the communication link between u and w_j, at farthest w_j can be located on the surface of the sphere centered at u with a radius r. In addition, the sign of the square root denotes the location of the target node w_j relative to the base node u (i.e., above or below). If the distance between two nodes u and v is less than the twice of the sensing range in both x and y axes, locating u and v at the same depth level yields to a coverage overlap. Thus, the determination of locations of the target nodes is a crucial process in minimizing the coverage overlaps.

To deal with this problem, every dominator keeps a list of possible relative locations. The reason to keep such a relative location list is to check all possible depths for the target node and choose the most suitable one. The entries of relative location list are two-tuples of the form {node, sign} . Among the elements of this tuple, the "node" represents the base node and the target node's depth will be calculated relative to this node. Sign represents whether the target node will be placed above or below vertically w.r.t. the base node. If sign is $+$, then it means that target node will be deployed above the base node, otherwise it will be deployed below.

The proposed depth adjustment algorithm works as follows: Initially relative location list of dominator u contains two entries which are $\{\{u,+\},\{u,-\}\}$. u starts iterating its dominatee list V_u. In the first iteration, the algorithm picks $w_1 \in V_u$ and calculates two depth levels d_1 and d_2 (i.e., one depth level for each relative location list entry).

$$d_1 = u.z + \sqrt{r^2 - \left((u.x - w_1.x)^2 + (u.y - w_1.y)^2\right)}$$
$$d_2 = u.z - \sqrt{r^2 - \left((u.x - w_1.x)^2 + (u.y - w_1.y)^2\right)}$$

For the target node w_1, if the coverage overlap of w_1 with other nodes at depth level d_1 is less than the coverage overlap at depth level d_2, then the algorithm sets $w_1.z = d_1$ otherwise $w_1.z = d_2$. After the depth level is set, u adds two more entries to the relative location list which becomes $\{\{u,+\},\{u,-\},\{w_1,+\},\{w_1,-\}\}$. In the second iteration, for the node $w_2 \in V_u$, u calculates four depth levels and chooses the best one in terms of minimizing the coverage overlap. This process continues until all the dominatee and dominators level of a base node are processed.

The depth computation process is depicted in Fig. 4. In Fig. 4a 2-D projection of the nodes before depth computation is illustrated. The figure shows that $w_1, w_2, w_3 \in V_u$ are very close to each other which yields to large coverage overlaps in 2-D. Therefore, to minimize overlaps, these nodes have to be at different depth levels while preserving the connectivity to the base node u. The proposed algorithm places the first node w_1 above u. In the next iteration, w_2 can not be placed above

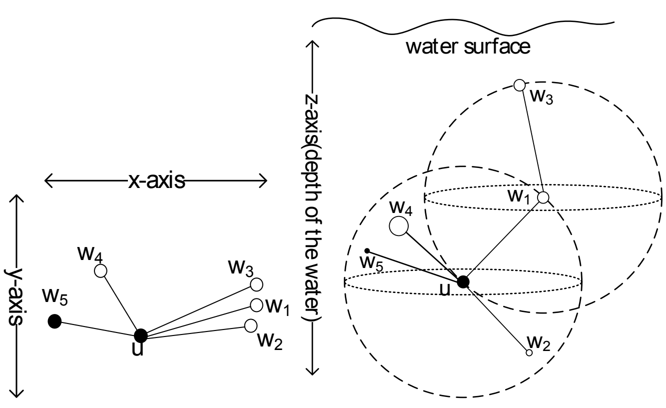

Fig. 4: (a) 2-D (bird-eye view) view of the graph before depth computation; (b) 3-D locations of nodes relative to u after depth computation. In order to give 3-D perception farther nodes have been drawn smaller. $w_1, w_2, w_3, w_4 \in V_u$ and $w_5 \in D_u$. Note that as a result of this process some nodes can be even be 2-hops away such as w_3.

Algorithm 1 DEPTH COMPUTATION

Require:
u: Starting node (Initially, the leader of the network)
$L_{previous}$: List of nodes deployed by the previous dominator. (Initially this list is empty for the leader)

Ensure: Depth levels of dominatees and neighbor dominators with maximized coverage and guaranteed connectivity

$V_u \leftarrow dominateesOf(u)$
$D_u \leftarrow neighborDominatorsOf(u)$
$L_{deployed} \leftarrow \emptyset$
$G \leftarrow \{\{u,+\},\{u,-\}\}$
for all $w_j \in V_u \cup D_u$ **do**
    if $w_j.z$ has already set **then**
        continue with the next iteration
    end if
    $minOverlap \leftarrow MAXVAL$
    $depth \leftarrow nil$
    for all $g \in G$ **do**
        $n \leftarrow g.node$, $s \leftarrow g.sign$
        $\Delta d \leftarrow \sqrt{r^2 - (n.x - w_j.x)^2 - (n.y - w_j.y)^2}$
        $d \leftarrow n.z + s \cdot \Delta d$
        $p \leftarrow overlap(w_j, L_{previous} \cup L_{deployed}, d)$
        if $p < minOverlap$ **then**
            $minOverlap \leftarrow p$
            $depth \leftarrow d$
        end if
    end for
    $w_j.z \leftarrow depth$
    $L_{deployed} \leftarrow L_{deployed} \cup \{w_j\}$
    $G \leftarrow G \cup \{\{w_j,+\},\{w_j,-\}\}$
end for
broadcast($w_j \in (L_{deployed} \cap (V_u \cup D_u)), L_{deployed}$)

u again, since it will overlap with w_1. Hence, the algorithm places w_2 below u. In the third iteration w_3 can be placed neither above nor below of u. So the next suitable location is selected which is above w_1. The dominator node u continues the iteration until the depths of all neighbor nodes are set. If the depth of a neighboring dominator $w_j \in D_u$ is previously set by another dominator, the algorithm skips that node.

After the dominator node is done with assigning depths for its dominatees and neighbor dominators, it passes the token along with the list of the nodes which are deployed by the dominator to the neighboring dominators and asks them to do the same thing for their neighbors as well. Using the list of the nodes deployed by u, the neighboring dominator will be able to avoid the coverage overlaps with its own target nodes and previously deployed nodes.

D. Pseudo-code

The pseudo code for this process is provided in Algorithm 1. We assume that the dominator node u runs the algorithm (initially the leader will run the code). The algorithm first initializes the list of dominatees and neighbor dominators (lines 1 and 2). Line 3 creates an empty list $L_{deployed}$, so that whenever the depth of any node is computed, the algorithm adds the deployed node to this list. In line 4, the algorithm initializes relative location list G with the entries $\{u, +\}$ and $\{u, -\}$. From line 5 to 24, u starts iterating its dominatees and neighbor dominators. If the depth of the node w_j which is picked at iteration j has already been computed, then the algorithm skips it and continues with the next iteration (lines 6, 7). Next, based on for each relative location in G, the algorithm calculates the depth d for w_j (lines 12, 13, 14) and then calculates coverage overlaps for w_j with the nodes in $L_{deployed} \cup L_{previous}$ at depth level d (line 15). If the overlap is less than the minimum overlap so far, the algorithm replaces best depth level with the new one (line 18). When the most suitable depth level is computed for w_j, the algorithm sets the depth of w_j to d (line 21) and adds w_j to the list of deployed nodes $Ldeployed$ (line 22) and also updates G with new relative locations (line 23). At the end of the iteration, the algorithm will be done with depth computation and it will broadcast a message to its dominatees and neighbor dominators whose depth levels are computed (line 25).

E. Algorithm Analysis

This section provides the message and run-time complexity of the proposed CDA approach.

Theorem 1: Assuming that the leader node is predetermined (i.e. no message cost for it), the worst-case message complexity of the proposed CDA approach per node and for the whole network is $O(1)$ and $O(n)$ respectively where n is the number of nodes.

Proof: There are two main phases of the proposed approach 1) Network Backbone Formation (i.e., CDS computation), 2) Depth Computation. In Network Formation phase, for all nodes the approach identifies dominator and dominatee nodes. In order to determine whether a node is a dominator or a dominatee, each node sends 4 messages [9]. In the second phase, starting from the leader node, each dominator calculates the depth for its dominatee and neighbor dominators, and broadcasts one message to announce the calculated depth to its neighbors and passes the token to the neighboring dominators. Therefore the message complexity for a single node is $O(1)$. Since there are n nodes in the network the overall message complexity is $O(n)$. ■

Theorem 2: The worst case time complexity of the depth computation process is $O(d^2)$ where d is the maximum node degree of a node (i.e., number of neighbors) in the CDS graph of the network.

Proof: There are two nested for loop in Algorithm 1 (i.e., lines from 5 to 24 and lines from 11 to 20). Let the cardinality of $V_u \cup D_u$ be d. The outer for loop iterates d times. The inner loop iterates twice the number of deployed nodes which is at most $2 \cdot d$. Therefore the overall worst case time complexity is $O(d^2)$ ■

V. Performance Evaluation

A. Simulation Setup and Baseline for Comparison

Two parameters are varied in the simulations, namely the transmission range/sensing range, referred to as α and number of nodes. In our first set of simulations we fixed the sensing range (s of the sensor nodes to 10m and number of nodes are fixed to 700. The transmission range (r) is calculated using $r = s{\cdot}\alpha$. Deploying 700 nodes uniform randomly to the area of interest (100m $\times$ 100m, with a maximum depth of 500m) we run the simulations for varying α values where $0.5 \leq \alpha \leq 3$. In the second set of simulations, we fixed s to 10m and r to 18 (i.e., α is fixed to 1.8) and varied number of nodes from 500 to 900.

The following performance metrics are used to evaluate the performance of the proposed approach:

- *Coverage*: The coverage metric indicates the ratio of total volume covered to the theoretical maximum volume which can be calculated as;
$$V_{max} = n \cdot \frac{4}{3}\pi s^3 \tag{2}$$
where n denotes the number of sensor nodes deployed. In other words, theoretical maximum presents a situation where, the nodes have no overlaps. Thus coverage percentage formula can be stated as follows:
$$CoveragePercentage = \frac{V_{covered}}{V_{max}} \tag{3}$$
where $V_{covered}$ is total volume covered by sensors.
- *Connectivity*: This metric indicates whether all the nodes in the network have a path to the surface station or not. Obviously, our goal is to have all the nodes connected to the surface station. To check this connectivity, we run a depth first search on the resultant topologies and calculate the number of connected components. If this number is 1, the network is connected. Otherwise, the network will be disconnected but the number of connected components will give us further idea about the number of disconnected nodes in the network.

We compared our approach with one of the contemporary approaches which is Cluster-based Graph Coloring (CGCA) approach presented in [6]. In CGCA, the authors cluster the nodes according to the node IDs where the node having highest node ID locally is elected as the cluster head. Then using a distributed graph coloring algorithm each cluster head minimizes the coverage overlaps among its neighbors. While this approach does not guarantee coverage, we will investigate how close we are to their coverage results.

B. Performance Results

This section provides the performance results. Each individual simulation involves 100 different topologies and the average result is reported. We observed that with a %95 confidence interval, our results stayed within %5-%10 of the sample mean. These are shown with the error bars in the graphs.

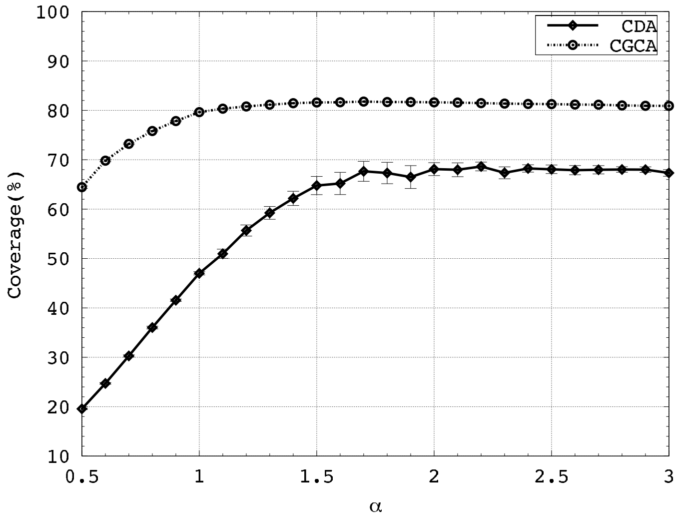

Fig. 5: Coverage percentage comparison for varying $\alpha = \frac{r}{s}$ where $n = 700$ and $s = 10$.

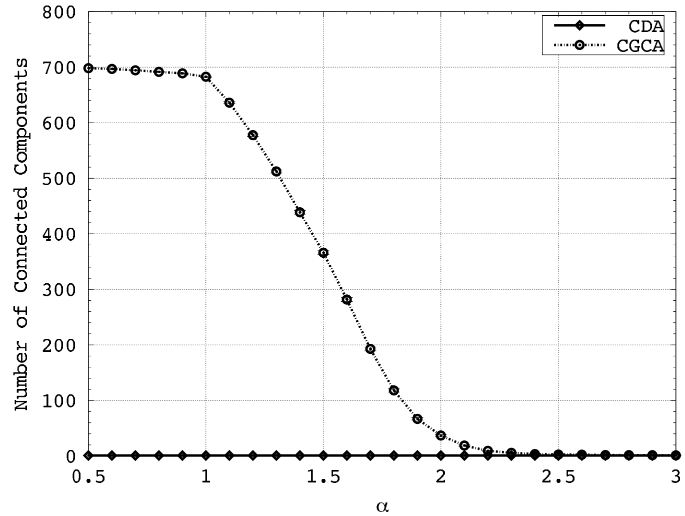

Fig. 6: Number of connected components comparison for varying α.

1) Experiments with Varying α: We first conducted experiment by varying the α from 0.5 to 3. The results are depicted in Fig. 5. The figure demonstrates that coverage percentage of all approaches increase as the r/s ratio increase. The reason for such an increase is that the higher r/s ratio means the higher transmission range. If the transmission range increases, then nodes can be located farther from each other which decreases the coverage overlap and maximizes the coverage.

Figure 5 also shows that the proposed CDA performs very close to CGCA in terms of coverage percentage. The performance gap between CDA and CGCA is almost constant and around %10, when $\alpha \geq 2$. When $r < 2 \cdot s$, then there will be a coverage overlap between two nodes u and v if we keep the communication link between them. However, for the α values greater than 2, nodes do not overlap much even if the link between them is maintained. Therefore, the performance of all algorithms becomes stable for $\alpha \geq 2$.

We also check the connectivity of the resultant topologies for all approaches as shown in Figure 6. As we have claimed, CDA approach always forms only 1 connected component regardless of the value of α. However, this is not the case for CGCA. Up to $\alpha = 2.5$, the network is not connected. This is because CGCA strives to maintain a high coverage ratio which causes to break the links among the nodes. Eventually, no paths exists between the nodes to the surface station. However, the number of connected components for CGCA reduces as α increases. When $\alpha = 2.5$, the network becomes connected.

These results indicate the trade-off between coverage and connectivity. Since connectivity is a must for data collection, then CDA should be preferred as long as $\alpha < 2.5$. Otherwise, CGCA can be a better option.

2) Experiments with Varying Node Count: In the next set of simulations, we evaluated the performance of the CDA approach in terms of coverage percentage for varying number of sensor nodes from 500 to 900. The results shown in Figure 7 indicates that the performance of both approaches

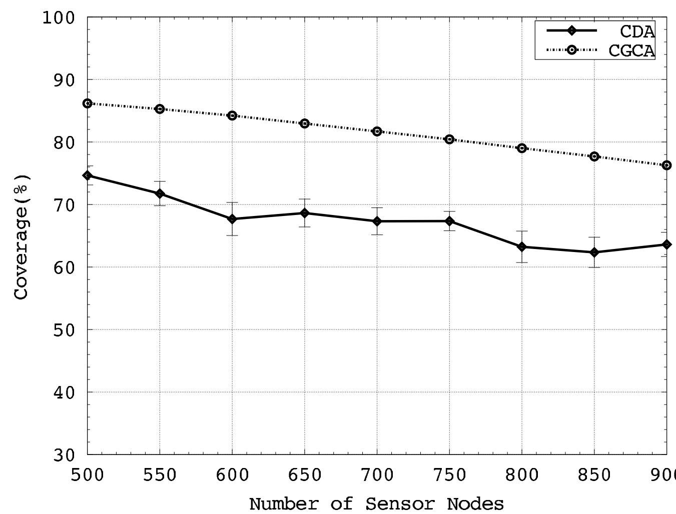

Fig. 7: Coverage percentage comparison for varying number of sensor nodes where $r = 18$ and $s = 10$.

gets worse as the number of sensors increases. While this seems conflicting, it can be explained as follows: As we increase the number of nodes, the theoretical upper bound V_{max} defined in Equation 2 increases. According to Equation 3, coverage percentage would decrease with the increased V_{max}. While the coverage we provide may increase with the increased node count, there will also be a lot of overlaps which will compensate the possible increase in the coverage. As a result, the coverage ratio will decrease. This decrease is more significant with CDA since there are more overlaps due to connectivity constraint. Some of the nodes cannot be deployed elsewhere to maintain connectivity.

We repeated the connectivity experiment by measuring the connectivity of the resultant topologies. Again, for any number of nodes, the resultant topology in CDA is connected as seen in Figure 8. However, this was not the case for the topologies

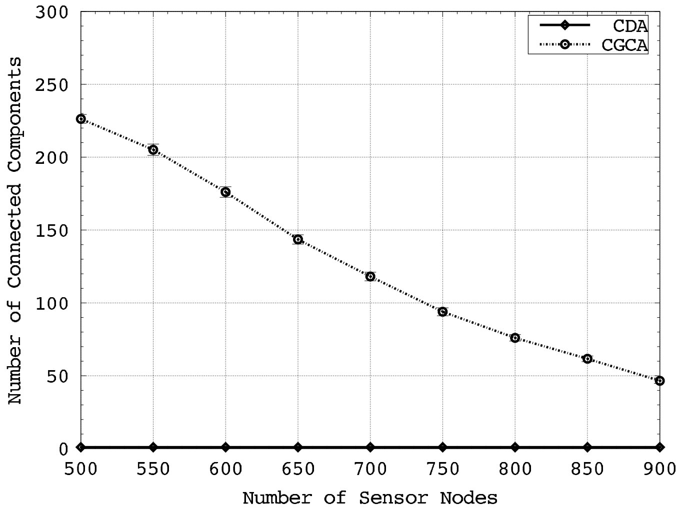

Fig. 8: Number of connected components comparison for varying number of sensor nodes where $r = 18$ and $s = 10$.

created as a result of CGCA. We observed that the number of connected topologies reduces as the increased node count but it never reaches 1. More nodes would be needed to guarantee connectivity. One of the reasons behind this results is that α value was set to 1.8 and this value is not good enough for CGCA to guarantee connectivity as shown in Figure 6.

VI. Conclusion

Due to inaccessibility of the environment, a distributed deployment mechanism is required in order to provide flexibility in realizing many UWASN applications which operate in inhospitable environments. Assuming that sensors are uniform randomly distributed to the area of interest by dropping them from a helicopter, the goal is to compute the depths of the sensors while maximizing the total sensor coverage and guaranteeing the connectivity of the nodes to the surface station. In this paper we have presented a purely distributed node deployment mechanism for UWASN with the idea of changing the depths of the sensors assuming that sensors can only move in vertical direction. The proposed approach is based on connected dominating set (CDS) of the sensors and runs in token based manner. The dominator holding the token (initially the leader node) computes the depths for its dominatees and neighbor dominators relative to itself while minimizing the coverage overlaps among its dominatees and neighbor dominators and preserving the communication links with them. After the dominator is done with the depth computation, it passes the token to its neighbor dominators and asks them to do the same computation. After all the depth computation is done for all sensors, they start sinking down to the computed depth levels.

We have validated the performance of the proposed through simulation. We have shown that our proposed approach perform very close to the baseline approach even though the network connectivity is guaranteed.

As a future work, we plan to extend the ideas in this paper considering an initially disjoint network on the surface of the water. This will require using multiple surface stations.

Acknowledgment

This work is supported in part by a research grant from TUBITAK EEEAG (Grant Number 113E234). The numerical calculations reported in this paper were partially performed at the Institute of Industrial Science, the University of Tokyo.

References

[1] 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.

[2] 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.

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

[4] J.-H. Cui, J. Kong, M. Gerla, and S. Zhou, "Challenges: building scalable and distributed underwater wireless sensor networks (uwsns) for aquatic applications," *Channels*, pp. 1–17.

[5] C. Detweiler, M. Doniec, I. Vasilescu, E. Basha, and D. Rus, "Autonomous depth adjustment for underwater sensor networks," in *Proc. of ACM Int'l Workshop on UnderWater Networks*, ser. WUWNet '10, New York, NY, USA, 2010.

[6] K. Akkaya and A. Newell, "Self-deployment of sensors for maximized coverage in underwater acoustic sensor networks," *Comput. Commun.*, vol. 32, no. 7-10, May 2009.

[7] D. Pompili, T. Melodia, and I. F. Akyildiz, "Deployment analysis in underwater acoustic wireless sensor networks." in *Proc. of ACM Int'l Workshop on UnderWater Networks (WUWNet)*, Los Angeles, CA, September 2006.

[8] C. Bettstetter, "On the connectivity of ad hoc networks," *The computer journal*, vol. 47, no. 4, pp. 432–447, 2004.

[9] J. Wu and H. Li, "On calculating connected dominating set for efficient routing in ad hoc wireless networks," in *Proc. of the 3rd Int'l workshop on Discrete algorithms and methods for mobile computing and communications*, ser. DIALM '99, New York, NY, USA, 1999.

[10] N. Malpani, J. L. Welch, and N. Vaidya, "Leader election algorithms for mobile ad hoc networks," in *Proc. of the 4th Int'l workshop on Discrete algorithms and methods for mobile computing and communications*, ser. DIALM '00, New York, NY, USA, 2000.

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

[12] S. N. Alam and Z. Haas, "Coverage and connectivity in three-dimensional networks." in *Proc. of ACM Int'l Conf. on Mobile Computing and Networking (MobiCom'06)*, Los Angeles, CA, Sept. 2006.

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

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

[15] 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.

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

[17] C. Perkins, E. Belding-Royer, and S. Das, "Ad hoc on-demand distance vector (aodv) routing," United States, 2003.