An Efficient Method for Constructing Underwater Sensor Barriers

Stanley Barr, Jie Wang, Benyuan Liu
Computer Science Department, University of Massachusetts Lowell, Lowell MA, USA

***Abstract*— Anti-submarine warfare(ASW) is a critical challenge for maintaining a fleet presence in hostile areas. Technology advancement has allowed submarines to evade standard sonar detection. A viable alternative is to place magnetic or acoustic sensors in close proximity to possible underwater pathways of submarines. This approach may require deploying large-scale underwater sensor networks to form 3-dimensional barriers.**

We present new results on sensor barriers for 3-dimensional sensor networks. We first show that when sensor locations follow a Poisson Point Process then sensor barriers in a large 3-dimensional fixed emplacement sensor field are unlikely to exist. We use the notion of 3-dimensional *stealth distance* to measure how far a submarine can travel in a sensor network without being detected.

We devise an energy-conserving scheme to construct 3-dimensional barriers using mobile sensors. We focus on developing an energy efficient matching of mobile sensors to cover grid points using distributed auction algorithms. Specifically, we try to minimize the maximum travel distance between any sensor and its assigned grid point. Through simulation we show in comparison to the optimal solution, the distributed auction based approach offers reduced computation time and similar maximum travel distance. This provides a promising new approach to constructing barriers in 3-dimensional sensor networks.

I. Introduction

ASW is a critical challenge for maintaining a fleet presence in hostile areas. International submarine sales on today's markets are not covered by any nonproliferation treaty, making it possible for nation states, organizations, or even individuals with sufficient resources to purchase submarines. Some of these submarines are capable of launching cruise missiles to deliver conventional, nuclear, chemical, or biological payloads [1]. Drug traffickers have, also, used submarines to smuggle illegal drugs into the US. The Washington Post reported that a total of thirteen submersible vessels of this kind were captured in 2007 [2], while many more were believed to have evaded US coastal patrols.

To make things worse, recent technology advances have made it possible for submarines to evade standard sonar detection [3]. In particular, submarine hulls can be fitted with rubber anti-SONAR protection tiles to thwart active SONAR detection. These rubber tiles, also, dampen intra-submarine noise to thwart passive acoustic detection at a distance.

Thus, finding alternative ways to detect submarines becomes an important and timely problem. Using magnetic or acoustic sensors in close proximity to the possible underwater pathways of submarines is a viable new approach. Recent advances in micro electro-mechanical systems and communication hardware have made this approach more reliable and relatively inexpensive with a decreased power consumption [4]. Large-scale deployments of underwater wireless sensor networks are expected to become a reality in the near future. These deployments may occur by dispersal from an aircraft or artillery ordinance. Sensors with different buoyancy may submerge to different depths. A sensor node may be equipped with a small ballast tank to control its weight and obtain different buoyancy. Such deployment strategies can deploy many sensors over a vast space quickly, but limit the control of sensor placement.

We want to form a barrier in a sensor network to detect moving objects. In 2-dimensional (2D) terrestrial strip sensor networks, a barrier is a chain of sensors from one end of the strip to the other end with overlapping sensing zones of adjacent sensors. No moving objects, regardless which paths they choose to pass through, can cross the chain undetected. In 3-dimensional (3D) underwater sensor networks, however, forming a barrier is much more subtle. For example, even if an unbroken chain of overlapping sensor zones between two opposing sides exists in a cuboid, intruders may still be able to pass under or over such a chain without being detected. Thus, an overlapping sensor chain no longer constitutes a barrier in a 3D space. Instead, a barrier in a 3D space should be a set of sensors with overlapping sensing zones of adjacent sensors that covers an entire (curly) surface that cuts across the space.

The goal of our research is to construct a 3D barrier that is scalable for large scale coastline protection in a timely fashion. As underwater movement is very power intensive our solution should minimize the maximal travel distance required of any sensor comprising our barrier. This will maximize the residual energy and allow for a longer area coverage period.

In order to form an underwater water barrier we plan on deploying sensors to set of predefined fixed grid points based on the sensor's detection radius. Such a deployment create a vertical barrier with no holes.

We assign and ultimately move sensors to grid positions based on grid-matching using the Hungarian method, a classic centralized approach. Through simulation we develop a rough lower bound on the maximum travel distance any one sensor needs to travel. This also gives us an upper bound on the computation time. No approximate solution should take longer than an optimal solution.

We first consider using a centralized approximate solution based on auction theory to reduce the computation cost. In many scenarios auctions provide a computationally feasible

Manuscript received January 5, 2011; revised May 10, 2011; accepted August 03, 2011.

doi:10.4304/jcm.6.5.370-383

alternative to the optimal solution. We compare the performance characteristics of the auction based approach to the optimal approach. We show that for our application auction algorithms provide desirable tradeoffs between computation time and maximum travel distance.

We finally consider a fully decentralized underwater sensor network where nodes must coordinate with each other and use auctions to construct a barrier. We break the barrier surface up into discrete zones. Each zone will elect a leader. The leader will coordinate an auction among available local sensors and their movement to their final positions.

Underwater 3D sensor networking is not a mature research area. We present the first set of results for constructing a barrier to detect intruding submarines in a 3D sensor network where sensor nodes are distributed uniformly at random modeled by a Poisson Point Process.

We show that a barrier in such a 3D sensor network for a finite density of sensors is unlikely to exist. This suggests the necessity to deploy sensors with at least limited mobility. Minimizing the energy consumed by the movement of underwater sensors is an important issue since solar panels may not be applicable for submerged sensors.

We compare the optimal centralized approach, to an centralized approximation, and finally to a decentralized approximation. We also analyze how environmental parameters and sensor density impact our barrier construction. Specifically we show the paremeters affect total time for construction and maximal movement required by sensors.

This paper is organized as follows. In Section II, we review work related to our research. In Section III, we develop our network model. Then we demonstrate our results for strong barrier coverage on 3D sensor networks and introduce the notion of stealth distance for 3D sensor networks. In Section IV, we describe our approach to mitigating issues raised in the previous section. We simulate and compare the centralized optimal solution based on the Hungarian method with the approximate solution based on Auction algorithms. In Section V, we develop a fully decentralized solution based on auction algorithms to construct a barrier in a timely manner while limiting the maximum movement required by any one sensor. In Section VI, we discuss our conclusions.

II. Related Work

The problem of area coverage for wireless sensor networks has been studied intensively since earlier in the 1990's. Coverage of a grid-based sensor network was considered in [5]. The authors derived requirements for the sensing range and sensors failure rates to cover fully a given area.

The problem of covering an arbitrary number of points with at least k sensors is known as the k-Coverage Problem and has been studied by several authors (see, e.g. [6]–[8]). Research has been conducted to understand the relationship between area coverage and network connectivity [9], [10].

The notion of *barrier coverage*, whose goal is to arrange sensors to detect intruders passing through a region, was first introduced in the context of cooperating robots for military applications and has been extensively studied recently for various scenarios [11]–[19]. Examination of an intruder's ability to navigate through a sensor field undetected is a well studied problem [20]–[24]. Determining if within a belt region that an intruder would be detected by at least k sensors is known as the k-Barrier Coverage Problem [25]. The authors proposed algorithms to determine if a region is k-barrier covered, established deployment pattern to guarantee k-barrier coverage, and derived conditions of k-barrier coverage in a randomly deployed sensor network.

Covering a set of points to minimize the number of sensors employed to cover the area or maximize network lifetime is known as the Point Cover Problem and has been studied in [26]. The average distance an intruder can move through a 2D sensor field without being detected has been developed into a quality of surveillance metric [27].

Although not in the context of barrier coverage, the authors of [28] reference the use of magnetometer enabled motes to detect moving submarines at distances of several hundred meters. This seems reasonable given the sensitivity required to detect submarines using magnetometers and the capabilities of magnetometer-based sensor boards for motes [29], [30].

Underwater sensor networks has recently been an active area for research. As previously stated, new threats [2] and new technologies [2] have given researchers the required impetus to research in this area. However, the economics of ocean operations is a significant hurdle in the demonstration of viable systems. Three of the major economic components of sensor networks are fabrication, deployment, and recovery [31]. Fabrication of underwater components is expensive. Components are exposed to corrosive sea water, must be able to withstand high water pressure, and be resilient to constant movement. Currently, the most common deployment mechanism involves using ships or airplanes to deliver sensors to the deployment zone. Finally, although current low cost technologies make it possible to deploy underwater sensors, most are simply too expensive and potentially too sensitive to be abandoned at sea. Corrosion and fouling can often lead to the requirement for sensor maintenance. Frequently, this requires the use of ships to once again collect the devices.

There are many open research areas when it comes to underwater networking. The authors of [32] point out that the acoustic communications channel is filled with technical challenges. The available bandwidth is very limited and is impacted by multi-path and fading [33], [34]. Propagation delay is roughly 1500m/s which significantly impacts many traditional MAC layer communications strategies [35]. Cross layer optimizations for improved application performance are open issues [36], [37].

Auction algorithms were introduced in 1987 as approximate solutions to the assignment problem [38], which creates a one-to-one mapping between equinumerous sets. They were extended to provide approximate optimal solutions to the transportation problem, where the cardinality of the two sets are different [39]. The author exploits a mechanism which allows him to create equinumerous sets from his data. Later, auction algorithms were extended to run in massive architectures [40]. Since that early work, they have been used in a variety of research activities from task allocation for robots moving objects to peer-to-peer satellite refueling [41] [42] .

III. Network Model of 3-dimensional barriers

A. Network Model

We consider a sensor network consisting of sensors deployed in a large-scale 3D rectangular cuboid. For the initial configuration, we assume that the locations of these sensors are uniformly and independently distributed in the cuboid. Such a random initial deployment is desirable in scenarios where prior knowledge of the region of interest is not available. This may be the result of certain deployment strategies. Under this assumption, the sensor locations can be modeled by a stationary 3D Poisson Point Process. The density of the underlying Poisson Point Process is denoted by λ. The number of sensors, $N(R)$ located in a 3D region R, follows a Poisson distribution with parameter $\lambda\|R\|$, where $\|R\|$ represents the volume of the region. We have

$$\mathbb{P}(N(R)=k)=\frac{e^{-\lambda\|R\|}(\lambda\|R\|)^k}{k!}. \quad (1)$$

We assume that each sensor can sense the environment and detect intruders in the 3D sphere of radius r. A point is said to be *covered* by a sensor if it is located in the sensing sphere of the sensor. The sensor network is partitioned into two parts: the covered space, which is the space covered by at least one sensor; and the uncovered space, which is the complement of the covered space. An intruder is said to be *detected* if its path lies within the covered space.

In reality, the sensing range of a sensor is usually not of a spherical shape due to hardware and environmental factors. Nevertheless, the sphere model can be used to approximate the real sensing range and provide bounds for the real case. For example, the irregular sensing area of a sensor can be lower and upper bounded by its maximum inscribed and minimum circumscribed circles, respectively.

For the rest of this paper, the shorthand $X \sim \exp(\mu)$ stands for $\mathbb{P}(X<x)=1-\exp(-\mu x)$, i.e., X is a random variable exponentially distributed with parameter μ.

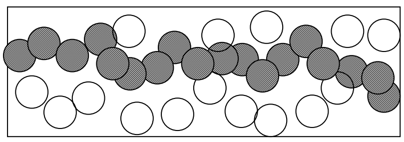

Figure 1: A 2D barrier.

B. 3D Sensor Barriers

In a 2D sensor network on a strip area, the barrier is directly related to the percolation of the network model [12], [24]. These connected sensor clusters act as "trip wires" to detect any crossing intruders as shown in Figure 1.

In a 3D sensor network, however, a sensor cluster connecting the opposite surfaces of the cuboid no longer constitutes a barrier that can detect crossing intruders. There could be "holes" where an intruder can cross the cuboid undetected. Percolation of sensors no longer provides barrier coverage and intruders can evade detection via the uncovered space. Figure 2 illustrates this effect, where the figure at the right-hand side shows the projection of the 3D cuboid in the direction of the arrow.

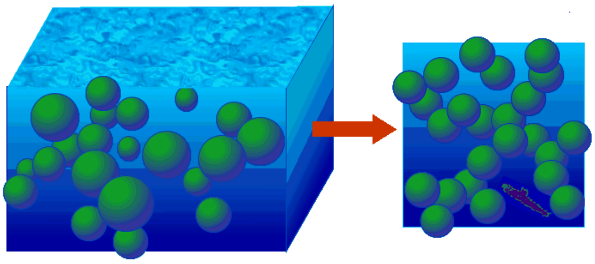

Figure 2: Barrier coverage in 3D sensor networks.

We consider a 3D cuboid of size $l \times w \times d$, where l, w, and d denote the length, width, and depth of the cuboid, respectively. A *crossing path* is a path that connects one surface of the cuboid to the opposite surface, where the ingress point and the egress point reside on two opposite surfaces of the cuboid. A crossing path is said to be covered if it intercepts at least one sensor. Intruders moving along covered crossing paths will be detected. A network provides a *barrier*, if any crossing path attempted by an intruder results in its being detected. A 3D sensor network is said to have a barrier in a specific direction if any crossing path intersecting the two surfaces perpendicular to the direction goes through the sensing zone of at least one sensor.

Theorem 1: A 3D sensor network is said to be k-barrier covered in a particular direction if

$$P(cp \text{ is } k\text{-covered}) = 1, \quad \text{w.h.p.}$$

where cp is any crossing path in the direction of interest.

Without loss of generality, we assume that intruders attempt to penetrate the cuboid in the direction of depth. Clearly, to form a 3D barrier requires a continuous surface that is fully covered by sensors. The following theorem characterizes the existence of 3D space barriers.

Theorem 2: In a 3D cuboid of size $l \times w \times d$ where sensors are distributed according to a Poisson Point Process of density λ, no barrier exists in the cuboid in the direction of depth for finite sensor density λ and depth d as $l, w \to \infty$.

Proof. Projecting the 3D cuboid in the direction of the cuboid depth results in a 2D surface of size $l \times w$. After the projection, a "barrier tunnel" in the original 3D cuboid where an intruder can pass through may not be present in the projected 2D surface. Figure 3 shows two different projections of four sensors deployed in a 3D environment. On the left-hand side, we can see clearly a hole that can be used by an intruder to cross undetected. However, in the projection at the right-hand side, no hole is seen. Therefore, the barrier coverage of the 3D cuboid is bounded above by the barrier coverage of its projected 2D surface. If an intruder can penetrate the projected 2D surface undetected then there exists for sure an uncovered crossing path in the 3D cuboid.

In the asymptotic case, when $l, w \to \infty$, the sensors on the projected 2D surface follow the Poisson point process

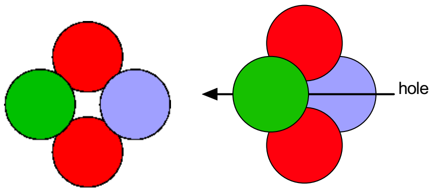

Figure 3: Two different projections of the same sensors deployed in a 3D environment.

distribution of density λd, with each point occupied by a disk of radius r. The fraction of the area that is covered in the projected 2D surface, f_a, can be readily obtained using the result of [12][Theorem 1]. That is,

$$f_a = 1 - e^{-\lambda d \pi r^2} < 1 \text{ if } \lambda, d < \infty.$$

Therefore, there is uncovered area in the projected 2D surface for finite sensor density and cuboid depth. An uncovered area in the project 2D surface corresponds to a "barrier tunnel" in the original 3D space where an intruder can pass through undetected. As a result, there is no barrier in the original 3D cuboid in the direction of depth. Simply stated, we can always make the cuboid large enough to create a hole for any fixed number of sensors. □

C. Stealth Distance in 3D Sensor Networks

In a wireless sensor network, as it moves across the network, an intruder will be detected whenever its path intersects with the sensing sphere of a sensor. It is interesting to investigate the distance an intruder travels before first being detected by sensors. This distance, referred to as the *stealth distance* and defined below, measures the intrusion detection performance of the sensor network.

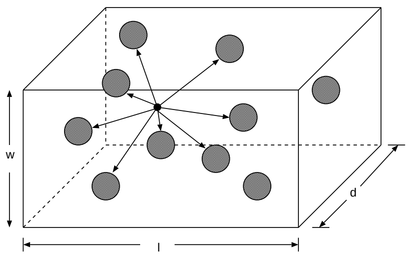

Figure 4: Stealth distance of intruder in 3D sensor networks.

Definition 1: Consider a 3D sensor network where sensors of sensing range r are distributed according to a Poisson point process of density λ. Assuming that an intruder is initially undetected and moves in a random direction along a straight line. The stealth distance of the intruder, X, is defined as the distance it travels before it is first detected by any sensor.

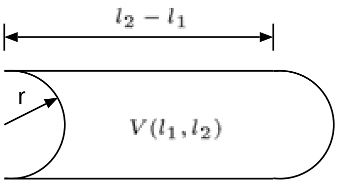

Figure 5: Projection of volume $V(l_1, l_2)$ covered by an intruder when moving along a straight line.

The notion of stealth distance was introduced in [27], and the authors derived an approximation of the expected stealth distance for 2D sensor networks. In the following theorem, we characterize the distribution of the stealth distance in a 3D sensor network. The distribution of the stealth distance in a 2D sensor network can be obtained similarly.

Theorem 3: In a 3D sensor network where sensors are distributed according to a Poisson point process of density λ, the stealth distance of an intruder, denoted by X, follows an exponential distribution with parameter $\lambda \pi r^2$, i.e.,

$$\mathbb{P}(X < x) = 1 - e^{-\lambda \pi r^2 x}. \tag{2}$$

Proof: We first show that for an intruder moving on a straight line in a random direction, the sequence of distances at which the intruder intersects the sensing sphere of a new sensor forms a Poisson process of intensity $\lambda \pi r^2$.

Let $V(l_1, l_2)$ denote the 3D region newly covered by the intruder when it moves on the straight line from location l_1 to location l_2, whose volume is denoted by $||V(l_1, l_2)||$. The projection of the volume is illustrated in Figure 5.

We show that the number of sensors whose sensing spheres intersect the intruder within interval $[l_1, l_2]$ is Poisson distributed with parameter

$$\lambda \pi r^2 (l_2 - l_1).$$

The probability that a sensor initially located at point $\mathbf{x} \in \mathbf{R^3}$ intersects the intruder within the moving interval $[l_1, l_2]$ is denoted by $\mathbb{P}(\mathbf{x} \in \mathbf{V(l_1, l_2)})$. This probability only depends on the direction and moving distance of the intruder. In particular, it does not depend on the initial Poisson process giving the positions of the sensors. We can thus define a thinned Poisson point process $\Phi(l_1, l_2)$ by selecting the sensors whose sensing spheres will intersect the intruder within the interval $[l_1, l_2]$. This is a non-uniform process with density

$$\lambda'(\mathbf{x}) = \lambda \mathbb{P}(\mathbf{x} \in \mathbf{V(l_1, l_2)}).$$

Note that $\Phi(l_1, l_2)$ is a set of points and the number of points contained in it, denoted by $|\Phi(l_1, l_2)|$, is a random variable. The number of sensors whose sensing spheres intersect the intruder within interval $[l_1, l_2]$ is equal to the number of points in the thinned process, which follows the Poisson distribution with mean

$$\begin{aligned}
\mathbb{E}(|\Phi(l_1, l_2)|) &= \int_{\mathbb{R}^3} \lambda'(\mathbf{x}) \mathbf{dx} \\
&= \lambda \int_{\mathbb{R}^3} \mathbb{P}(\mathbf{x} \in \mathbf{V(l_1, l_2)}) \, \mathbf{dx} \\
&= \lambda \int_{\mathbb{R}^3} \mathbb{E}\big(1_{\{\mathbf{x} \in \mathbf{V(l_1, l_2)}\}}\big) \, \mathbf{dx} \\
&= \lambda \mathbb{E}\Big(\int_{\mathbb{R}^3} 1_{\{\mathbf{x} \in \mathbf{V(l_1, l_2)}\}} \mathbf{dx}\Big) \\
&= \lambda \mathbb{E}(||V(l_1, l_2)||) \\
&= \lambda \pi r^2 (l_2 - l_1), \qquad (3)
\end{aligned}$$

where $1_{\{\cdot\}}$ denotes the indicator function of the event $\{\cdot\}$. Furthermore, it is straightforward to see that $\mathbb{E}(||V(l_1,l_2)||) = \pi r^2(l_2 - l_1)$.

We note that sensors whose sensing spheres intersect the intruder within disjoint moving intervals are independent. This is because if $[l_1,l_2] \cap [l_3,l_4] = \emptyset$, then each sensor is either selected in $\Phi(l_1,l_2)$, or in $\Phi(l_3,l_4)$, or not selected at all. Therefore, $\Phi(l_1,l_2)$ and $\Phi(l_3,l_4)$ are two independent processes.

Thus, the sequence of distances at which the intruder intersects the sensing spheres of a new sensor is a Poisson process of intensity $\lambda\pi r^2$.

When the intruder intersects the sensing sphere of a sensor, it is immediately covered by the sensor until it moves out of the covering sphere of the sensor. There is no constraint on the number of sensors covering the intruder. Therefore, the covered/uncovered sequence experienced by the intruder can be seen as a $M/G/\infty$ queuing process, where the service time is the time duration that the intruder is covered by a sensor until it moves out of the sensors covering sphere. The idle period of the $M/G/\infty$ queue corresponds to the time duration that the point is not covered. Idle periods in such queues have exponentially distributed durations [43]. Therefore, we have

$$X \sim \exp(\lambda\pi r^2).$$

□

It follows from Theorem 3 that the expected stealth distance of a randomly located intruder is

$$E[X] = \frac{1}{\lambda\pi r^2},$$

which is inversely proportional to the sensor density (λ) and the projected area of the sensing sphere (πr^2). Therefore, in order to shorten the stealth distance of an intruder, one can add more sensors, or use sensors with larger sensing range. To guarantee that the expected stealth distance of an intruder be smaller than a specific threshold l_0, we should have

$$\frac{1}{\lambda\pi r^2} < l_0.$$

The above relationship between the stealth distance and the deployment parameters (sensor density and sensing range) provides important guidelines to the planning of sensor networks for intrusion detection.

IV. Centralized Solution

We believe simulation is a valid mechanism for vetting concepts in the construction of large 3D barriers. This is because there are many issues in developing for the deployment of large-scale sensor networks. Here is short list of some of the major issues researchers will face: conducting networking experiments that are too time consuming, too expensive to do in real life, or where there is limited availability of components.

The first step in constructing our 3D barrier is the development of some basic intuitions about the nature of the problem. Our first question is "What is the time complexity needed to construct the optimal assignment for varying sizes of sensor fields?". The optimal solution will give an upper bound on the time complexity allowed to construct approximate solutions. Our second question is "What is the maximum travel distance required of any sensor in the optimal solution and how does it change for varying sizes of sensor fields?". This allows us to understand the movement distance requirements that must be placed on sensors during deployments. It also allows us to compare approximate solutions to the optimal solution. This section analyzes the optimal solution and offers an alternative approximate solution that preserves the maximal travel distance while offering a considerable time complexity improvement.

A. Moving Barriers

The results in the previous section show that barriers in a 3D network are unlikely to exist with Poisson distributed stationary emplacement sensors. To overcome this obstacle, we propose to use mobile sensors to form a barrier. To accomplish this task, we require that sensors have the following three features:

- **X, Y, Z positional control -** Be able to move across the surface of the ocean and also to different depths.
- **Radio based communications -** Be able to communicate in an untethered manner.
- **Localization -** Be able to identify their own X, Y, Z position to some degree of certainty.

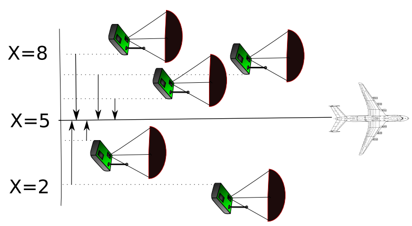

Figure 6: Phase one: an air drop of sensors along a path resulting in scattered placement along that path.

We note that sensors may be able to localize their own positions, after their deployment, using a set of anchor nodes. Each of the anchor nodes is equipped with an underwater acoustic modem that submerges in the water, and a GPS that stays on the water surface. The GPS provides the location information of the node. An underwater sensor may communicate with a set of anchor nodes through their underwater acoustic modems and use the propagation delays of acoustic signals and the locations of the anchor nodes to calculate its location.

We propose a three phased approach for constructing a 3D barrier, which minimizes the max energy used by any sensor:

1) Find an optimal vertical plane at which the sensors will form a grid-based barrier as shown in Figure 6.
2) For each sensor, identify its best grid-based assignment as shown in Figure 7.
3) Move sensors from their initial dropped position directly to their final assigned grid positions.

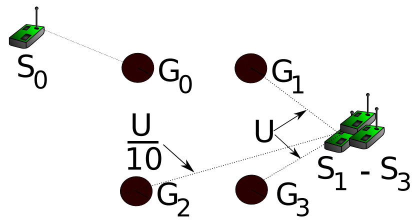

Figure 7: Phase two example where sensors $S_1 - S_3$ derive the same utility U, from being assigned to G_1 or G_3. All receive a minute utility ($.1U$), from being assigned to G_2. S_0 does not have the energy to move to any grid point other than S_0. An auction occurs to assign $S_1 - S_3$ to $G_1 - G_3$.

B. Finding the optimal barrier location

Suppose as in Figure 6 an airplane drops a set of mobile sensors such that plane X is parallel to the coast line to be protected. The sensors must move to some plane X, such that X minimizes the energy expended by any one sensor. For this calculation, we are assuming that the sensors will only be moving in one direction to approach the X location.

Suppose X represents the optimal line for each of the m sensors to approach. Let $d(x_i, x)$ be $|x - x_i|$ the distance traveled by any sensor s_i in the direction of X. We are attempting to minimize the maximal distance traveled by any sensor. To accomplish this let $x = (\max x_i - \min x_i)/2$, then the maximum distance any one sensor travels is minimized. In Figure 6, all the sensors will move to the line $X = 5$. It is straightforward to show that X must be somewhere inside the set x-coordinates of the sensors. As any two sensors move to meet one another, the total distance they travel is the distance between them. Thus, the sensors on either edge move the farthest, and half the distance between them minimizes the maximum distance any one sensor moves.

C. Optimal assignment of sensors to grid points

Assigning n sensors to n grid positions is related to the Assignment Problem, where we would seek a one-to-one matching between sensors and grid positions. The cost of each possible matching is the energy required to move the sensor to a grid point. A classic optimal solution is the Hungarian Method and can be computed in $O(n^3)$, where n is the number of sensors [44]. It produces an one-to-one assignment which minimizes the total energy expended.

However, in our context, we are looking to minimize the maximal amount of energy drain any one sensor incurs moving to a grid position. This will maximize the network lifetime. Let k be the maximal travel distance we allow any sensor to travel for moving to cover a grid position. We seek to find the smallest value of k such that each sensor expends at most k, yet all sensors can move to cover a grid position.

We can apply the Hungarian Method to compute k. Assume we have a function $HungarianK(S, G, k) \rightarrow Assignment, k'$. This function takes a list of sensor positions S, a list of grid positions G which are to be covered, a maximal value k, and computes a feasible assignment. If all the edges in the intermediate result do not exceed k then the assignment is returned otherwise the empty-set is returned. In addition k' represents the largest edge found in the assignment.

With this new function a centralized solution can be accomplished at a central node as follows. Sensors transmit positional information to a central node. The central node creates a sorted list of edge weights. Each sensor can be paired with every grid position resulting in a list of edge weights whose number is n^2. A binary search is conducted on the edge list and the midpoints of the edge weight list are fed as the parameter k to $HungarianK$. During the binary search, a feasible solution will result in $high$ being reduced otherwise low is increased. The running time for a solution will be $O(n^3 \log n)$, where $O(n^3)$ is for each running of Hungarian Method and $O(\log n)$ for the binary search. We provide two practical improvements for this experiment. We used the index associated with the generated k' instead of the current $high$ for the construction of the next midpoint as we know $k' <= k$. This algorithm is depicted in Algorithm 1.

Algorithm 1 Calculate the minimax sensor assignment.

Require: Lst is a sorted list of distances between sensors and grid points
Require: $Hungarian$ is a function that takes S Sensors, GP GridPoints, E Edges, and minimum inadmissible edge weight. Hungarian returns an assignment the max edge weight in the solution.
$minIdx \Leftarrow 0$
$best \Leftarrow \emptyset$
$k \Leftarrow Hungarian(S, GP, E, \infty)$
$maxIdx \Leftarrow Lst.index(k)$
while $minIdx < maxIdx$ **do**
  $k = Lst[(minIdx + maxIdx)/2]$
  $assign, k' \Leftarrow Hungarian(S, GP, E, k)$
  if $k' < k$ **then**
    $maxIdx = Lst.index(k')$
    $best = assign$
  else
    $minIdx = Lst.index(k') + 1$
  end if
end while
return $best$

D. Auction based assignment of sensors to grid points

The classic Hungarian method provides a centralized optimal solution to our assignment problem. To reduce computation cost, we present in this section a centralized approximate solution to the problem of assigning sensors to grid positions using *auction algorithms* [45]. Auction algorithms offer a reduced computational expense to the Hungarian method, while delivering close approximations to the optimal solution.

Auction algorithms arrive at an approximately optimal solution based on a set of agents bidding for the same resource, where each sensor labeled as i is bidding to be assigned to a particular grid position labeled as j. Each sensor associates a utility, which in our case is inversely proportional to a sensor's distance to a grid point. Let u_{ij} denote the utility of assigning sensor i to grid point j. Closer grid points have a higher utility to a sensor. Thus a sensor is willing to pay an increased price to compete for a grid point that maximizes its utility. The global known price for grid point j is denoted by p_j. The value $u_{ij} - p_j$ derives the benefit a sensor i achieves by being assigned to grid point j. Therefore, if the price at a particular grid point increases then a sensor will opt for a grid point with a lower utility and a higher benefit if it exists.

The algorithm starts by constructing an arbitrary assignment of sensors to grid positions. If each sensor derives the maximum benefit from its current assignment then the algorithm terminates. Otherwise, a sensor i seeks to be associated with j_i, which maximizes its benefit as shown in Eq 4.

Once the optimal candidate is identified, i raises p_j by the amount of $Increase$, which is the difference between the best benefit and the next best benefit. Now sensor i is associated with j_i and whoever j_i was associated with is associated with j. As j_i has the optimal benefit, $BestBen \geq NextBestBen$, therefore $Increase \geq 0$.

$$u_{ij_i} - p_{j_i} = \max_{j=1,\ldots,n}(u_{ij_i} - p_{j_i}) \tag{4}$$

$$p_{j_i} = p_{j_i} + Increase \tag{5}$$

$$Increase = BestBen - NextBestBen \tag{6}$$

$$BestBen = \max_{j}(u_{ij} - p_{j_i}) \tag{7}$$

$$NextBestBen = \max_{j \neq j_i}(u_{ij} - p_{j_i}) \tag{8}$$

It could turn out that $BestBen = NextBestBen$, i.e. a sensor derives the same benefit from being assigned to either of two separate grid positions. In that case, the bid increment would be zero. This could lead to repeated exchanges among a small group of sensors without increasing the price. This would result and an infinite loop and the system would fail to converge. To force convergence, we require that any bid must increase the price. To accomplish this in the algorithm, we replace Eq 6 with Eq 10 with a positive value ε. Further, we say that the system is in ε-equilibrium if each sensor is no more than ε off the maximum benefit possible. We accomplish this by substituting Eq 4 with Eq 9.

$$u_{ij_i} - p_{j_i} >= \max_{j=1,\ldots n}(u_{ij_i} - p_{j_i}) - \varepsilon \tag{9}$$

$$Increase = BestBen - NextBestBen + \varepsilon \tag{10}$$

We present an example to show how infinite loops might occur during the bidding process in phase two based on Eq 4 and Eq 6. Suppose that the sensors have limited movement so that S_0 can only bid for position G_0. Suppose that $S_1 - S_3$ are equidistant from G_1 and G_3. Then the sensors derive the same utility, denoted by U, from being assigned to G_1 or G_3. Suppose they can move all the way to G_2, but it would drain their energy reserves almost completely. Thus, G_2 offers a small non-zero utility to the sensors, which we denote by $.1 \times U$. Finally, suppose the initial mapping is $S_0 \to G_0, S_1 \to G_1, S_2 \to G_2, S_3 \to G_3$. To keep this example as simple as possible we remove S_0 and G_0 from our example. This configuration is depicted in Figure 7. The resulting infinite loop is presented in Table I. The top two benefits are identical for all the sensors and the initial price for all grid points is zero. Therefore, sensors S_1, S_2, and S_3 will wind up endlessly swapping assignments.

If we substitute in Eq 9 and Eq 10 we see the prices associated with being assigned to a grid point increasing. This is depicted in Table II. During each round of bidding the price for either G_1 or G_3 is increased by at least ε. Let $m\varepsilon$ be the first price that exceeds the utility U. When such a price is reached for either grid point, the benefit $(U - m\varepsilon) < 0$. At this point, the small utility $(.1 \times U)$, offered by G_2 to any sensor and its zero price will cause a sensor to prefer to be assigned to G_2. Once a sensor chooses G_2, we will have a feasible assignment and the algorithm can terminate. As demonstrated, if a feasible assignment exists then this algorithm is guaranteed to terminate in a finite number of steps (see [45] for details). Once complete, sensors move from their deployment positions to their assigned grid positions.

TABLE I.: An example of a situation where a naive auction can lead to bids with no price increase.

Round	Prices	Mappings	Bidder	Pref.	Inc.
1	(0,0,0)	$S_1 : G_1, S_2 : G_2, S_3 : G_3$	2	1	0
2	(0,0,0)	$S_1 : G_2, S_2 : G_1, S_3 : G_3$	1	1	0
3	(0,0,0)	$S_1 : G_1, S_2 : G_2, S_3 : G_3$	2	1	0

TABLE II.: Example of increasing bids for a grid position. Eventually the price will exceed the utility afforded by an assignment and a sensor bid for grid position G_2, allowing the algorithm to terminate.

Round	Prices	Mappings	Bidder	Pref.	Inc.
1	(0,0,0)	$S_1 : G_1, S_2 : G_2, S_3 : G_3$	2	1	ε
2	(ε,0,0)	$S_1 : G_2, S_2 : G_1, S_3 : G_3$	1	3	2ε
3	(ε,0,2ε)	$S_1 : G_3, S_2 : G_2, S_3 : G_2$	2	1	2ε
4	(3ε,0,2ε)	$S_1 : G_2, S_2 : G_1, S_3 : G_2$	1	2	2ε
...	...	...	...	...	...

E. Performance Evaluation

We use simulations to evaluate how the approximate solution might perform under our assumed deployment strategy and give some comparisons to the optimal solution. The average ocean depth is 3,790m [46]. Specialized submarines such as the Trieste can dive to 11,015m in the water [47]. Other mainstream submarines such as the Komsomolets series have dived to 1,300m [48]. Our simulations propose a sensor network that would be able to detect submarines up to 3,790m.

Recent papers have stated that commercial motes with magnetometers can detect submarines at distances of several hundred meters [28]. For our simulation, we chose a fixed detection range of 460m and if sensors are deployed every 640m in a grid pattern there will be no uncovered space. A column of six sensors with the first being deployed at a depth of 320m allows coverage of 3840m. We allow for our sensors to sink from between 320m to 3520m. The resulting cube's *depth* is 3200m. Modern submarines are capable of a submerged launch of anti-aircraft missiles [3]. This requires dispersal from a high flying plane to minimize the risk of being shot down during deployment. We allow sensors to drift away from the drop position with a radius up to 160m in any direction, so a cube's *width* is 320m. Finally, each sensor column is 640m apart. We assume that the aircraft uniformly drops sensors from the first column to the final column. The *length* of a cube ranges from 1920m to 55680m. Our simulations were run on an Intel Xeon ™ 3.20GHz CPU. The operating system was Red Hat Fedora Core 8. All software was written in Python.

F. Performance of the centralized optimal solution

First, we compare the running time of the traditional $Hungarian$ Method and our variant $HungarianK$ for different sized cubes. The former computes the minimum total weight solution and the latter computes the minimum maximum edge weight. Figure 8 depicts our results. The x-axis

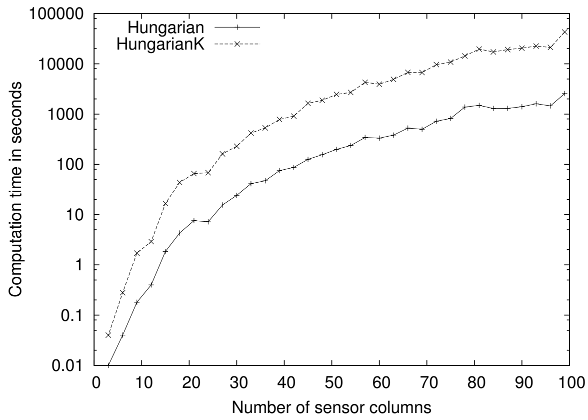

Figure 8: Comparison of running time between the $Hungarian$ and our $HungarianK$ variant.

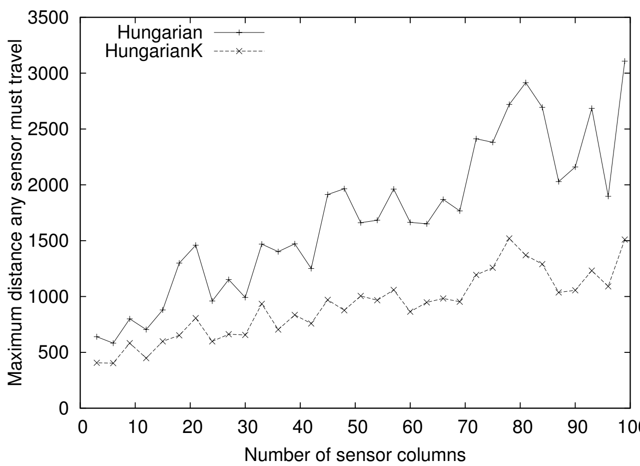

Figure 9: Comparing the $Hungarian$ Method with our $HungarianK$ variant to determine maximum moving distance.

represents the number of sensor columns being tested, which each sensor column being comprised of six sensors. The y-axis represents the total computation time required to achieve a result. This graphic shows that the computation time for computing HungarianK is within $O(n^3 \log n)$.

Figure 9 depicts the maximum movement any one sensor must travel to get to its optimal assignment. The x-axis is as before. The y-axis represents the maximum distance any one sensor must travel. As the network size increases, the $Hungarian$ Method requires more movement from sensors. Figures 8 and 9 show that there is a tradeoff to be made between computation time/energy and the total movement distance. This is an important consideration in energy constrained environments where movement consumes far more energy than computation.

G. Performance of the centralized approximate solution

To seed the initial assignment during each run, we associated each sensor to an arbitrary grid position for all but the last experiment. This mapping was chosen to allow for an unbiased estimate of how quickly the system would converge. After some initial analysis, we chose five different values of ε to show how the system behaved. The values chosen were 0.5, 1.0, 2.0, 5.0, and 10.0. Each experiment computed a result using a single value of ε.

In the last experiment we started the search for a solution with $\varepsilon = 10.0$. Once a feasible solution was determined, the solution was used as an initial assignment to find a solution with $\varepsilon = 5.0$. Any of previous assignments who satisfy the reduced maximal travel distance imposed can be used as part of the solution.

Those re-used assignments may not the be optimal, however as they are acceptable we can use them as a partial solution. This reduces the remaining assignments to be computed and thefore reduces the overall time complexity. This search strategy can be considered as a *simulated annealing* process [49]. We continued to reduce the ε value until $\varepsilon = 0.5$. Our simulations show how varying values of ε impact finding the optimal match between sensors and grid positions. We discuss different tradeoffs arising from varying the values of ε. The annealed-search based auction is denoted in the figures by the lines $\varepsilon = Var$.

In Figure 10, the x-axis represents the number of columns in the network with 6 sensors per column. The y-axis represents the average maximum number of bids any sensor makes to be associated with the grid position of its choosing. We can see that smaller values of ε result in a far higher number of bids. The annealed search results in generating numbers of bids similar to the smallest value of ε. For simplicity, this simulation assumes that all sensors communicate in a synchronous fashion with no communications loss.

In Figure 11, the x-axis is the same and the y-axis represents the total number of bids placed in the average case of the bidding process. As the network size increases, the total number of bids for smaller values of ε increase very quickly. If small values of ε are chosen in a real world deployment, the large volume of messages required to be transmitted and received would have a significant impact on battery lifetime.

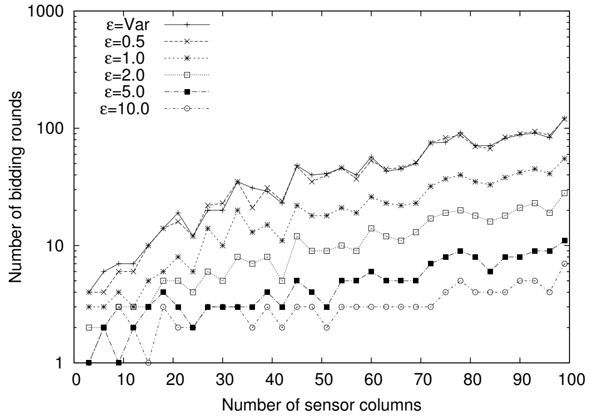

Figure 10: Size vs. number of rounds for varying values for ε.

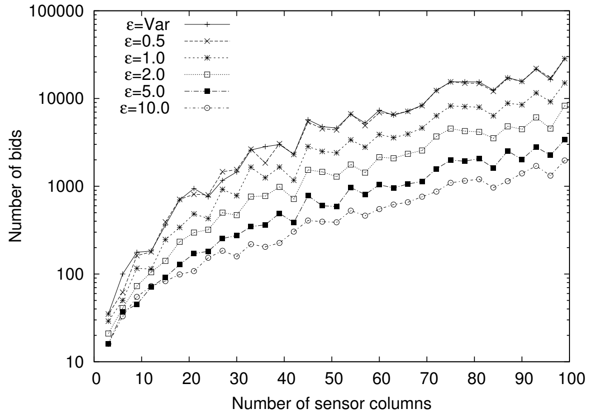

Figure 11: Network size vs. the average total number of bids broadcast for associating all sensors to grid positions given varying values for ε.

This would result in a tradeoff of sensor lifetime versus coverage.

In Figure 12, the x-axis is the same and the y-axis represents the maximum number of meters any sensor must travel. The smaller the value of ε, the shorter the travel distance for any sensor. In this case, the annealed search out performs all others. On average, the Hungarian method requires 1.34 times the distance of the annealed search and the modified Hungarian method with binary search, denoted by HungarianK, is .75 times better than the annealed search.

In Figure 13, the x-axis is the same and the y-axis represents the computation time of the approximate solution. For the annealed search, we see that the computation is more expensive when the number of sensor columns is smaller. As the number of sensor columns increase, the annealed search offers a noticeable computational improvement over smaller values of ε. In general, the computation of the Hungarian method is comparable to when $\varepsilon = .5$. The HungarianK function grows $O(\log n)$ faster making it difficult to simultaneously display all curves in a single graph.

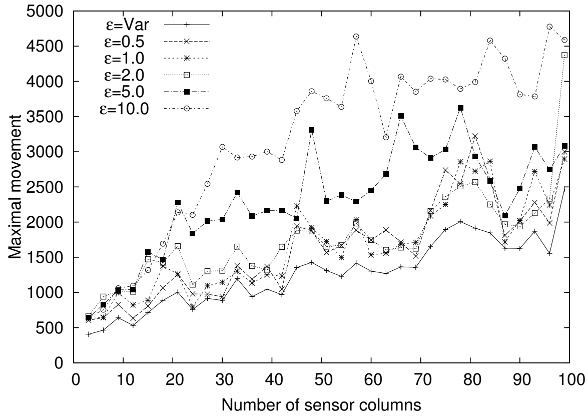

Figure 12: Network size vs. maximal movement required of individual sensor when varying ε.

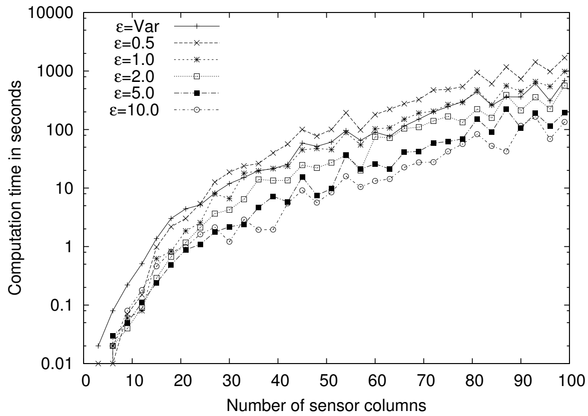

Figure 13: Network size vs. computation time for the different values of ε.

H. Discussion

In this Section, we started by using an optimal centralized solution to construct a barrier. The solution resulted in the minimum energy being expended for construction. The solution provides a construction with minimum energy consumption. To improve scalability, we showed how an auction algorithm can produce results similar to the optimal while reducing the computation time required to produce such a result. This work allows us to assert that auction algorithms can be used as a basis for mapping sensors to grid points.

V. Decentralized Solution

In the last section, we focused on centralized solutions. We assumed an omniscient node in control of the system. This node received positional information without considering communication time. The analysis was restricted to the computation time for finding solutions. First, the centralized solutions required finding a vertical plane that minimized the maximal travel distance required for all sensors to approach it. Once found, a vertical grid based barrier was constructed along the plane following the constraint that sensors minimized their maximal travel distance. We required that there be a one-to-one mapping between sensors and grid points. Further, we allowed sensors to travel an unlimited distance. This guaranteed that a feasible solution existed.

In this section, we develop a model that more realistically simulates the nodes in an underwater environment. We require that the system be decentralized and nodes must elect leaders who will control the local barrier construction. Barrier construction is completed in a stepwise fashion. The zone closest to the first drop zone completes and the next adjacent area begins. This continues until the last zone is reached and the system terminates its construction operation. During the construction, sensors compete to be assigned to grid positions in a zone by sending their positional information to a zone leader for an auction. If a sensor can move to cover a grid position in an adjacent zone, it's information can be forwarded to that zone leader as well. When an auction is complete, the zone leader transmits positional information for each winning bid. The auction attempts to minimize the maximum distance traveled by any one sensor. We relax the requirement that sensors first find the locally optimum line. If we don't there is the possibility that gaps between the locally optimal lines might exist and intruders knowledgeable of our barrier may be able to slip through undetected. Instead, we form the vertical barrier directly along the flight path of the plane.

Sensors have a limited range of movement. A limited travel range means that there may be situations where we can not assign available sensors to cover open grid points. Traditional auctions only guarantee termination when there is a feasible solution. To lessen this risk, we augmented the traditional algorithm. The changes described force the auction to terminate with or without a feasible solution. When a solution does not exist the simulation is terminated.

A. Experimental design

1) Definable inputs: In order to create an accurate simulation, we modeled the underwater communications channel as a broadcast medium with a bandwidth of 1KBps and a propagation delay of 1500m/s. We allowed sensors to move a maximum distance of 2200m. Currently, there are three major input features to our current system.

Our first parameter is packet loss. Packet loss affects the network's ability to produce short routes. Packet loss can lengthen the time required for zone leaders receive and send bid requests and responses. We expect higher packet loss to increase the computation time taken for the system to converge.

The next parameter is sensor density. By placing more than the required amount of sensors in the zone, we can limit the maximum travel distance. As more sensors mean some sensors will have a closer proximity to grid points. On the other hand since the transducer is broadcast based, too many sensors can clog the MAC layer increasing the communication time and thus convergence time.

The final parameter is algorithm initiation time. If we start immediately after discovering the minimum amount of sensors, we will quickly come to a solution. However, nodes on the edges who can not move to cover over zones maybe left out of the solution. This means that we require sensors capable of moving to cover an adjacent zone. This can create a situation where the next zone does not have enough sensors to cover it. Waiting longer allows sensors positional information in our multihop network to arrive at the zone controller and allows us have a better chance to include sensors incapable of moving anywhere else. Increasing the waiting time can increase the time taken to converge at a solution.

We simulated different uniform packet loss levels of 5%, 10%, and 15%. We believe that these values represent low, medium, and high loss levels given a fixed transmission distance and a fixed transmission power. We simulated Back Off timer settings of 30s, 60s, and 90s. When packet loss occurs, our transport layer retransmits unacknowledged packets after 15 seconds. This means that our Back Off intervals allow for at least 1, 3, and 5 retransmissions respectively. These intervals should allow even sensors on a zone's outer boundary the opportunity to be included in the bidding process. We simulated five different density levels for sensors, 1.0, 1.25, 1.5, 1.75 and 2.0. This spans the range from the minimum sensors required to twice the required sensors.

2) Measurable outputs: Finally, we analyze how these input factors impact barrier construction. The simulation run has two possible termination possibilities. If the Back Off timer expires and too few sensors have registered with the zone leader who is currently running an auction then the simulation exits with a failure status. If there is no assignment matching sensors to grid points within the sensor's maximal movement distance then the simulation terminates with a failure status. If each zone leader in turn conducts an auction and each auction results in a feasible assignment, the simulation terminates after the final zone leader conducts its auction.

We will analyze how varying these input features will impact our simulation's output variables: auction compute time, total time taken, maximum travel distance, and the number of MAC layer packets exchanged.

TABLE III.: Output variables of the simulator

Output variable	Description
MAC In	MAC layer packets received by a sensor
MAC Out	MAC layer packets transmitted by a sensor
MAC Error	MAC layer error encounter by a sensor
Max move dist	The maximum distance any sensor traveled
Finish time	The run time from the simulation wall clock
Auc.Time	The time the auctions alone took

3) Obtaining statistical significance: During this experiment, we performed sensitivity analysis to know how varying the input parameters would impact our output variables. Our output variables were the MAC layer statistics, maximum movement distance, finish time, and the total auction time. The method chosen of calculating the number of observations required to draw statistically significant conclusions is given by the Eq 11 and whose development can be found here [50]. In this equation d is the confidence interval centered at μ and $Z_{\alpha/2}$ is the two-tailed standardized normal statistic value.

$$n = \frac{(\sigma Z_{\alpha/2})^2}{(d)^2} \qquad (11)$$

We had two issues. We did not know the range of feasible output for any of the variables and we did not have an estimate of σ. Instead of setting an arbitrary value for our confidence interval, we decided to set it as $\frac{\sigma}{k}$. This approach can be found in [51]. This allows us to estimate the number of observations required without knowing the specific variance of each of the output variables.

We are specifically looking for an estimate within $\mu \pm \frac{\sigma}{k}$ where $k = 4$ with a probability of .95 so $Z_{\alpha/2} = 1.96$. We assumed that our output variables are normally distributed. Based on that assumption and the equations given, we need to collect 61 observations for an indication of statistical significance of our measured response parameters.

B. Positive experimental results

Each sensor can communicate 1500m and move 2200m. Therefore, each sensor can only interact with at most two zones during a run. We simulated constructing barriers with five zones. We believe even with the interactions that we can consider each zone's results a unique set of measurements. With 13 randomized runs, we were able to collect 65 unique observations. Based on statistical analysis we have developed, we state that this number of observations allows us to draw inferences on the interaction between our input parameters.

Factorial experimental design is a sound technique allowing us to extract the effect of the input parameters(factors) and their interactions on the output response economically. Without such a design, we must perform many more experiments varying only one factor at a time. With factorial designed, we can vary more parameters at a time yet still estimate each parameters individual and collective effects. The interested reader should consult [51], [52].

Table IV depicts our interactions. The leftmost column depicts the parameters that varied for the run of that row. The top row denotes the output variable in question. Each of the boxes represents the internal measures interaction of the input parameters on the output variable.

- **MAC In -** Varying packet loss and density together increase the number of packets received by a sensor. Packets are acknowledged by the end points. So one explanation is that losing a packet inside the mesh and the packet's retransmission becomes more problematic when these parameters increase. The only decrease is when all three are varied simultaneously.
- **MAC Out -** As with MAC In, the largest increase comes from varying packet loss and density together. The rational previously stated still applies and this result gives credence to those statements. The only decrease is when all three are varied simultaneously.
- **MAC Err -** Increasing the density increases the MAC error rate as more sensors sending messages increases network contention. As discussed earlier, this suggests that the MAC layer could be improved. Increasing packet loss decreases the errors, further suggesting contention is a problem.
- **Maximum Movement Distance -** Density is the single factor that decreases a sensor's maximum distance. However, when density is held constant and packet loss and delay are increased there is, also, a decrease in the travel distance. This suggests that there is an interaction between delay and loss that decreases the travel distance. We believe this is caused by network contention.
- **Finish Time -** Increasing the loss, density, or delay independently increase the finish time and that is expected. However, when combined there is an interaction that decreases the finish time. The interactions are more pronounced when density is a factor. This suggests that increasing the density can mitigate issues raised by delay or packet loss.
- **Total Auction Time -** The largest impact to the total auction time was when we varied the density level of the sensors. The existence of more sensors increases the time required to compute a feasible solution. Increasing the density and delay at the same time, also, increases the auction time. Our interpretation is that when we combine the two factors, the increased density creates paths for outliers to reach the zone leader resulting in less lost bids and higher delay assures those bids are considered during the auction.

TABLE IV.: The interactions among the variables

	MAC In	MAC Out	MAC Err	Max Mov Dist	Fin. Time	Tot. Auc Time
Loss,Delay,Density	-24.3	-1.55	-4.87	40.08	97.2	-0.26
Loss,Delay	27.2	1.96	2.6	-135.22	-0.3	-0.32
Delay, Density	31.62	1.4	5.22	89.22	-36.46	1.56
Loss, Density	85.06	2.9	9.64	40.28	-69.34	-0.58
Loss	15.4	2.2	-1.01	-15.27	104.12	-0.27
Delay	0.17	0.12	-0.47	-4.98	118.85	0.05
Density	78	0.41	10.82	-374.92	118.84	13.67

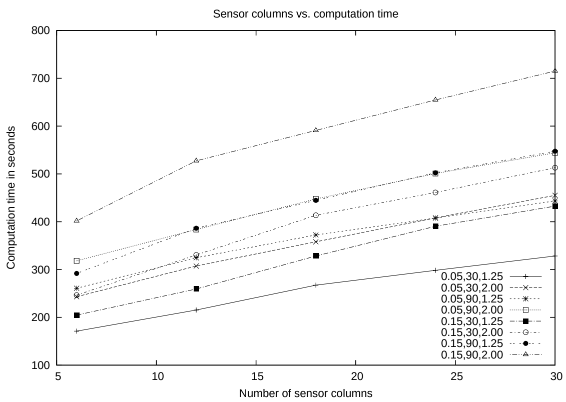

Figure 14: Sensor columns vs. computation time.

In Figure 14, the x-axis represents the number of sensor columns with each sensor column being comprised of six sensors. The y-axis represents the total computation time required to achieve a result. This figure takes into account all the communication costs associated with constructing our barrier. It depicts a linear growth rate as opposed to Figure 8.

Figure 15 depicts the maximum movement any sensor travels to get to its optimal assignment. The x-axis is as before. The y-axis represents the maximum distance any

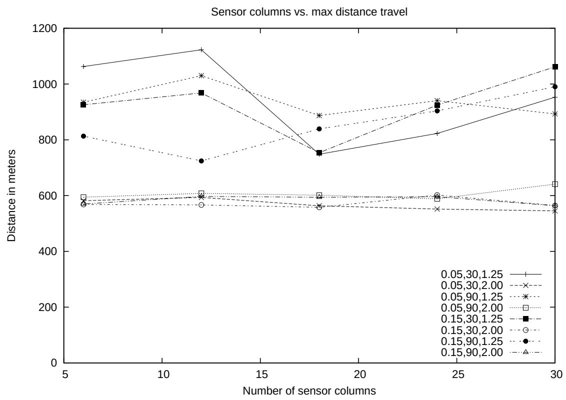

Figure 15: Sensor columns vs. maximal movement distance.

sensor travels. The figures depicts that maximal movement distance is not overly affected by increasing the number of zones.

C. Negative experimental results

During our experiments, we observed a number of failures that resulted in the condition of an inability to construct a barrier. Tables V, VI, and VII show the failures experienced by our simulator. One reason for a failure is that the auction did not produce a feasible solution. The available sensors were unable to move far enough to cover all required grid positions. In the tables, this type of failure is listed under the column with the headline *No assignment*. The other type of failure occurs when not enough sensors have registered with the zone leader at the expiration of the Back Off timer. These failures are listed under the column with the headline *Lack of sensors*.

Table V depicts the failures at a density of 1 sensor per grid point. This is the minimum number of sensors required to construct a one-to-one matching between sensors and grid points. The figures shows that there were few successful constructions at this density. There were no successful constructions with a Back Off timer of 30s and very few with higher settings. In most cases, the failures were the result of there being a lack of sensors available at the start of the auction.

Table VI depicts the failures at a density of 1.25 sensors per grid point. There were a high number of successful runs at this sensor density level. With the Back Off timer set at 60s, two of the three experiments did not encounter any failures. In most cases, the failures were the result of the auction failing to produce a feasible solution.

Table VII depicts the failures at a density greater than 1.25 sensors per grid point. Here the failure is the exception rather than the rule. In both, the failures depicted in this table there were not enough sensors available at the start of the auction.

D. Pilot study of longer runs

As previously stated, the bulk of the experiments were run with five zones resulting in five auctions. We ran a pilot experiment to show that the simulation would continue to provide linear growth in computation time and increasing the zones would not impact the maximal travel distance required by any one sensor. In the following experiment, we did an extended run of simulating the contruction of a barrier with 10 zones. The parameters we chose were 5% Packet Loss, 30s Back Off timer, and 1.5 sensors per grid point.

In Figure 16, the x-axis represents the number of sensor columns with each sensor column being comprised of six sensors. The y-axis represents the total computation time required to achieve a result. This figure takes into account all the communications cost associated with constructing our barrier. It depicts a linear growth rate even for longer runs.

Figure 17 depicts the maximum movement any sensor must travel to get to its optimal assignment. The x-axis is as before. The y-axis represents the maximum distance any sensor must travel. The figures depicts that maximal movement distance is not overly affected by increasing the number of zones.

Figure 18 offers an interesting insight. Once the auction has completed, the zone leader informs local sensors of their winning bids and then forwards the termination token onto the next zone leader. As the carrier sense operation takes .5 seconds, the 37th packet will not be transmitted till at least 18 seconds after generation. Thus, the strict FIFO(first in first out) queuing approach has direct impact on the total time taken to arrive at a solution. This does not take into account propagation delay, store and forward delay incurred by intermediate sensor nodes, or possible MAC back off delay due to current packets being rebroadcasted. All these effect the time to some level or another.

TABLE V.: Simulation failures with a density 1 sensor per grid point

Pkt Loss	Delay	Density	Success	Fail no Assignment	Fail lack of sensors
5%	30	1.00	0.0%	18.75%	81.25%
5%	60	1.00	6.7%	14.29%	85.71%
5%	90	1.00	0.0%	29.41%	70.59%
10%	30	1.00	0.0%	41.18%	58.82%
10%	60	1.00	5.9%	31.25%	68.75%
10%	90	1.00	0.0%	29.41%	70.59%
15%	30	1.00	0.0%	23.53%	76.47%
15%	60	1.00	0.0%	35.29%	64.71%
15%	90	1.00	12.5%	50.00%	50.00%

TABLE VI.: Simulation failures with a density 1.25 sensors per grid point

Pkt Loss	Delay	Density	Success	Fail no Assignment	Fail lack of sensors
5%	30	1.25	93.8%	100.00%	0.00%
5%	60	1.25	87.5%	100.00%	0.00%
5%	90	1.25	94.1%	100.00%	0.00%
10%	30	1.25	94.1%	100.00%	0.00%
10%	90	1.25	94.1%	100.00%	0.00%
15%	30	1.25	86.7%	50.00%	50.00%
15%	90	1.25	94.1%	0.00%	100.00%

TABLE VII.: Simulation failures with a density greater than 1.25 sensors per grid point

Pkt Loss	Delay	Density	Success	Fail no Assignment	Fail lack of sensors
5%	30	1.50	93.8%	0.00%	100.00%
5%	90	1.75	92.9%	0.00%	100.00%

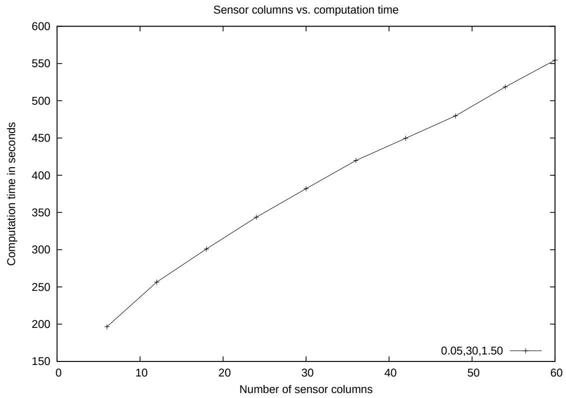

Figure 16: The average time taken to run a 10 zone simulation.

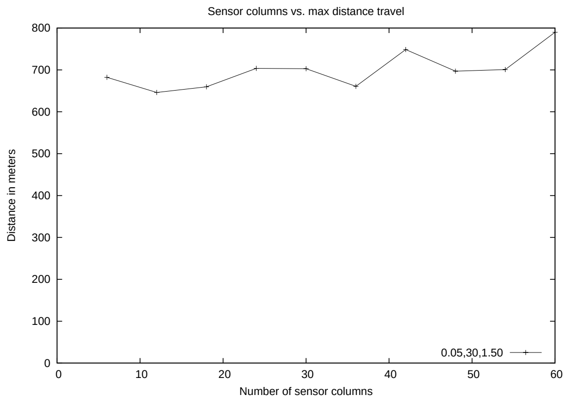

Figure 17: The maximal travel distance required for any sensor in a 10 zone run.

VI. Conclusions

We have provided a set of early results in barrier construction for detecting underwater intruders where sensors are distributed uniformly at random and modeled by a Poisson Point Process. We showed with such a distribution there are holes through which intruders can pass undtected. This motivated us to analyze contruction methods by moving mobile sensors to coverage positions. First, we examined the centralized optimal solution to the problem. Then we simulated a centralized approximation and a distributed approximation to the optimal solution, where we we also analyzed how sensor density and enviromental factors influence barrier construction.

Directly comparing centralized optimal and distributed approximation is problematic. The centralized approach assumes a leader node who knows the position of sensors without having that information forwarded to him. The distributed approach requires first electing a node that will be in charge, propagating that information, and finally collecting that position in a lossy error prone channel. The centralized approach assumed that every sensor could move to cover any grid point. In the distributed approach there was a limit on a sensor's maximal travel distance. The centralized approach computed one solution and the simulation terminated. The distributed approach computed a partial solution and then forwarded information over the lossy error prone channel to further

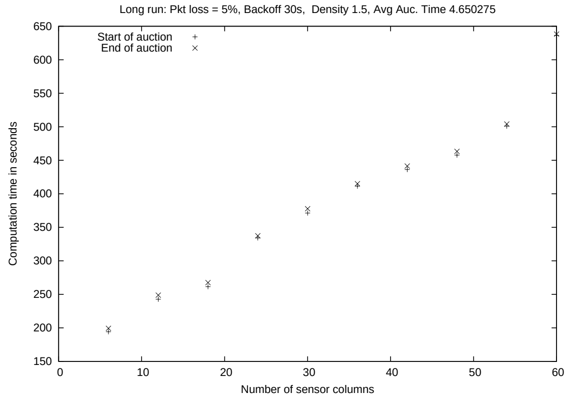

Figure 18: Each pair of points represent the start and finish times for the auction in the 10 zone simulation.

compute another partial solution and this continued until the total solution was computed.

Putting those differences aside there are conclusions we can draw. The centralized optimal approach is not scalable due to its unacceptable computational complexity and the energy used for computation. The decentralized approach offers a scalable approach. We have shown that the computational complexity grows at a linear rate. We have shown that the decentralized approach offers a travel distance that doesn't expand over longer ranges like the centralized approach. This strongly suggests that the dencentralized approach can be considered a replacement for the centralized approach.

We believe that the failures at a sensor density of 1.0 sensors per grid point suggest that with increased Back Off delay's we may have had more opportunities to successfully construct our barrier. A conclusion to draw is more research needs to be done on what the extremal values are needed have a high probability of constructing a barrier with low sensor densities.

Finally, our research suggests that even with simple MAC layer protocols barriers can be constructed using the auction mechanism in a timely fashion. However, our research suggests that more research specifically directed at MAC layer for such a system may lead to decreasing the time required to construct a barrier.

VII. Acknowledgements

This research is supported in part by the National Science Foundation under grants CCF-0830314, CNS-0953620, and CNS-1018303.

References

[1] (2008) Global submarine proliferation: Emerging trends and problems. [Online]. Available: http://www.nti.org/e_research/e3_74.html
[2] J. Forero, "Drug traffic beneath the waves," in *Washington Post*, February 6, 2008, p. A04.
[3] J. Kong, J. Kong, J.-h. Cui, D. Wu, and M. Gerla, "Building underwater ad-hoc networks and sensor networks for large scale real-time aquatic applications," in *Proc. IEEE Military Comm. Conf. MILCOM 2005*, J.-h. Cui, Ed., 2005.
[4] H. Gharavi, H. Gharavi, and S. Kumar, "Special issue on sensor networks and applications," *Proc. of the IEEE*, vol. 91, no. 8, pp. 1151–1153, 2003.

[5] S. Shakkottai, R. Srikant, and N. Shroff, "Unreliable sensor grids: Coverage, connectivity and diameter," in *Proc. IEEE Infocom*, 2003.
[6] C. Huang and Y. Tseng, "The coverage problem in a wireless sensor network," in *ACM Internat. Workshop on Wireless Sensor Networks and Appl. (WSNA)*, 2003, pp. 115–121.
[7] Z. Zhou, S. Das, and H. Gupta, "Connected k-coverage problem in sensor networks," in *Intl. Conf. on Computer Comm. and Networks (ICCCN)*, 2004.
[8] S. Kumar, T. Lai, and J. Balogh, "On k-coverage in a mostly sleeping sensor network," in *Proc. of ACM Mobicom*, 2004.
[9] X. Wang, G. Xing, Y. Zhang, C. Lu, R. Pless, and C. Gill, "Integrated coverage and connectivity configuration in wireless sensor networks," in *ACM Conf. on Embedded Networked Sensor Systems (SenSys'03)*, 2003.
[10] H. Zhang and J. Hou, "Maintaining sensing coverage and connectivity in sensor networks," in *invited paper in Internat. Workshop on Theoret. and Algorithmic Aspects of Sensor, Ad Hoc Wireless and Peer-to-Peer Networks*, 2004.
[11] D. Gage, "Command control for many-robot systems." in *Proc. of the Nineteenth Annual AUVS Tech. Sympos. (AUVS-92)*, 1992.
[12] B. Liu and D. Towsley, "A study on the coverage of large-scale sensor networks," in *The 1st IEEE Internat. Conf. on Mobile Ad-hoc and Sensor Systems*, 2004.
[13] S. Barr, B. Liu, and J. Wang, "Barriers with sensor networks using auction algorithms," in *ICCCN 2009 Proceedings*.
[14] B. Liu, P. Brass, O. Dousse, P. Nain, and D. Towsley, "Mobility improves coverage of sensor network," in *ACM MobiHoc, 20005*.
[15] S. Barr, B. Liu, and J. Wang, "Barrier coverage for underwater sensor networks," in *Milcom 2008 Proceedings*.
[16] A. Saipulla, C. Westphal, B. Liu, and J. Wang, "Barrier coverage of line-based deployed wireless sensor networks," in *IEEE Infocom, 2009*.
[17] A. Saipulla, B. Liu, G. Xing, X. Fu, and J. Wang, "Barrier coverage with sensors of limited mobility," in *to appear in the Eleventh ACM International Symposium on Mobile Ad Hoc Networking and Computing (Mobihoc), 2010*.
[18] A. Saipulla, B. Liu, and J. Wang, "Finding and mending barrier gaps in wireless sensor networks," in *IEEE Globecom, 2010*.
[19] S. Barr, B. Liu, and J. Wang, "Constructing underwater sensor based barriers using distributed auctions," in *Milcom 2009 Proceedings*.
[20] S. Meguerdichian, F. Koushanfar, M. Potkonjak, and M. B. Srivastava, "Coverage problems in wireless ad-hoc sensor networks," in *Proc. IEEE Infocom*, 2001, pp. 1380–1387.
[21] S. Meguerdichian, F. Koushanfar, G. Qu, and M. Potkonjak, "Exposure in wireless ad-hoc sensor networks," in *ACM Mobile Computing and Networking*, 2001, pp. 139–150.
[22] P.-J. W. X.-Y. Li and O. Frieder, "Coverage in wireless ad-hoc sensor networks," *IEEE Transactions on Computers*, vol. 52(6), pp. 753–763, June 2003.
[23] G. Veltri, Q. Huang, G. Qu, and M. Potkonjak, "Minimal and maximal exposure path algorithms for wireless embedded sensor networks," in *Proc. of ACM Sensys*, 2003.
[24] B. Liu, O. Dousse, J. Wang, and A. Saipulla, "Strong barrier coverage of wireless sensor networks," in *Proc. of Mobihoc '08*.
[25] T. H. L. S. Kumar and A. Arora, "Barrier coverage with wireless sensors," in *Proc. ACM Mobicom*, 2005.
[26] M. Cardei and J. Wu., *Handbook of Sensor Networks*. CRC Press, 2004, ch. Coverage in Wireless Sensor Networks.
[27] C. Gui and P. Mohapatra, "Power conservation and quality of surveillance in target tracking sensor networks," 2004.
[28] T. Nguyen, T. Nguyen, D. Nguyen, and D. Tran, "Stochastic binary sensor networks for noisy environments," in *Proc. First Internat. Conf. on Comm. and Electronics ICCE '06*, D. Nguyen, Ed., 2006, pp. 41–46.
[29] J. Lenz, H. Inc, and M. Minneapolis, "A review of magnetic sensors," pp. 973–989, 1990.
[30] (2008) 1- and 2-axis magnetic sensors. [Online]. Available: http://www.ssec.honeywell.com/magnetic/datasheets/hmc1001-2_1021-2.pdf
[31] J. Partan, J. Kurose, and B. N. Levine, "A survey of practical issues in underwater networks," *SIGMOBILE Mob. Comput. Commun. Rev.*, vol. 11, no. 4, pp. 23–33, 2007.
[32] I. F. Akyildiz, D. Pompili, and T. Melodia, "Underwater acoustic sensor networks: research challenges," *Ad Hoc Networks (Elsevier*, vol. 3, pp. 257–279, 2005.
[33] J. Proakis, E. Sozer, J. Rice, and M. Stojanovic, "Shallow water acoustic networks," *Communications Magazine, IEEE*, vol. 39, no. 11, pp. 114–119, 2001.
[34] J. Preisig, "Acoustic propagation considerations for underwater acoustic communications network development," 2007.
[35] A. F. Harris, III and M. Zorzi, "Modeling the underwater acoustic channel in ns2," pp. 1–8, 2007.
[36] A. Syed, W. Ye, and J. Heidemann, "T-lohi: A new class of mac protocols for underwater acoustic sensor networks," in *Proc. INFOCOM 2008. The 27th Conference on Computer Communications. IEEE*, 2008, pp. 231–235.
[37] M. Molins and M. Stojanovic, "Slotted fama: a mac protocol for underwater acoustic networks," in *Proc. OCEANS 2006 - Asia Pacific*, 2006, pp. 1–7.
[38] D. P. Bertsekas, "The auction algorithm: a distributed relaxation method for the assignment problem," *Ann. Oper. Res.*, vol. 14, no. 1-4, pp. 105–123, 1988.
[39] D. P. Bertsekas and D. A. Castanon, "The auction algorithm for the transportation problem," *Ann. Oper. Res.*, vol. 20, no. 1-4, pp. 67–96, 1989.
[40] J. Wein and S. Zenios, "Massively parallel auction algorithms for the assignment problem," in *Proc. rd Symposium on the Frontiers of Massively Parallel Computation*, 1990, pp. 90–99.
[41] A. Dutta and P. Tsiotras, "Asynchronous optimal mixed p2p satellite refueling strategies," 2005.
[42] B. P. Gerkey and M. J. Mataric, "Sold!: Market methods for multi-robot control," *IEEE Transactions on Robotics and Automation, Special Issue on Multi-robot Systems*, 2001.
[43] L. Kleinrock, *Queuing Systems*. John Wiley & Sons, 1976.
[44] H. Kühn, "The Hungarian Method for the Assignment Problem," *Naval Research Logistics Quarterly*, vol. 2, pp. 83–97, 1955.
[45] D. Bertsekas, *The auction algorithm: A distributed relaxation method for the assignment problem*. Springer, 1988, vol. 14, no. 1.
[46] (2008) Encyclopedia britannica online: Surface area, volume, and average depth of oceans and seas. [Online]. Available: http://www.britannica.com/eb/article-9116157/Surface-area-volume-and-average-depth-of-oceans-and-seas
[47] (2006) Seven miles down: The story of the bathyscaph trieste. [Online]. Available: http://bjsonline.com/watches/articles/0022_3.shtml
[48] (2008) Russia685. [Online]. Available: http://www.globalsecurity.org/military/world/russia/685-specs.htm
[49] C. D. G. S. Kirkpatrick and M. P. Vecchi, "Optimization by simulated annealing," *Science, New Series*, 1983.
[50] *Essentials of Simulation*. Prentice Hall, 1968.
[51] R. Shannon, *Systems Simulation: The art and science*, Prentice-Hall, Ed. Prentice-Hall, 1975.
[52] R. Langley, *Practical Statistics Simply Explained*, Dover, Ed. Dover, 1971.