This full text paper was peer reviewed at the direction of IEEE Communications Society subject matter experts for publication in the IEEE INFOCOM 2010 proceedings
This paper was presented as part of the main Technical Program at IEEE INFOCOM 2010.

Pressure Routing for Underwater Sensor Networks

Uichin Lee†, Paul Wang, Youngtae Noh, Luiz F. M. Vieira‡, Mario Gerla, Jun-Hong Cui§
†Bell Labs, Alcatel-Lucent UCLA ‡UFMG §University of Connecticut
uichin.lee@bell-labs.com, {p2wang,ytnoh,gerla}@cs.ucla.edu, ‡lfvieira@dcc.ufmg.br, §jcui@cse.uconn.edu

Abstract*—A SEA Swarm (Sensor Equipped Aquatic Swarm) is a sensor *cloud* that drifts with water currents and enables 4D (space and time) monitoring of local underwater events such as contaminants, marine life and intruders. The swarm is escorted at the surface by drifting sonobuoys that collect the data from underwater sensors via acoustic modems and report it in real-time via radio to a monitoring center. The goal of this study is to design an efficient anycast routing algorithm for reliable underwater sensor event reporting to any one of the surface sonobuoys. Major challenges are the ocean current and the limited resources (bandwidth and energy). In this paper, we address these challenges and propose HydroCast, a hydraulic pressure based anycast routing protocol that exploits the measured pressure levels to route data to surface buoys. The paper makes the following contributions: a novel opportunistic routing mechanism to select the subset of forwarders that maximizes greedy progress yet limiting co-channel interference; and an efficient underwater *dead end* recovery method that outperforms recently proposed approaches. The proposed routing protocols are validated via extensive simulations.

I. Introduction

Underwater sensor networks were recently proposed to support time-critical aquatic applications such as submarine tracking and harbor monitoring [1], [22]. Unlike traditional tethered sensors, a large number of underwater mobile sensor nodes are dropped to the venue of interest to form a SEA Swarm (Sensor Equipped Aquatic Swarm) that moves as a group with water current [24], [37]. Each sensor is equipped with a low bandwidth acoustic modem and with various sensors (e.g., Drogues [18]). Moreover, it can control its depth through a fish-like bladder apparatus and a pressure gauge. The swarm is escorted by sonobuoys at the sea surface that are equipped with both acoustic and radio (e.g., WiFi or Satellites) communications and GPS (See Figure 1).

There are several major advantages of SEA swarm architecture. First, mobile sensors provide 4D (space and time) monitoring, thus forming dynamic monitoring coverage. Second, the multitude of sensors (as in the SEA swarm) help provide extra control on redundancy and granularity. Third, floating sensors increase system re-configurability because they can control their depth; moreover they resurface once depleted of energy and can thus be recovered and reused.

In the SEA swarm architecture, each sensor monitors local underwater activities and reports time-critical data to any one of the sonobuoys using acoustic multi-hopping; then the data are delivered to a monitoring center using radio communications. The main focus of this paper is to design an efficient anycast routing protocol from a mobile sensor to any one of the sonobuoys on the sea level. However, this is challenging due to node mobility and limited resources (bandwidth and energy) of mobile sensors. An underwater

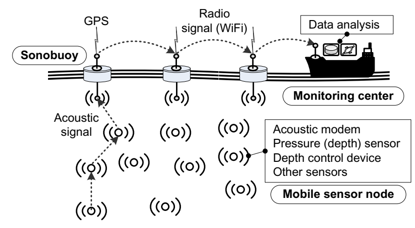

Fig. 1. SEA Swarm architecture

acoustic channel has low bandwidth and propagation latency five orders of magnitude higher than the radio channel [32]. Acoustic transmissions consume much more energy than terrestrial microwave communications. Such severe limitations in communication bandwidth coupled with high latency and limited energy make the network vulnerable to congestion due to packet collisions. Under these circumstances, minimizing the number of packet transmissions is important for at least two reasons: minimizing congestion and minimizing energy consumption.

Conventional proactive/reactive routing protocols (e.g., OLSR, AODV, etc.) rely on systematic flooding for route discovery and maintenance, potentially causing excessive energy consumption and collisions. In a SEA swarm scenario, general 3D geographic routing is preferable as it is stateless. However, geographic routing requires online, distributed localization of mobile sensors which is expensive and takes a long time to converge. Also, Durocher et al. [8] showed that efficient recovery from a local maximum may not always be feasible in 3D geographic routing, thus requiring an expensive exhaustive search such as 3D flooding and random walks [11].

Fortunately, our georouting problem is specialized in that it is anycast to any buoy on the surface. Thus, it suffices to route a packet upwards to lower depths. Given that the on-board hydraulic pressure gauge can accurately estimate depth (avg. error $<$ 1m [19]), we can use depth information for geographic anycast routing. Yan et al. [33] recently proposed a *greedy* method called Depth Based Routing (DBR) [33] where packet forwarding decisions are locally made based on the measured pressure level (or depth) at each node such that a packet is greedily forwarded to the node with lowest pressure among the neighbors. However, a forwarding node may find no other neighbors with a lower pressure level as it encounters a void region in the swarm. Like face routing in 2D [21], it must fall back to the recovery mode to route the packet around the void, but this was not addressed in [33]. Note that this hydraulic pressure based anycast routing is stateless and does not require expensive distributed localization [6]. In our scenario, the tagging of the sensed data with its location can be performed when the data come to surface. For example, a

978-1-4244-5837-0/10/$26.00 ©2010 IEEE

monitoring center can efficiently perform *off-line localization* using only local neighbor information collected from each node.

The main challenges of hydraulic pressure based routing are the unreliable acoustic channel and the presence of voids, thus requiring efficient greedy forwarding and dead end recovery methods. In this paper, we address these challenges and propose a generalized hydraulic pressure based anycast routing protocol called HydroCast. The following are the key contributions of the paper.

We consider wireless channel quality and take advantage of simultaneous packet receptions among one's neighbors to enable opportunistic forwarding by a subset of the neighbors that have received the packet correctly. To suppress hidden terminals, existing forwarding set selections use a heuristic to pick nodes in a geographic region facing the direction towards destination (in our case, upward direction) [14], [38], [33], [20]. We show that these approaches do not maximize the expected progress toward the destination, and in general, finding such a set is computationally hard. Thus, we propose a simple greedy heuristic that searches for a cluster of nodes with maximum progress and without hidden terminal problems, using only local topology information. Our simulation results validate that the proposed approach can find a set whose expected progress is very close to that of the optimal solution.

We then propose an efficient recovery method with delivery guarantee. The key idea is that a node can tell whether it is on the local maximum because only depth information is used for routing; i.e., a local maximum happens when there is no neighboring node with lower depth than the current depth. In our scheme, each local maximum node maintains a recovery route to a node whose depth is lower than itself. After one or several path segments that go through local maxima, a packet can be routed out of the void and can switch back to the greedy mode. Since any nodes located beneath the void area can potentially suffer from the void and opportunistic forwarding along the recovery path is feasible, our approach is more efficient than a random walk-based approach [11]. For efficient route discovery, we propose a route discovery method that implements hop-limited 2D flooding over the surface of void regions, which is a major improvement over simple 3D flooding.

II. Related Work

Geographic routing under channel fading: In geographic routing, a packet is greedily forwarded to the closest node to the destination in order to minimize the average hop count. Due to channel fading, however, the further the transmission range, the higher the attenuation, and the more the likelihood of packet loss. Researchers have tried to incorporate the associated cost such as the number of transmissions and energy consumption in geographic routing [23], [29]. For instance, Lee et al. [23] proposed a generalized link metric called Normalized Advance (NADV) where the amount of progress is normalized by its associated cost. However, these protocols did not consider the simultaneous packet receptions by one's neighbors and their ability of opportunistic packet forwarding by scheduling the set of nodes that received the packet correctly based on their distances (or associated costs) to the destination [2], [7], [38], [14].

A key design issue of opportunistic routing is the selection of a subset of neighbors that can make the best progress toward the destination, yet without the hidden terminal problem: i.e., when a higher priority node transmits a packet, other low priority nodes should be able to suppress forwarding to prevent redundant packet transmissions and collisions. Most opportunistic routing protocols (also called anypath routing) such as ExOR [2], Least Cost Opportunistic Routing (LCOR) [7] that do not use geographic information, require global topology and link quality information (like link state routing) to find a set of forwarding groups toward the destination; thus, they are more suitable for *static* wireless mesh or sensor networks. In practice, geographic routing can also benefit from opportunistic forwarding as in Geographic Random Forwarding (GeRaF) [38], Contention Based Forwarding (CBF) [14], and Focused Beam Routing (FBR) [20], though not optimal due to lack of global knowledge. In the literature, researchers typically used a geometric shape (e.g., a triangle/cone [14], [20]) that is *faced toward the destination* for forwarding set selection to prevent hidden terminal problems. The notion of the expected progress of opportunistic forwarding (in meters), called Expected Packet Advance (EPA) was recently established by Zeng et al. [34]. However, none of the previous works [38], [14], [34], [20] attempted to find a forwarding set with the maximum EPA and without the hidden terminal problem. In this paper, we show that finding such a set is a variant of the maximum clique problem, which is computationally hard, and thus propose a simple greedy heuristic method that well approximates the optimal solution.

Geographic routing recovery mode: The recovery mode in geographic routing can be classified as stateful or stateless. In 2D, face routing [21] is a widely used stateless (memoryless) strategy. The basic idea is to planarize a network graph using a simple local method and to forward a packet along one or possibly a sequence of adjacent faces, thus providing progress towards the destination node. To our surprise, for 3D networks it has been shown that there is no *local* memoryless routing algorithm that delivers messages deterministically as in 2D face routing [8]. Based on this observation, Flury et al. [11] proposed a randomized geographic routing using random walks. Nodes in the network are arranged in a virtual 3D grid coordinate using a localized algorithm where each grid point is a cluster of nodes in close proximity. A random walk is then performed on this virtual coordinate.

There are several stateful approaches proposed in the literature [26], [27], [35]. Greedy Distributed Spanning Tree Routing (GDSTR) [26] uses a spanning tree where each node has an associated convex hull that contains within it the locations of all its descendant nodes in the tree. A node exhaustively searches the tree for recovery by traversing sub-trees one by one. Liu et al. [27] proposed a backtracking method over a virtual coordinate system where a packet is routed towards one of the anchors (used to build the virtual coordinate system), hoping that it can switch back to the greedy mode on its way. Geo-LANMAR [35] inherits the group motion support

This full text paper was peer reviewed at the direction of IEEE Communications Society subject matter experts for publication in the IEEE INFOCOM 2010 proceedings
This paper was presented as part of the main Technical Program at IEEE INFOCOM 2010.

of Landmark Routing (LANMAR) that dynamically elects cluster-heads (landmark nodes). It circumvents a void in the network using the topology knowledge of landmark nodes as in [27]. In this paper, given the unique characteristic of our scenario where any nodes located beneath the void area can potentially suffer from the void, we consider keeping some state to reduce the recovery overhead (preventing expensive random walks to overcome the same void) and to exploit opportunistic packet forwarding along recovery paths.

III. Problem Statement

Instead of attacking generalized 3D geographic routing which requires an expensive distributed localization due to slow convergence speed, we propose to use 1D geographic anycast routing in a single (vertical) direction to the surface of the ocean using the depth information from a pressure sensor.[1] This routing simplification is justified by the fact that in our scenario communications are strictly *vertical*, from sensors to surface nodes. The need for global, distributed localization is relaxed via off-line localization at a monitoring center which uses local distance measurements (collected along with sensor data). Given this, the problem boils down to exploiting the opportunistic packet receptions under channel fading and developing an efficient recovery mechanism from a local maximum.

A. Forwarding set selection

Due to channel fading, the further the distance, the higher the signal attenuation and the more the likelihood of packet loss. We need to normalize the progress by its associated cost, which can be represented using Normalized Advance (NADV) [23]: for a given node, NADV to a neighbor node n that has the packet delivery probability p_n and the progress to the destination d_n^P (in meters) is given as $\frac{d_n^P}{1/p_n} = d_n^P \times p_n$. NADV can be extended to opportunistic forwarding as well. All neighboring nodes who receive the packet will access their priority based on how close they are to the destination; i.e., the closer to the destination, the higher the priority. A node will forward the packet when all the nodes with higher progress to the destination fail to send it. This can be easily scheduled by setting a back-off timer proportional to the distance to the destination. Since nodes can hear each other, those nodes with lower priorities will listen to the packet (either data or ACK packet) transmitted by a higher priority node and suppress their transmissions, thus excluding the possibility of collisions and redundant packet transmissions. Assume that source S has a set of k neighboring nodes Γ_k ordered based on their priorities as $n_1 > n_2 > \cdots > n_k$. The Expected Packet Advance (EPA) is simply the normalized sum of advancements made by this neighboring set [34]. The highest priority contributes on average $d_{n_1}^P p_{n_1}$ (=NADV). Since the next node can only contribute if the highest node fails, its

[1]Note that distributed localization typically requires many iterations, each of which takes a considerable amount of time because of large propagation delay and limited bandwidth underwater (often exacerbated by node mobility). We confirm this in the extended version of this paper and show that the overhead is closely related to the localization accuracy requirement [25].

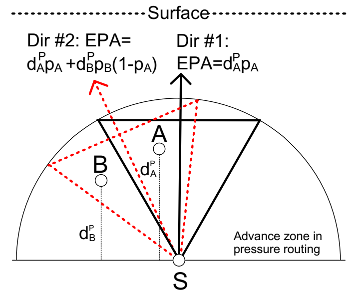

Fig. 2. Impact of direction in pressure routing (Dir #2 > Dir #1)

contribution is $d_{n_2}^P p_{n_2}(1 - p_{n_1})$. In general, the EPA is given as:

$$EPA(\Gamma_k) = \sum_{i=1}^{k} d_{n_i}^P p_{n_i} \prod_{j=0}^{i-1}(1 - p_{n_j}) \tag{1}$$

The above equation shows that as long as a node can make a positive advancement, we can include it to maximize the EPA, but we realize that including too many nodes may result in the hidden terminal problem that leads to redundant transmissions and packet collisions. Since the node degree is higher in 3D than 2D networks, 3D networks have higher probability of suffering from hidden terminals collisions than 2D networks [28]. Despite the fact that minimizing the number of transmissions in resource constrained sensor networks is one of the most important design criteria, none of the existing solutions [14], [38], [34], [20] consider the EPA metric and the hidden terminal problem simultaneously. However, the challenge is that finding such a forwarding set is a variant of the maximal clique problem that finds the largest clique in a graph, which is computationally hard; i.e., to be precise, find a clique with maximal EPA. Recall that a clique in a graph is an induced subgraph which is complete (i.e., every node can hear one another). As a simple heuristic, we could use a geometric volume, say a cone with the vertex on the transmitter and the base facing the direction to destination, which is a 3D extension of 2D methods reported in [14], [38], [20]. The problem is that they often fail to maximize EPA, as illustrated in Figure 2. In this paper we propose simple heuristics that search for a cluster of hidden-terminal-free nodes that maximizes EPA, using only local topology information. Also, we validate that our approach can find a set whose EPA is very close to that of the optimal solution.

B. Geographic routing recovery mode

After it was reported that for 3D networks there is no *local* memoryless routing algorithm that delivers messages deterministically [8], the state-of-the-art recovery scheme proposed in the field is a randomized geographic routing protocol using random walks [11]. However, we claim that this randomized approach may not be suitable for our scenario where nodes need to periodically send their local coordinate information (for off-line localization) and sensor data to the surface nodes. Since nodes vertically forward packets to the surface, any nodes located beneath the void area can potentially suffer from the void, and every packet originating that area has to perform an expensive random walk to overcome the same void. The overall amortized cost will be very high. Besides, it is not clear how we can exploit the opportunistic packet forwarding using random walks.

To these reasons, we opt for a stateful approach as in [16], [26], [27], [35]. The main departure from existing methods is that a node can easily tell whether it is on a local maximum by checking its neighbors' pressure level; i.e., it is on the local maximum if there is no neighboring node with a lower pressure level. If we assume that every local maximum node has a recovery route to a node whose depth is lower than itself (either another local maximum or a non-local maximum node where greedy forwarding can resume), the scheme successfully recovers from the voids. Namely, after one or several path segments that go through local maxima, the packet can be routed out of the void and can switch back to the greedy mode. Then, the key step is to efficiently find the recovery routes. The brute force approach is 3D flooding: i.e., local maximum nodes perform hop-limited 3D flooding until they find better escape nodes. In our scenario, we note that the route discovery overhead can be significantly reduced via *route discovery over the void floor surface* using 2D flooding. However, the challenge is to detect whether a node is on the void floor surface. In this paper, we present an efficient localized void surface floor detection algorithm and show that the aforementioned local *lower-depth-first* recovery method guarantees packet delivery.

IV. Forwarding Set Selection

A. Packet delivery probability estimation

We use the following underwater acoustic channel model to estimate delivery probability [32], [3]. The path loss over a distance d for a signal of frequency f due to large scale fading is given as $A(d, f) = d^k a(f)^d$ where k is the spreading factor and $a(f)$ is the absorption coefficient. The geometry of propagation is described using the spreading factor $(1 \leq k \leq 2)$; for a practical scenario, k is given as 1.5. The absorption coefficient $a(f)$ is described by the Thorp's formula [3].

The average Signal-to-Noise Ratio (SNR) over distance d is thus given as

$$\Gamma(d) = \frac{E_b/A(d,f)}{N_0} = \frac{E_b}{N_0 d^k a(f)^d} \quad (2)$$

Here, E_b and N_0 are constants that represent the average transmission energy per bit and noise power density in a non-fading additive white Gaussian noise (AWGN) channel. As in [32], [4], we use Rayleigh fading to model small scale fading where SNR has the following probability distribution

$$p_d(X) = \frac{1}{\Gamma(d)} e^{-\frac{X}{\Gamma(d)}} \quad (3)$$

The probability of error can be evaluated as

$$p_e(d) = \int_0^\infty p_e(X) p_d(X) dX \quad (4)$$

where $p_e(X)$ is the probability of error for an arbitrary modulation at a specific value of SNR X. In this paper, we use BPSK (Binary Phase Shift Keying) modulation that is widely used in the state-of-the-art acoustic modems [12]. In BPSK, each symbol carries a bit. In [30], the probability of bit error over distance d is given as

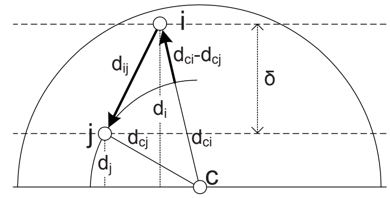

Fig. 3. Timer scheduling for prioritization

$$p_e(d) = \frac{1}{2}\left(1 - \sqrt{\frac{\Gamma(d)}{1+\Gamma(d)}}\right) \quad (5)$$

Thus, for any pair of nodes with distance d, the delivery probability of a packet with size m bits is simply given as

$$p(d, m) = (1 - p_e(d))^m \quad (6)$$

B. Packet forwarding prioritization

We use a distance based timer to prioritize packet forwarding where the distance denotes the progress toward the surface. When the current forwarder broadcasts a packet, nodes that receive the packet set the timer such that the greater the progress, the shorter the timer. Among those who receive the packet, the highest priority node becomes a next hop forwarder. The rest of lower priority nodes then suppress their packet transmissions after listening to the next hop forwarder's data or ACK packet.[2]

Unlike [38], [14], we define a linear timer function of a receiver x, which is customized for acoustic communications, as $f(d_x^P) = \alpha(R - d_x^P)$ where α is a constant, R is the maximum progress (i.e., transmission range), and d_x^P is progress of a receiver. Consider two nodes i and j with progress d_i^P and d_j^P respectively (see Figure 3). If we have $d_i^P > d_j^P$, we want to guarantee that $f(d_i^P) < f(d_j^P)$. Assuming that we use an ack for suppression, the timer function has to satisfy the following inequality: $t_{ci} + f(d_i^P) + t_{ij} + t_{ack} < t_{cj} + f(d_j^P)$ where t_{ab} is the propagation delay from node a to node b and t_{ack} is the transmission delay of an ACK packet. By plugging in $f(d^P)$, we have the following:

$$\alpha > \frac{t_{ci} - t_{cj} + t_{ij} + t_{ack}}{d_i^P - d_j^P} \quad (7)$$

The numerator is the sum of the propagation delay to travel $d_{ic} - d_{jc} + d_{ij}$ and the ack transmission delay, as shown in the figure (thick arrows). The progress difference between two nodes $(d_i^P - d_j^P)$ is critical; if it is too small, the constant α will be very large, thus resulting a very long delay. For a given candidate forwarding set, we can find the α by examining every pair using local topology information, which takes $O(n^2)$ steps where n is the number of neighbors. However, α may be too large. To handle this case, we have a system parameter that sets the maximum allowable delay per hop, denoted γ. In the following, we address how we choose a forwarding set that satisfies the delay constraint.

[2]Note that compared to a short ACK packet, a passive ACK (i.e., overhearing a data packet) is unreliable due to channel fading/collision.

C. Forwarding set selection methods

Nodes in the forwarding set must hear each other to prevent hidden terminal collisions. 3D networks have a higher probability of suffering from collisions than 2D networks, because for equal connectivity the node degree is higher in 3D than 2D networks. At the same time, we want to maximize progress (i.e., EPA). As discussed earlier, finding the optimal set is computationally hard and thus, we propose a simple clustering heuristic, inspired by the Multi-Point distribution Relay (MPR) selection in OLSR [17]. To this end, the current forwarder C requires the knowledge of 2-hop connectivity and neighboring nodes' pairwise distances. We assume that each node measures the pair-wise distance via Time of Arrival (ToA),, which is widely used in underwater networks [6], and the data are periodically reported to the surface for off-line localization. We take advantage of such periodic reports to obtain 2-hop neighbor information.

We assume that node C has computed the NADV of each neighbor as a forwarder *upwards* to surface. Like MPR selection where a node that covers the highest number of nodes is greedily picked, we use a simple greedy approach. The greedy clustering starts from the highest NADV neighbor, say S. Node S acquires all the other neighbors (of C) at distance $< \beta R$, where β is a constant ($\beta < 1$) and R is the acoustic range. In our design, we simply use $\beta = 1/2$ so that all the nodes clustered by S can hear each other. Then, if other neighbors are left, clustering proceeds starting from the highest value remaining neighbor and so on, until no nodes left. After this, each cluster is expanded by including all the additional nodes such that the distance between any two nodes in the cluster is smaller than R. This condition guarantees that nodes in the set can hear each other. We repeat this for all other clusters in turn and find the cluster with the highest EPA. Note that for a given set, we can find the minimum α value for priority scheduling, and this should be smaller than the maximum allowable delay per hop (γ). Thus, we remove one of the nodes with a lower NADV when detecting $\alpha > \gamma$ during the clustering process.

As an alternative, we can use a *cone shape* (3D counterpart of a Reuleaux triangle) to select a forwarding set. Unlike existing approaches [14], [38] that always orient a geometric contention shape along the line between source and destination, we need to search for the forwarding direction that maximizes the EPA. This requires local topology information – given n neighboring nodes with their depth information and pairwise distances, it is a realization of a graph with n nodes whose edges are weighted based on distance. We find the local topology using the Sweep algorithm that is known to work well for both sparse and dense networks [15]. In Sweep, we start from three known vertices and localize each neighboring node one at a time by computing all possible positions consistent with neighbor positions via a series of bilaterations until all vertices are localized. Using the local topology information, we discretize the 3D space into a unit degree of θ, thus generating total $2\pi^2/\theta^2$ directions over the hemisphere (advance zone). We then linearly scan each direction and calculate the EPA to find the direction with maximum EPA.

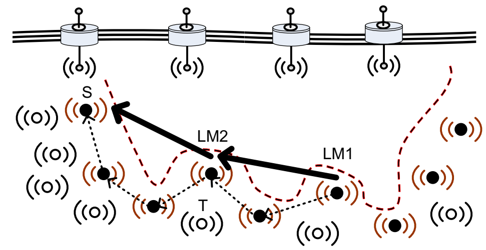

Fig. 4. Recovery mode

After forwarding set selection, we need to include the chosen forwarding set in the data packet. To reduce the overhead, we use a Bloom filter, a space efficient membership checking data structure. The membership checking is probabilistic and false positives are possible, but we can bound the probability of false positive by properly adjusting the filter size. In a practical scenario, the set size will be smaller than 15 (in the hemisphere advance zone). Fan et al. [10] showed that a filter size of 150 bits (19B) to represent 15 items has a false positive rate smaller than 1%. We can also include sender's pressure level and max/min angle information to filter out quite a few of neighboring nodes that are not in the forwarding set. Furthermore, noting that there could be many other packets that have to travel through a certain node, and topology slowly changes over time, we may only need to include the set information in the data packet whenever there is a sufficient change. Thus, the amortized overhead could be much smaller.

V. Recovery Mode

We present a local lower-depth-first recovery method that guarantees the delivery and provide an efficient recovery route discovery method using 2D surface flooding, instead of expensive 3D flooding. Note that opportunistic forwarding over a recovery path is illustrated in the extended version of this paper [25].

A. Local lower-depth-first recovery

Unlike traditional geographic routing where the local maximum is determined by the location of a destination node, in our scenario, each node can easily tell whether it is on the local maximum by checking its neighbors; i.e., a node is on the local maximum if there is no neighboring node with lower pressure level. Having said that, we propose a lower-depth-first recovery method as follows. Every local maximum node searches for a node whose depth is lower than its current depth and explicitly maintains a path to the node (via some route discovery method). This node could be another local maximum where there is a new recovery path, or the point where the greedy forwarding can be resumed. Whenever a packet hits a local maximum, it is re-routed along the recovery path either safely to a node that can resume greedy forwarding or to a new local maximum. In Figure 4, for instance, we have two local maxima, namely $LM1$ and $LM2$. $LM1$ maintains a path to $LM2$ which has a path to node S. A packet can be routed from $LM1$ to $LM2$ to S. It then can be switched back to the greedy mode and can be delivered to a node on the ocean surface. In practice, we can recover from the local maximum after a few iterations.

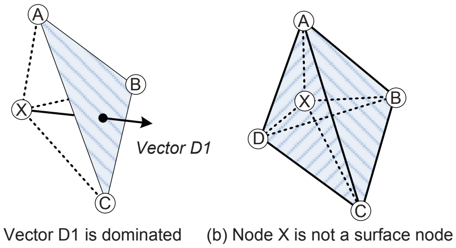

Fig. 5. Illustration of domination and non-surface node

The following theorem proves the delivery guarantee and loop-free property of lower-depth-first routing.

Theorem 1: Local lower-depth-first routing is loop-free and guarantees the packet delivery.

Proof: Consider a local maximum graph $G = (V, E)$. A vertex $v \in V$ in the graph represents a local maximum node, and two vertices are connected if there is a recovery path. There is also a sink vertex that can reach the surface. If each vertex (local maximum) can reach the surface directly without visiting into other local maxima, it is connected to the sink. Assume that a packet arrives at a local maximum, say vertex v_i. If v_i is connected to the sink, the packet is safely delivered. Otherwise, it will be re-routed to another local maximum (say v_j) whose depth is lower than the current depth by definition; i.e., $D(v_i) > D(v_j)$ where $D(v_k)$ returns the depth of node v_k. Since the distance to the surface decreases in each step, a packet can be delivered after a finite number of steps that is strictly less than the total number of local maxima. This monotonic behavior also guarantees that there is no loop. ■

B. 2D void floor surface flooding for recovery path search

Now the important step is to find the recovery route. The brute force approach is 3D flooding; i.e., nodes at the local maxima perform *expensive* hop-limited 3D flooding to discover escape nodes where greedy mode can resume or there are recovery paths to better escape nodes. This brute force approach is not deemed suitable because the appropriate *scope* of the limited 3D flooding is difficult to estimate and the 3D flood can degenerate to involve all nodes in the sensor mesh. To improve efficiency, we use *2D flooding on the void floor surface*. This flood involves a much more manageable set of nodes. Figure 4 illustrates the approach in 2D. Nodes on the envelope (or surface) can become aware of their void floor surface status using local connectivity information and thus forward the packet. Nodes that are dominated by surface neighbors are not on the surface and refrain from forwarding. For instance, node S does not have any nodes on its right and is a surface node. Node T is surrounded by its neighboring nodes and it is not on the surface. We now formally define *domination* and a *void surface node* in 3D environments.

Definition 1: For a given node, a random vector emanating from the node is dominated if and only if there is a dominating triangle formed by the node's neighbors that intersects with the vector. A node is on the surface if and only if there exists a vector that is not dominated (no dominating triangle for the vector).

Consider Figure 5(a). Random vector $D1$ emanating from node X is dominated because it intersects with the triangle ABC. Any random vector pointing inside the tetrahedron $XABC$ is dominated by the triangle. In Figure 5(b), node X is completely surrounded by a set of tetrahedra that dominates every possible direction. Thus, node X is a non-surface node.

Surface node detection can be formally described as follows. Consider the point set P in 3D Euclidean space where the set is composed of node X and its neighbors. Any pair of points in the set is connected if their distance is less than or equal to the transmission range. The point set centered at node X is now normalized (as a unit vector), thus lying on the surface of a unit ball. The connectivity among points in the normalized point set does not change. Given that we have a normalized point set and connectivity information among set members, the surface node detection is to decompose the point set into a set of non-overlapping tetrahedra that exhaustively cover the unit ball. It is known that the length constraints (due to the communication range) make these kinds of tetrahedralization problems intractable [31], [13]. If there is no length constraint, the problem becomes a decomposition of the convex hull of P into non-overlapping tetrahedra, which can be solved in $O(n \log n)$ where n is the point set size [9].

In this paper, we propose a simple Monte Carlo approximation method: pick k random directions and check whether there is a dominating triangle for each direction. The number of directions k should be large enough to correctly identify the surface node. Otherwise, void floor surface detection may fail, generating a false negative – a surface node is declared as a non-surface node. We detail our approximation method as follows. We first generate a set R of k random vectors. There are $O(n^3)$ triangles that can be formed by node X's neighbors. For each triangle, we repeat the following procedure: check all vectors in the set R to know whether they are dominated by the triangle and remove the dominated vectors from the set R. If R becomes empty, we declare that node X is not on the void floor surface. Otherwise, our algorithm declares that node X is on the void floor surface. Thus, the worst case complexity is given as $\Theta(kn^3)$. Note that the detection algorithm is localized, requiring only 1-hop topology which can be constructed using only periodic beacons. Thus, it does not cause any additional packet exchanges. Moreover, the processing overhead is minimal because it is triggered only when nodes detect that local network topology has sufficiently changed.

The accuracy of this method can be analyzed as follows. Let X be a node that is on the void floor surface. The volume of a sphere centered at node X is given as $\frac{3}{4}\pi R^3$ where R is the transmission range. Assuming that the volume of a void area that intersects with the sphere is x, a random vector hits the void area with probability $p = \frac{4x}{3\pi R^3}$. A false negative happens when all k random vectors miss the void. Thus, the probability of a false negative is given as $(1 - p)^k$, thus exponentially decreasing with k. In practice, the volume size of a void is large enough, and we can achieve high accuracy with small k. For instance, when the intersecting void volume is one fifth of the sphere ($p = 1/5$) and $k = 20$, the probability of a false negative is about 1%.

So far we assume that a void area always causes the local maximum. In our pressure routing, interestingly not every void

This full text paper was peer reviewed at the direction of IEEE Communications Society subject matter experts for publication in the IEEE INFOCOM 2010 proceedings
This paper was presented as part of the main Technical Program at IEEE INFOCOM 2010.

area causes the local maximum. Such void areas are usually located inside the swarm (à la the air bubbles in bread dough), and greedy forwarding can successfully bypass the void areas (say an egg-shape area). We call this kind of void a *bubble*. The size of a bubble is closely related to node density; as node density increases, there will be fewer bubbles whose sizes are also diminishing. Figure 5(b) shows that we need at least four nodes in order not to rule out a surface node. Given that a well connected 3D network requires each node to have around 30 neighbors (in 2D, 15 neighbors) [28], these bubbles will likely happen particularly when node density is very low. In practice, those nodes on the bubble surface will not cause a problem. The special case that needs our attention happens when a bubble *contacts* the real void floor surface. Under this circumstance, those nodes will receive route discovery packets and also participate in the flooding process, causing redundant packet transmissions. To prevent this, nodes should be able to tell whether the void area is a bubble or not. However, this is an expensive process, requiring more than 2-hop information. It is part of our future work to investigate how this case can be efficiently handled.

VI. Simulations

In this section, we evaluate the proposed approaches via simulations using QualNet. First, we investigate the forwarding set selection to answer: how important is the hidden terminal problem?; and how good are our forwarding set selection heuristics? Second, we evaluate the recovery mode to answer: how often is a packet trapped in a local maximum for varying node density?; and how effective is our proposed void surface detection scheme? Finally, we compare the performance of various depth-based routing strategies (e.g., different forward set selection methods and recovery modes).

A. Simulation setup

For acoustic communications, the channel model in Section IV-A is implemented in the physical layer of QualNet. We generate different channel fading conditions in the simulations by adjusting the transmission power in dB re μ Pa and SNR threshold [32].[3] Unless otherwise mentioned, the transmission power is set to 105 dB re μ Pa. We use a transmission range of 250m, and the data rate is set to be 50Kbps as in [36]. We use the CSMA MAC protocol. In CSMA, when the channel is busy, a node waits a back-off period and senses carrier again. Every packet transmission is MAC layer broadcasting. For reliability, we implement ARQ at the routing layer as follows. After packet reception, the receiver sends back a short ACK packet. If the sender fails to hear an ACK packet, a data packet is retransmitted; and the packet will be dropped after five retransmissions.

We randomly deploy varying numbers of nodes ranging 100 to 450 in 3D region of size 1000m $\times$ 1000m $\times$ 1000m. Due to mobility, nodes would move beyond this region. They move according to an extended 3D version of the Meandering Current Mobility (MCM) Model [5], an underwater mobility model that considers the effect of meandering sub-surface currents (or jet streams) and vortices. In the model, we set the main jet speed to 0.3m/s.

[3]The signal intensity is measured in dB re μPa of the power flux $[Wm^{-2}]$ delivered into the water by a source.

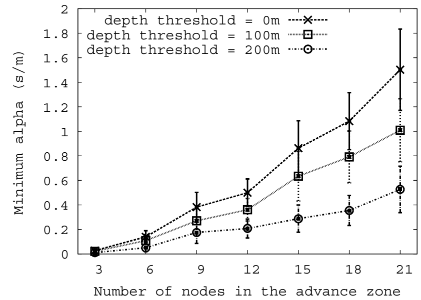

Fig. 6. Minimum alpha value

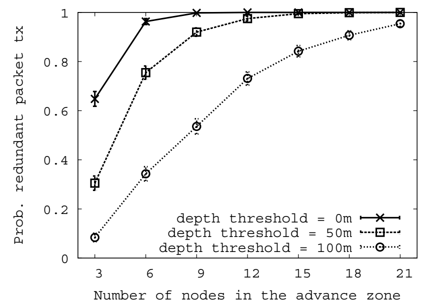

Fig. 7. Probability of a redundant packet transmission

Each node measures the distance to its neighbors every 30 seconds (with random jitters to prevent synchronization) and broadcast the measured information to its one hop neighbors. Every 60 seconds, each node reports the sensed data and distance measurements up to the surface. Note that a node in the main jet stream will have moved 20m in 60s. With a range of 250m, we expect the 60s refresh rate to be adequate to track topology changes for off-site localization. The size of a packet is a function of the number of neighbors, and the average packet size is less than 200B in our simulations. We measure delivery ratio, delay and overhead. The delivery ratio of a source is the fraction of the packets delivered; the delay is the time for a packet to reach any of the sink nodes on the surface; and the overhead is measured in terms of the total number of packet transmissions. In our simulations, each run lasts 3600s. Unless otherwise specified, we report the average value of 50 runs with the 95% confidence interval.

B. Simulation results

In Section IV, we show that the forwarding set selection and its prioritization must be properly done to prevent the hidden terminal problem. Otherwise, there will be redundant transmissions and collisions. To show the impact, we evaluate a simple forwarding set selection method proposed in DBR (Depth Based Routing) [33]. Recall that DBR is the first underwater routing scheme to exploit pressure (and thus depth) awareness at each node for routing packets to surface. It implements a basic greedy forwarding design with an opportunistic forwarding flavor. All the nodes higher than the current forwarder by more than a depth threshold (h) act as opportunistic forwarders. Moreover, DBR uses a fixed holding timer at each hop. We randomly deploy varying numbers of nodes in

This full text paper was peer reviewed at the direction of IEEE Communications Society subject matter experts for publication in the IEEE INFOCOM 2010 proceedings
This paper was presented as part of the main Technical Program at IEEE INFOCOM 2010.

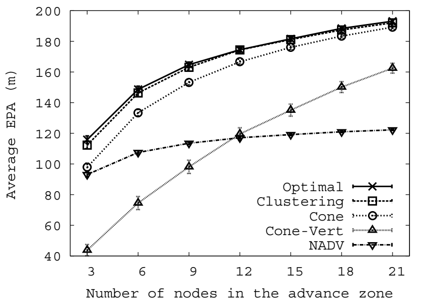

Fig. 8. EPAs of different forwarding set selection schemes

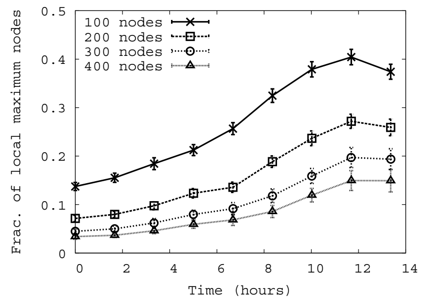

Fig. 9. Fraction of local maximum nodes over time

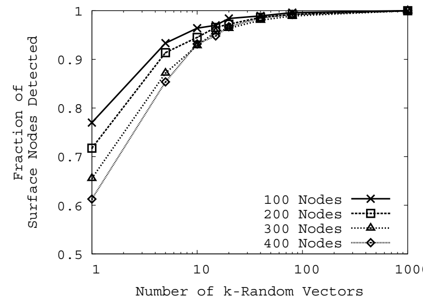

Fig. 10. Void floor surface detection using a Monte Carlo method

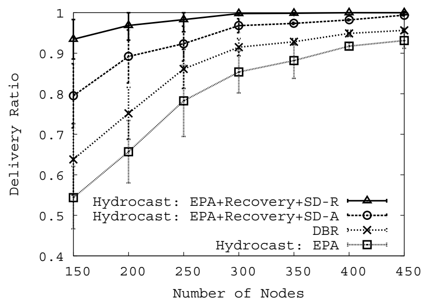

Fig. 11. Packet delivery ratio

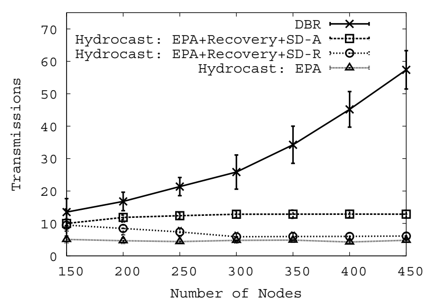

Fig. 12. Number of transmissions for delivery

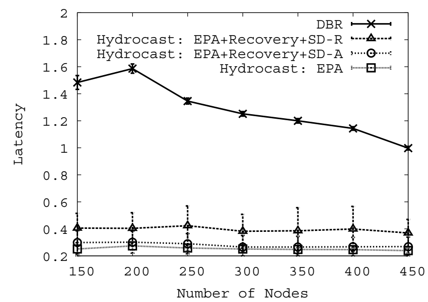

Fig. 13. Average end-to-end delay

the hemisphere (3-21 nodes). For a given configuration, we calculate the minimum α value using Equation 7 using three different minimum depth thresholds (h=0m, 100m, 200m). We plot the average α value of 1000 random configurations with 95% confidence interval. Figure 6 clearly shows that as density increases, it is more likely that two nodes are in close proximity (or have low depth difference), and thus, the average minimum alpha value significantly increases. For example, the alpha value of 1 can result in the maximum delay of 250s in our scenario. In Figure 7, we plot the probability of redundant packet transmissions caused by the hidden terminal problem. We see that the hidden terminal problem persists even with a high depth difference value. For instance, a 10 node scenario has more than a 60% chance of redundant packet transmissions with depth difference of 100m. Therefore, it is mandatory to suppress such redundant transmissions.

We now evaluate how good our forwarding set selection algorithm is. In Figure 8, we plot the progress (EPA) of the different forwarding set selection schemes: optimal, cone-based, clustering, and simple NADV. In the optimal scheme, we perform exhaustive search on the neighbor set to find the maximum EPA. NADV denotes the case where we only choose the node whose NADV is the largest (i.e., a single node in the forwarding set). Cone-Vert only considers the vertical direction as in [14], [38]. The figure shows that our clustering method is very close to the optimal solution, outperforming the cone based approaches. The results also show that the more the number of nodes, the higher the EPA (as expected).

We measure the fraction of local maximum nodes over time under MCM mobility (half a day). Since this is closely related to the node density, we vary the number of nodes ranging from 100 to 400 nodes. For a given configuration, we sample the number of local maximum nodes every 1.7 hours. Figure 9 shows the results. When the node density is low, the fraction of local maximum nodes is high. As time passes, the fraction of local maximum nodes increases. This is due to the fact that nodes tend to disperse over the simulated area (beyond the original 1000x1000x1000 cube) due to ocean currents (i.e., jet streams and vortices).

We analyze the accuracy of our proposed void floor surface detection method. We measure the fraction of the surface nodes detected by varying the number of nodes in the network (100-400) and the number of random vectors ($k = 1 - 1000$). For the sake of clarity, we divide the number of detected surface nodes by that of the case with $k = 10,000$. Figure 10 shows the results. The lower the density, the higher the detection probability because the area that intersects with the void is larger (i.e., larger p). As the number of random vectors increases, the detection probability approaches to 1. The figure shows that the detection probability is over 95% with $k = 20$.

Finally, we compare the performance of HydroCast with DBR under different settings. Recall that DBR implements a basic greedy forwarding design with an opportunistic forwarding flavor and uses a fixed holding timer at each hop. HydroCast uses a more elaborate opportunistic forwarding strategy and supports also recovery from voids. To show the benefit of our 2D surface flooding (denoted as SD-R), we also implement a simple angle-based selection heuristic: i.e., when a node X broadcasts a route discovery packet, any neighboring node A whose adjacent angle formed by the X-axis and XA is less than 60 degrees participates in the flooding (denoted as SD-A). Figure 11 shows the packet delivery ratio. When node density is low, DBR has higher delivery ratio than HydroCast without recovery. Unlike HydroCast, DBR does not suppress redundant packet transmissions and thus, it is likely to deliver packets on multiple paths, improving reliability. The same figure also reports the plot for HydroCast with forwarding set selection and recovery. We note that recovery support from voids significantly improves the reliability of HydroCast and puts it above DBR. Accurate surface detection helps us to achieve better PDR, because angle-based selection may not include some of the surface nodes, failing to find the recovery path (especially, when density is low). In Figure 12, we plot the average number of packet transmissions to deliver a data packet, including the recovery process. Due to redundant packet transmissions and multi-path packet delivery, DBR results in significantly more number of transmissions than other schemes. Interestingly, the impact of recovery reduces as the density increases. This is because there will be fewer voids and fewer hops to switch back to the greedy mode, and more number of nodes is involved in packet forwarding; thus, the amortized recovery cost decreases. In the case of angle-based selection, the overall overhead remains the same, because there will be more redundant packet transmissions as density increases (fewer voids, but much higher costs). Finally, Figure 13 shows that HydroCast has lower end-to-end delay than DBR thanks to HydroCast's adaptive timer setting at each hop. As density decreases, the average delay in HydroCast slightly increases because of the increased frequency of voids requiring recovery and thus longer paths.

VII. Conclusion

We investigated hydraulic pressure based anycast routing that allows reporting time-critical sensor data to the sonobuoys on the sea level using acoustic multi-hopping. Since acoustic transmissions are power hungry, our goal was to minimize the number of packet transmissions in underwater sensor deployments challenged by ocean currents, unreliable acoustic channels and voids. In this paper, we proposed HydroCast, a hydraulic pressure based anycast routing protocol with the salient features: novel opportunistic routing mechanisms to select the subset of forwarders that maximizes greedy progress, yet limits co-channel interference; and an efficient underwater *dead end* recovery method that outperforms recently proposed approaches (e.g., random walk, 3D flooding, etc). Simulation results confirmed that the proposed protocols can effectively handle the challenges.

References

[1] I. F. Akyildiz, D. Pompili, and T. Melodia. Underwater Acoustic Sensor Networks: Research Challenges. *Elsevier Ad Hoc Networks*, 3(3):257–279, 2005.

[2] S. Biswas and R. Morris. Opportunistic Routing in Multi-Hop Wireless Networks. In *SIGCOMM'05*, Aug. 2005.

[3] L. M. Brekhovskikh and Y. Lysanov. *Fundamentals of Ocean Acoustics, Third Edition*. Springer, 2003.

[4] C. Carbonelli and U. Mitra. Cooperative Multihop Communication for Underwater Acoustic Networks. In *WUWNet'06*, Sept. 2006.

[5] A. Caruso, F. Paparella, L. F. M. Vieira, M. Erol, and M. Gerla. The Meandering Current Model and its Application to Underwater Sensor Networks. In *INFOCOM'08*, Apr. 2008.

[6] V. Chandrasekhar, Y. S. Choo, and H. V. Ee. Localization in Underwater Sensor Networks – Survey and Challenges. In *WUWNet'06*, Sep. 2006.

[7] H. Dubois-Ferriere, M. Grossglauser, and M. Vetterli. Least-Cost Opportunistic Routing. In *Allerton'07*, Sept. 2007.

[8] S. Durocher, D. Kirkpatrick, and L. Narayanan. On Routing with Guaranteed Delivery in Three-dimensional Ad Hoc Wireless Networks. In *ICDCN'08*, Jan. 2008.

[9] H. Edelsbrunner, F. P. Preparata, and D. B. West. Tetrahedrizing Point Sets in Three Dimensions. *Symbolic and Algebraic Computation*, 1990.

[10] L. Fan, P. Cao, and J. Almeida. Summary Cache: A Scalable Wide-Area Web Cache Sharing Protocol. In *SIGCOMM'98*, Aug.-Sept. 1998.

[11] R. Flury and R. Wattenhofer. Randomized 3D Geographic Routing. In *INFOCOM'08*, Apr. 2008.

[12] L. Freitag, M. Grund, S. Singh, J. Partan, K. Ball, and P. Koski. The WHOI Micro-modem: An Acoustic Communications and Navigation System for Multiple Platforms. In *Oceans'05*, Sept. 2005.

[13] S. P. Y. Fung, C.-A. Wang, and F. Y. L. Chin. Approximation Algorithms for Some Optimal 2D and 3D Triangulations. *Handbook of Approximation Algorithms and Metaheuristics, CRC Press*, 50, 2007.

[14] H. Füßler, J. Widmer, M. Käsemann, M. Mauve, and H. Hartenstein. Contention-Based Forwarding for Mobile Ad-Hoc Networks. *Elsevier Ad Hoc Networks*, 1(4):351–369, Nov. 2003.

[15] D. Goldenberg, P. Bihler, M. Cao, J. Fang, B. D. Anderson, A. S. Morse, and Y. R. Yang. Localization in Sparse Networks using Sweeps. In *MobiCom'06*, Sep. 2006.

[16] T. He, J. Stankovic, C. Lu, and T. Abdelzaher. SPEED: A Stateless Protocol for Real-Time Communication in Sensor Networks. In *ICDCS'03*, May 2003.

[17] P. Jacquet, P. Mühlethaler, T. Clausen, A. Laouiti, A. Qayyum, and L. Viennot. Optimized Link State Routing Protocol for Ad Hoc Networks. In *INMIC'01*, 2001.

[18] J. Jaffe and C. Schurgers. Sensor Networks of Freely Drifting Autonomous Underwater Explorers. In *WUWNet'06*, Sep. 2006.

[19] B. Jalving. Depth Accuracy in Seabed Mapping with Underwater Vehicles. In *Oceans'99 Riding the Crest into the 21st Century*, Sept. 1999.

[20] J. M. Jornet, M. Stojanovic, and M. Zorzi. Focused Beam Routing Protocol for Underwater Acoustic Networks. In *WUWNet'08*, Sep. 2008.

[21] B. Karp and H. T. Kung. GPSR: Greedy Perimeter Stateless Routing for Wireless Networks. In *MobiCom'00*, Aug. 2000.

[22] 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 *IEEE MILCOM'05*, Oct. 2005.

[23] S. Lee, B. Bhattacharjee, and S. Banerjee. Efficient Geographic Routing in Multihop Wireless Networks. In *MobiHoc'05*, May 2005.

[24] U. Lee, J. Kong, J.-S. Park, E. Magistretti, and M. Gerla. Time-Critical Underwater Sensor Diffusion with No Proactive Exchanges and Negligible Reactive Floods. In *ISCC'06*, June 2006.

[25] U. Lee, Y. Noh, P. Wang, L. F. M. Vieira, M. Gerla, and J.-H. Cui. Pressure Routing for Underwater Sensor Networks. Technical report, UCLA, 2009.

[26] B. Leong, B. Liskov, and R. Morris. Geographic Routing without Planarization. In *NSDI'06*, May 2006.

[27] K. Liu and N. B. Abu-Ghazaleh. Virtual Coordinate Backtracking for Void Traversal in Geographic Routing. In *ADHOC-NOW'06*, Aug. 2006.

[28] S. Poduri, S. Pattem, B. Krishnamachari, and G. S. Sukhatme. Sensor Network Configuration and the Curse of Dimensionality. In *EmNets'06*, May 2006.

[29] D. Pompili, T. Melodia, and I. F. Akyildiz. Routing Algorithms for Delay-insensitive and Delay-sensitive Applications in Underwater Sensor Networks. In *MobiCom'06*, Sept. 2006.

[30] T. Rappaport. *Wireless Communications: Principles and Practice, 2nd Edition*. Prentice Hall, 2002.

[31] J. Ruppert and R. Seidel. On the Difficulty of Tetrahedralizing 3-Dimensional Non-convex Polyhedra. In *ACM Symposium on Computational Geometry*, June 1989.

[32] M. Stojanovic. On the Relationship Between Capacity and Distance in an Underwater Acoustic Communication Channel. In *WUWNet'06*, Sept. 2006.

[33] H. Yan, Z. Shi, and J.-H. Cui. DBR: Depth-Based Routing for Underwater Sensor Networks. In *IFIP Networking'08*, May 2008.

[34] K. Zeng, W. Lou, J. Yang, and D. R. Brown. On Geographic Collaborative Forwarding in Wireless Ad Hoc and Sensor Networks. In *WASA'07*, Aug. 2007.

[35] B. Zhou, Y. Z. Lee, M. Gerla, and F. de Rango. Geo-LANMAR: A Scalable Routing Protocol for Ad Hoc Networks with Group Motion. *Wireless Communications and Mobile Computing*, 6(7):989–1002, 2006.

[36] Z. Zhou and J.-H. Cui. Energy Efficient Multi-Path Communication for Time-Critical Applications in Underwater Sensor Networks. In *MobiHoc'08*, May 2008.

[37] Z. Zhou, J.-H. Cui, and A. Bagtzoglou. Scalable Localization with Mobility Prediction for Underwater Sensor Networks. In *INFOCOM'08*, Apr. 2008.

[38] M. Zorzi and R. R. Rao. Geographic Random Forwarding (GeRaF) for Ad Hoc and Sensor Networks: Energy and Latency Performance. *IEEE Transactions on Mobile Computing*, 2(4):349–365, 2003.