Phero-Trail: a Bio-inspired Location Service for Mobile Underwater Sensor Networks*

Luiz Filipe M. Vieira
UCLA
Computer Science
Department
luizfilipe@ucla.edu

Uichin Lee
UCLA
Computer Science
Department
uclee@cs.ucla.edu

Mario Gerla
UCLA
Computer Science
Department
gerla@cs.ucla.edu

ABSTRACT

A SEA Swarm (Sensor Equipped Aquatic Swarm) moves as a group with water current and enables 4D (space and time) monitoring of local underwater events such as contaminants and intruders. For prompt alert reporting, mobile sensors forward events to mobile sinks (i.e., autonomous underwater vehicles) via geographic routing, thus requiring a location service. In this paper, we analyze various design choices to realize an efficient location service in a SEA Swarm. We find that conventional ad hoc network location service protocols cannot be directly used, because the entire swarm moves along water current. We show that maintaining location information in a 2D plane is an optimal design choice. Given this, we propose a bio-inspired location service called a Phero-Trail location service protocol. In Phero-Trail, location information is stored in a 2D upper hull of a SEA Swarm, and a mobile sink uses its trajectory (à la a pheromone trail of ants) projected to the 2D hull to maintain location information. This enables mobile sensors to efficiently locate a mobile sink via an expanding spiral curve search. Our preliminarily results show that Phero-Trail performs better than existing approaches.

1. INTRODUCTION

A large-scale Underwater Sensor Network (UWSN) architecture has recently been proposed to explore the ocean and in particular, to support solutions for time-critical aquatic applications such as submarine tracking and harbor monitoring [27]. Unlike traditional aquatic monitoring or surveillance applications where sensors are usually tethered to the sea floor or attached to pillars or surface buoys, we assume that a large number of underwater sensor nodes are air-dropped to the venue of interest to create a SEA Swarm (Sensor Equipped Aquatic Swarm) [28]. Each node is equipped with a low bandwidth acoustic modem and with various sensors. It can dynamically control the depth through a fish-like bladder apparatus and a pressure gauge [20].

*This material is based upon work supported by the National Science Foundation under Grant No. CNS 0722046. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the National Science Foundation.

WUWNet'08 September 15, 2008, San Francisco, CA, US.

A SEA Swarm operates and moves as a group (swarm) with water current. Each sensor monitors local underwater activities and reports critical data (or events) in real-time using multi-hop routing to a distant data collection center, e.g., surface buoys or Autonomous Underwater Vehicles (AUVs). We assume that both surface buoys and AUVs are more capable than regular sensor nodes in terms of energy, storage, and communications. They are equipped with GPS and thus can also be used for localization of less capable mobile sensor nodes (e.g., DNR [10, 11]). They are also equipped with wireless communication devices for over-the-surface data reporting (e.g., WiFi and Satellite communications). For tactical missions, AUVs may be equipped with special devices (e.g., weapons to attack enemy submarines).

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) helps provide extra control on redundancy and granularity. Third, floating sensors can help increase system reusability because we can control the depth of a sensor node. When the battery is low or the mission is over, the sensors resurface and can be recharged/reused.

Since high-frequency "radio" signals are quickly absorbed by water, underwater networking must rely on an underwater acoustic channel that has low bandwidth [35] and large propagation latency with five orders of magnitude lower than in the radio channel. An acoustic data transmission consumes much more energy than terrestrial microwave data communications. Moreover, such drastic reduction in communication bandwidth coupled with high latency makes the whole network vulnerable to congestion due to packet collisions. In particular, systematic flooding is unacceptable as it would be intrusive to other services (e.g., a number of AUVs with different tasks). Consequently, minimizing the number of packet transmissions is a very important criterion for protocol design

Because of protocol overhead concerns in a SEA Swarm, we propose that mobile sensors report events to mobile sinks (i.e., AUVs) via 3D geographic routing. Unlike proactive or reactive routing protocols (e.g., OLSR, AODV, etc.), geographic routing is efficient, because statelessness obviates the need of route discovery and maintenance. However, an essential prerequisite of geographic routing is a location service that tells where the destination is.

Our goal in this paper is to devise an efficient, scalable, and robust location service for the SEA Swarm. In general, a location service protocol maintains a set of location servers (or a quorum set) for a given node. These servers are updated and consulted for location updates and queries, respectively. How to maintain a quorum set is the key design factor. For example, we can store location information in a single point using geographic hashing [33], along a horizontal line that covers the network [36] or a circle [34], or in a hierarchical tree over the network area using geographic hashing [12, 30, 40]. Das et al. [7] showed that a hierarchical scheme is most efficient, because geographical hierarchy enables localized updates (i.e., most of location updates are confined in the area where a node resides) and distance-sensitive location discovery (i.e., location discovery cost roughly scales as the distance between inquirer and traget) [7].

In a SEA Swarm, however, it is non-trivial to adopt hierarchical schemes such as GLS [30], HIGH-GRADE [40] and MLS [12], because the entire swarm "moves" along water current. To use geographic hashing, the entire swarm must periodically find a common reference point in the network (e.g., the node with the minimum x, y, z coordinates in the swarm). Maintaining this common reference, however, requires very expensive network-wide 3D flooding. Besides, non-hierarchical schemes that store information in a 1D line must be extended to a 2D plane to guarantee a successful location discovery, but this requires frequent 2D flooding and thus, it is costly as well. In fact, we can formally show that it is a better design choice to provide a location service in 2D planes, instead of 3D space. Yet, in 2D planes, tracking reference point is still expensive, and 1D schemes are less efficient than hierarchical schemes because of their high location update costs.

In this paper, we propose a bio-inspired location service called a Phero-Trail location service protocol. We use the 2D upper hull of a SEA Swarm to store location information. Unlike conventional 1D location protocols that maintain location information in some geometric shapes (line or circle), we use the trajectory of a mobile sink (à la a pheromone trail of ants) to point to its location. To update its location, a mobile sink forwards its current location to its projection in the 2D upper hull. The sequence of projections is basically the equivalent of a "pheromone trail." The maintenance cost of a pheromone trail is minimal, because a mobile sink (AUV) is self propelled and generally moves much faster than the current drifting sensors. Like an airplane contrail, a pheromone trail is slowly diffused in the water current.[1] The length of a trail is controlled by setting a pheromone expiration timer. A mobile sink can also update its trail such that the probability that an update packet propagates a certain distance is inversely proportional to the distance and thus, the average update cost is comparable with that of a hierarchical scheme. For a location query, a sensor node uses an expanding spiral curve search (i.e., a 1D curve instead of a 2D disk as in a typical expanding ring search) in the 2D hull to find the target location. This enables distance-sensitive search in most scenarios. We analyze the performance of location update/retrieval and storage overhead and validate the performance using simulations. Our results show that in practice Phero-Trail yields the key benefits of hierarchical schemes (i.e., localized updates and distance-sensitive search) even without maintaining geographical hierarchy.

This paper is organized as follows. In Section 2, we describe a SEA Swarm application scenario. In Section 3, we evaluate the design space of a location service in a SEA Swarm sensing platform. In Section 4, we propose a novel location service protocol. In Section 5, we evaluate the proposed protocol via simulations. In Section 6, we overview the related work. Finally, we conclude the paper and describe future work in Section 7.

2. SEA SWARM APPLICATION SCENARIO

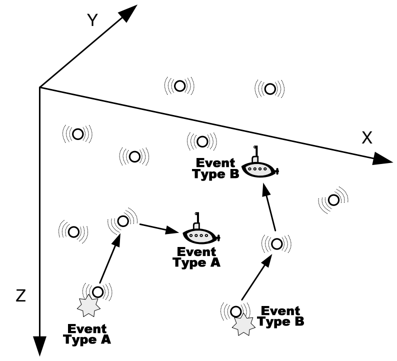

Figure 1: An illustration of a SEA Swarm: mobile nodes detect events and reports them to corresponding mobile sinks

We are interested in protecting critical installation such as harbor, underwater mining facility, and oil rigs. A SEA Swarm is deployed in a square region of size L m $\times$ L m whose depth ranges from D_1 to D_2 ($D = D_2 - D1$). Nodes can adjust their depths using bladders and on-board pressure gauges [20].

The swarm moves along water current, searching for a stranded submarine or scouting the waters around a friendly convoy to detect underwater intruders. There are a few unmanned submarines (or AUVs) in the swarm that assist in further investigation of the alert situation. For instance, they can help the rescue of stranded vessels, and they may also carry weapons to destroy attackers. The mission requires the exchange of information among sensors and "sink" nodes (See Figure 1). Whenever a sensor detects an event of interest, it must route an alert to one or more of unmanned submarines. The selection of a submarine depends on the nature of the detected event. There is at least one corresponding submarine for each type of an event. For instance, if a sensor detects a type A event, it sends the event to submarine A; type B to submarine B, and so on. We assume that the submarines patrol constantly the swarm area using random direction mobility.[2]

An ad hoc routing protocol is necessary in order for the moving sink to receive the data from mobile sensors. As illustrated earlier, proactive routing protocols such as OLSR requires period updates and reactive routing protocols such as AODV requires flooding for route discovery. Directed Diffusion [19], a popular technique in sensor networks to establish routes to sinks, also involves flooding for route discovery and requires route maintenance especially in mobile environments. To avoid the overhead of these approaches, we use beaconless geographic routing in 3D environments [13, 15, 24]. Once the location of the destination node (i.e., the sink node) is known, a mobile sensor can exchange location updates while communicating with a mobile sink. Geographic routing requires a geographic location service, and this paper aims at designing an efficient and scalable location service for a SEA Swarm. In the following, we show our design choice and describe the protocol details.

Note that we assume that nodes can localize their positions using existing localization techniques. In particular, we use protocols where nodes can "passively" learn their positions from other nodes (e.g., AUVs) to minimize energy consumption such as Dive'N'Rise

[1] Contrails or vapor trails are condensation trails (or artificial clouds) made by the exhaust of aircraft engines that precipitate a stream of tiny ice crystals in moist, frigid upper air.

[2] They can also use a Lévy Walk where occasional long jumps are combined with short range searches. This produces a more effective search when the target locations are unknown and are randomly dispersed in a large area [38]. Thanks to the long jumps, the area covered by the agents will be much larger than the area that would have been covered by only Random Walk movement patterns.

(DNR) [11], mobility prediction [41] or using autonomous underwater vehicle [10].

3. PROTOCOL DESIGN SPACE ANALYSIS

We analyze the performance of existing location service protocols in a SEA Swarm scenario. We assume that N mobile nodes are deployed in a L m $\times$ L m $\times$ D m cubic space, and mobile node has a radio range of R m. For the sake of analysis, we assume that $D = L$, but in practice, the width of a monitoring region size is much greater than the depth. Let M denote the number of hops to travel a width of a network; i.e., $M = L/R$. We consider the following location service protocols, namely naïve flooding, quorum-based schemes, and hierarchical schemes. The operations of a location service are location "update" and "retrieval" (query).

- *Naïve flooding*: A node periodically floods its current position to the entire network.
- *Quorum-based schemes*: Each location update of a node is sent to a subset of nodes (or update quorum), and similarly, a location query is sent to a subset of nodes (or query quorum). The query will be resolved only when these two subsets are designed such that their intersection is non-empty. For instance, in XYLS, each node propagates its location update in the vertical direction, whereas any location queries are disseminated in the horizontal direction (see Figure 2).
- *Hierarchical schemes*: Location servers are chosen via a hash function in the nodes' identity space. The area in which nodes reside is recursively divided into a hierarchy of smaller grids, thus building a hierarchical tree of the network area. The root of a tree covers the whole network area, and each of its siblings covers a smaller grid whose size is one fourth of the network area (see Figure 3). For each node, one or more nodes in each grid at each level of the hierarchy are chosen as its location servers; i.e., a mobile node stores information along the path of a tree. Recall that each node in a path of the tree is the part of the network area where a mobile node resides. A location query is routed successfully to higher hierarchical levels until it finds a rendezvous point. It then traverses down the hierarchy to find the exact location. Location updates are processed locally and propagate to the higher levels only when a mobile node crosses the boundary. We assume that the lowest level has a grid size of R m $\times R$ m where R is the radio range. Level i grid has the size of $2^{i-1}R$ m $\times 2^{i-1}R$ m. Let H denote the maximum level of the hierarchy, satisfying the relationship of $L = 2^{H-1}$.

We analyze the approximate costs of update and query of the above schemes in a SEA Swarm. Note that we report the performance in an asymptotic notation with a function of M and H. For this, both values are functions of the radio range which is a function of the number of nodes in the network. Gupta et al. [17] showed that given a unit cubic area ($L = 1$ m), the radio range must be set as $R = \sqrt[3]{\log n/n}$ in order to guarantee connectivity.

Let us first consider a 3D environment. Naïve flooding requires network-wide flooding and the cost is $\Theta(M^3)$. In quorum-based schemes, we must store information in a 2D plane such that any vertical query traversal can retrieve the location information. The update cost scales as $\Theta(M^2)$, and the query costs scales as $\Theta(M)$.

In hierarchical schemes, we must first find a reference point for geographic hashing and propagate this information to every node. Every node starts broadcasting its location to one's neighbors. After receiving a set of location information from its neighbors, a node

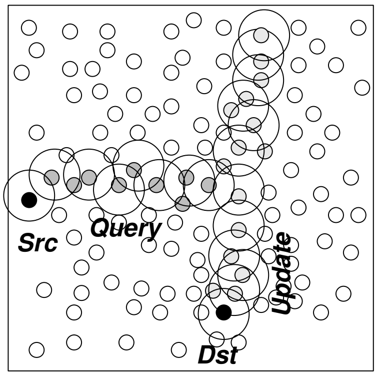

Figure 2: Quorum-base scheme: XYLS

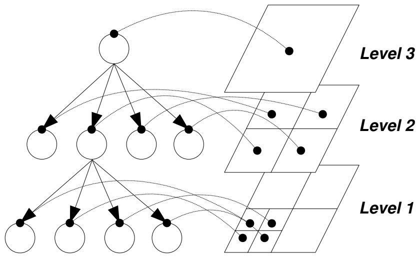

Figure 3: Hierarchical scheme: GLS

broadcast the smallest coordinate that it has ever seen. By repeating this process, nodes can find the reference point in $\Theta(M^3)$ time steps. Since every node must participate in updating this information, the aggregate cost is upper bounded by $O(nM^3)$.[3] After setting the reference point, we can form a hierarchy tree in the cube. The average update cost can be calculated as follows. The probability that the update distance is 2^kR is given as $1/2^k$. The average update distance is $\sum_{k=1}^{H} 2^k RPr[d = 2^kR] = H$ where H is the maximum hierarchical level. Since a submarine can be in a random position in the network, the average distance between nodes is proportional to the width of a network, and the average query cost is given as $O(M)$. Here, we note that the reference updating cost dominates other costs (i.e., update and query).

Given this, we may want to consider the design choice of storing location information in a 2D plane. Since we can control the depth of sensor nodes, it is not difficult to store information in a 2D plane at a certain depth of a SEA Swarm. Location update and query packets are first "vertically" routed to a 2D plane and then are routed according to a location service protocol in a 2D environment. The cost of each operation can be calculated in the same way as above. Naïve flooding requires network-wide flooding, and update and query costs are $\Theta(M^2)$ and $\Theta(1)$ respectively. In quorum-based schemes, we must store information in a 1D line such that any horizontal query traversal can retrieve the location information. Update and query costs scale as $\Theta(M)$. In hierarchical schemes, the reference update, location update and query opera-

[3] Since the smallest coordinate has the highest priority, the overall process finishes after it is broadcast to all nodes. Keshavarz-Haddad et al. [25] showed that broadcasting requires $\Theta(1/R^3) = \Theta(M^3) = \Theta(n/\log n)$ transmissions. In the worst case, every transmission opportunity in the network has to be fully used for this purpose, and thus, the total number of transmissions is upper bounded by $\Theta(nM^3) = \Theta(n^2/\log n)$.

tions take $\Theta(M^2)$, $\Theta(H)$, and $\Theta(M)$ respectively.[4]

In summary, we show that any geographic hashing based scheme is not efficient in a SEA Swarm, because the overhead of "periodical" reference point updates dominates the update/query overhead. Keeping a quorum set in a 3D environment is costly, because it requires 2D flooding for location updates. If we keep a quorum set in a 2D environment, this can be reduced to $\Theta(M)$ in spite of vertical routing overhead. From this, we conclude that among all the design choices, a 2D quorum based approach provides the least expensive solution. Thus, we adapt a 2D quorum-based protocol to design an efficient and scalable location service in a SEA Swarm.

We briefly review the two key benefits of hierarchical schemes, namely localized updates and distance-sensitive location discovery, which plays a key role in the design of the proposed protocol in the paper. First, the current location updates are maintained locally in order to minimize update costs, yet this still enables other nodes to find a list of pointers that lead to the up-to-date information. Information updates need not be synchronized at all the levels. In contrast, the conventional quorum-based protocols must update the quorum sets for every update, thus incurring high update costs. Second, the hierarchical schemes guarantee that the query cost is proportional to the distance between inquirer and target; i.e., given that two nodes are under level k, they can find the pointer that leads to the target in level $k+1$ and thus, the search area size is bounded. Note that distance sensitivity is only important when the depth of a deployed area is much smaller than the width. Otherwise, the overall cost is dominated by the vertical routing cost. Although the query routing overhead in a plane is strictly less than $\Theta(M)$, the vertical routing overhead is still $\Theta(M)$. Thus, the overall overhead remains the same as $\Theta(M)$. In the following section, we will consider all above issues in the design of the SEA Swarm location service.

4. PHERO-TRAIL LOCATION SERVICE

In this section we describe the protocol design assumptions and present the details of the proposed location service protocol. We derive the asymptotic performance results of the Phero-Trail scheme in terms of location update/query overheads and storage costs.

4.1 Protocol Design Assumptions

Mobile Node Characteristics: We assume that the speed of submarines is much faster than that of the sensor nodes. For instance, a sensor node moves along the water current at the average speed of 0.5 m/s [6], whereas a submarine can move at up to 10 knot speed, or 5.2 m/s. Given that low cost commercial off-the-shelf AUVs are able to deliver speeds up to 7 knots [22], we can assume that our specialized submarine (equipped with weapons as well) can move faster than such AUVs. We assume that submarines patrol the area at a constant speed with random direction mobility. More precisely, a node first picks a direction and travel distance; then, it moves to the target destination along the geographically shortest path.

Convex Hull of a SEA Swarm: We use a 2D convex hull of a SEA Swarm for the following reasons. First, fixing an arbitrary depth as a reference does not guarantee that there will be sensors at that depth all the time. Second, it is difficult to maintain the plane "thin" and "flat" because a sensor never knows whether it is part of the plane or slightly above or below; i.e., the height of a plane may be thicker than the upper hull. Third, the performance loss due to routing packets to the upper hull is minimal, because a typical swarm has a vertical dimension smaller than other dimensions. Moreover, vertical communications are more efficient than

[4]The worst case update cost is bounded by $\Theta(n/M \times M^2) = \Theta(nM)$.

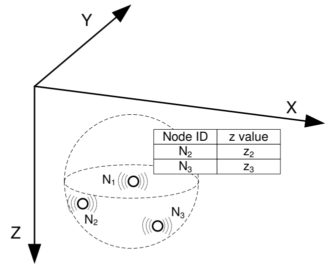

Figure 4: Approximate convex hull construction: a node with the smallest z within its communication range becomes part of the hull servers

horizontal communications [2].

4.2 Phero-Trail Protocol Operations

We describe how to maintain an upper convex hull in a SEA Swarm and show the details of Phero-Trail protocol operations, namely location update and query.

Convex Hull Maintenance: In Phero-Trail, mobile nodes on the surface of deployment becomes location servers. The decision is based solely on local information and requires only the z-coordinate. Note that z coordinate is zero at surface and increases with depth. Sensors can estimate their depth via on-board pressure gauges. Once they acquire this information, they broadcast this information. Nodes within the communication range will receive the message and store the node ID and z-coordinate value. After receiving messages from all its neighbors, a node can decide if it will act as a server by checking whether it has the smallest z value among all its neighbors.

Figure 4 illustrates the construction of a distributed convex hull geographic location server. Nodes exchange the z-coordinate values among themselves. The one with the smallest depth, represented as node N_1 becomes part of the hull servers. This simple method only finds an approximation of the upper convex hull, because a node with locally smallest z may not be on the convex hull. However, a convex hull node must be "locally minimal." Consequently, the approximate scheme includes extra nodes as servers, thus improving robustness.

Location Update: The key idea of Phero-Trail is to store the location updates along the projected trajectory of an AUV on the upper hull. While moving, the AUVs leaves pheromones on mobile sensor nodes in the upper convex hull by sending a pheromone packet that has the current position and other information such as directions [29] and a trajectory equation [32] of the submarine (see Figure 5). Upon receiving a pheromone packet, a node on the hull will extend a pheromone trail by connecting to the previous node on the trail. The length of a pheromone trail is fixed to $\mu 2^{H-1}$ where μ is a constant system parameter. It scales as the width of a network area, and pheromone packets expires after $\mu 2^{H-1} R/v$ seconds where v is the speed of a submarine. For efficient location retrieval, we can set the location update to propagate along the trail. We mimic the behavior of a hierarchical scheme such as GLS by setting the probability that the update propagation distance is $2^k R$ is to be given by $1/2^k$ where we have $k < H$ and H is the maximum level.

Unlike conventional 2D quorum-based protocols such as XYLS [36] or Double Ruling [34], Phero-Trail does not frequently update the trail, but updates are locally stored by forming a pheromone trail just as in a hierarchical scheme. A pheromone packet is not necessary stored in a specific geographic location. Instead, mobile sensors cooperatively keep the pheromone trail such that it can sur-

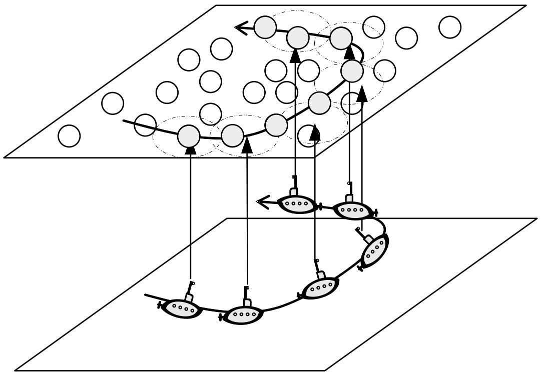

Figure 5: Location updates on the upper convex hull

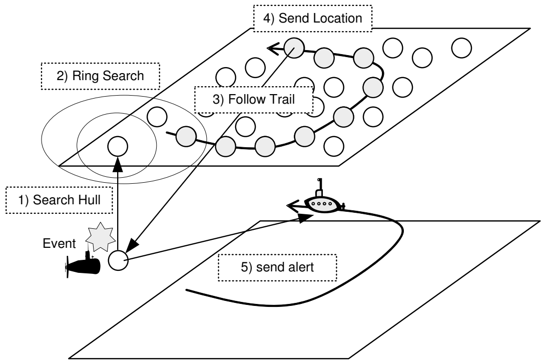

Figure 6: Location query in Phero-Trail

vive even when some of nodes with pheromone packets move away from the trail. A pheromone trail moves along the water current. Recall that the speed of a submarine is much faster than mobile sensors. For instance, it takes 2000 s for an AUV moving at the speed of 5 m/s to travel 10 km, creating a 10 km long pheromone trail. The pheromone that an AUV left in the beginning will have the maximum dispersion of 1 km (at the speed of 0.5 m/s). The average dispersion will be much smaller than 1 km in practice. Moreover, pheromone expires, and its hop stretch (the difference between the original trail length and the dispersed trail length) can be bounded.

Note that it is possible that there is a hole in the convex hull. In this case, the last node uses hop-limited flooding to connect to the previous node in the trail. Holes below the convex hull do not affect the integrity of the pheromone trail. They do affect the "query" packet position routing to the hull, but this problem can be handled using known recovery modes for 3D geographic routing proposed by Flury et al. [13].

Location Query: A mobile node first routes a query packet vertically upwards to the node on the projected position of the convex hull plane. After this, the node performs an expanding spiral curve search to find a pheromone trail. Note that it is not a conventional ring search; instead of a disk-based search (2D disk), it uses a spiral curve such that a query packet travels along a circle with radius r. For each step k, we expand the radius exponentially, i.e., $2^k R$. If the spiral curve intersects with the trail (i.e., a trail is found), the node first calculates the expected travel distance of a submarine based on the timestamp in the pheromone packet. If the cost of trail traversal is greater than that of an additional curve search, the node performs another curve search by incrementing k. Otherwise, the node sends a packet along the trail to find the up-to-date destination coordinate. Note that the node starts from the best point found on the trail after curve search. While traversing the trail, we can take the advantage of short-cuts provided by the periodical updates. The various phases of the Phero-Trail search are illustrated in Figure 6.

4.3 Phero-Trail Analysis

We analyze location update/query overheads and storage requirement of Phero-Trail. Since the location update scheme shares the same idea as the hierarchal schemes, the location update cost is given as $\Theta(H)$, assuming that the vertical routing overhead is much smaller than the horizontal routing overhead (i.e., the depth of a SEA Swarm is much smaller than the width of a swarm). If the vertical routing cost is given as $O(M)$, the overall cost of the location update becomes $O(M)$. To show the search cost, we analyze the worst case of an expanding spiral curve search. Since we exponentially increase the ring size, the overhead is given as $\sum_{k=1}^{H} 2^k R = \frac{2^{H+1}-1}{2} R = \Theta(2^H) = \Theta(M)$. The search cost is the same as in conventional schemes. Unlike a hierarchical scheme where the storage cost is proportional to $\Theta(H)$, the storage overhead is proportional to the trail length, i.e., $\mu 2^H$. Note that for robustness of a trail, we can keep the width of a trail as wR where $w \geq 1$, the trail area size becomes $\mu 2^H 2wR$. Given that the density of nodes is ρ (uniform density), the storage requirement can be approximated as $\mu 2^H 2wR\rho = \Theta(2^H)$.

As long as the cost of vertical routing is much smaller than that of horizontal routing, we show that the performance of Phero-Trail is comparable with that of a hierarchical scheme such as GLS and HIGH-GRADE even without geographic hashing! Another key property of a hierarchical scheme is "location-sensitive" query resolution. Unfortunately, Phero-Trail does not guarantee such behavior. In particular this happens when there is a spiral movement pattern. Say that an inquiring node is located very closely to the target node. Although the node is able to find the intersection points as a result of curve search, one has to follow a long "spiral curve" to reach the target. Yet, we assume that submarines are moving based on random direction mobility. The probability that a node travels along such a spiral is very low. In practice, a node is able to find the location without traveling longer distance and "roughly" provides "location-sensitive" query resolution.

5. SIMULATIONS

As a proof of concept, we perform a preliminary simulation study of our proposed scheme. We implement Phero-Trail and a geographical routing protocol in a simple C simulator. Nodes are deployed in a grid space whose size is 1 km $\times$ 1 km $\times$ 1 km. A network is configured dense in order to provide connectivity, and thus, geographic routing uses greedy forwarding. We compare the performance of Phero-Trail with naïve flooding. We use the Meandering Current Mobility (MCM) Model that captures a micro mobility of water current [6]. The communication range is set to 300 m, and submarines move at the speed of 5 m/s. We vary the network size n, or the number of nodes in one dimension, and measure the number of transmitted messages. Figure 7 shows the number of messages during update. The figure clearly shows that the Phero-Trail scheme has much smaller overhead. Figure 8 shows the number of transmitted messages with the number of submarines. The graph clearly shows Phero-Trail performs much better than a naïve flooding approach.

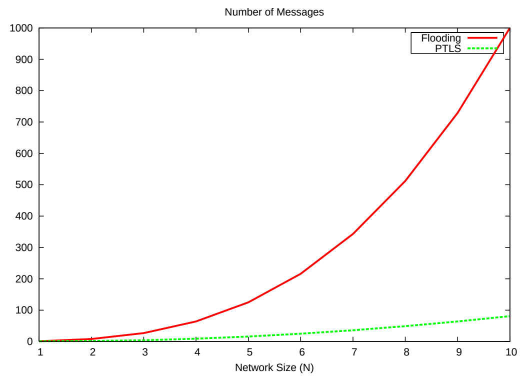

Figure 7: Number of messages for updating as a function of network size

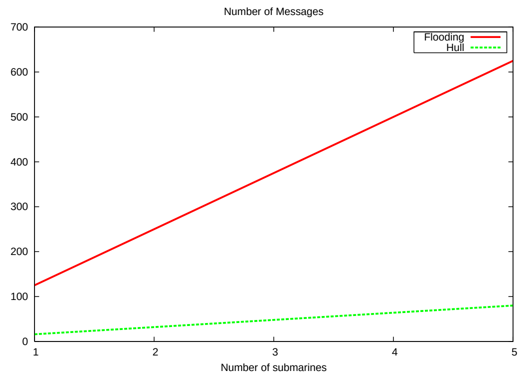

Figure 8: Number of messages as a function of the number of submarines

6. RELATED WORK

Location Services: Location services can be divided into two categories in general, namely flooding-based, and rendezvous-based approaches. Intermediate approaches can also be constructed. As in ad hoc routing protocols, flooding based protocols are either proactive or reactive. For instance, DREAM [3] is a proactive algorithm; LAR [26] is a reactive algorithm. In rendezvous-based protocols, all nodes agree upon a certain mapping that maps one's ID to a set of nodes (or location servers, rendezvous nodes) in the network. There are two different mapping methods widely used, namely quorum-based and hashing-based schemes.

In the quorum-based method [18], each update is sent to a set of nodes (update quorum); similarly, a location query is sent to another set of nodes (query quorum). If two sets (update/query quorums) intersect, a query is resolved. Various methods of keeping the quorums have proposed and readers find the survey in [14]. In XYLS [36], a node stores location updates in a vertical line and retrieves the location by sending a query in a horizontal direction. Sarkar et al. [34] proposed double ruling methods where they store information on a 1D curve (circle) in a sensor network. The consumer travels along another curve which guarantees to intersect with the producer curve. It is an extension of the hashing scheme of GHTs with improved query locality, i.e., consumers close to producers. Phero-Trail uses a similar concept, but a location update is locally stored instead of always propagating on a curve.

In hashing-based protocols, location servers are chosen via a hash function in the node's identifier space. Hashing schemes can be further divided into flat or hierarchical schemes. In a flat scheme such as GHT [33], location information is stored in a single geographic location. In a hierarchical hashing-based scheme such as GLS [30], HIGH-GRADE [40] and MLS [12], the area in which nodes reside is recursively divided into a hierarchy of smaller grids. At each level, a set of nodes determined by the hash function serve as location servers for a given node. HIGH-GRADE [40] introduces the concept of "level-of-indirection" where instead of storing exact location, location servers in a hierarchy stores only the pointers to lower levels. MLS [12] further improves HIGH-GRADE in terms of hop stretch and protocol correctness.

Besides the above-mentioned schemes, there are intermediate approaches using encounter history and mobility (also known as last encounter routing). A node publishes its current location to those who encounter a target node. Encounter history is disseminated via node mobility. For location discovery, a node searches for any intermediate node that encountered the target node more recently through expanding disk search. Having found such an intermediate node, EASE (Exponential Age SEarch) [16] estimates target's location as the position where the intermediate node encountered the target, and FRESH (FResher Encounter SearcH) [9] does it as the current position of the intermediate node. Since a query packet can travel much faster than swarming sensor nodes, by repeating this process, one can quickly find the target's location. This can be seen as "approximate" level-of-indirection to the exact location as in hierarchical schemes. These approaches assume mobility models where encounter history well diffuses around the network. However, the mobility of water current is directional, and its speed is much slower than the mobile sinks, making encounter history dissemination hard. As a result, warming-up will take a very long time, and the expanding disk search will incur lots of overhead. Thus, these schemes will not work well in a SEA Swarm.

Based on this concept, Westphal et al. [39] proposed Bread Crumb (BC) protocols for a scenario where a mobile sink harvests data in a "static" sensor network. Unlike the above approach, there is no mobility and the encounter history becomes a "precise" level-of-indirection. Thus, a query packet is forwarded by following the static age gradient to the mobile sink. However, a mobile sink has to cover the whole network to first create such a gradient at an arbitrary position, which will take excessively long time with mobility of nodes in a SEA Swarm scenario. Moreover, BC neither handles the node mobility nor provides a search mechanism. Phero-Trail takes advantage of high speed mobility of AUVs, explicitly maintains a pheromone trail whose length is controllable, and provides an efficient spiral curve search algorithm to retrieve location information.

Bio-inspired Networking Systems: Understanding key ideas of how living organisms efficiently organize unreliable and dynamically changing resources and applying these ideas to distribute computing has been an active area of research for the past decade. Babaoglu et al. [1] summarize this by proposing a conceptual framework that captures a few basic biological processes such as diffusion, chemotaxis, and stigmergy. Readers can find the principles of collective animal behavior in [37]. Benefits of bio-inspired technologies for network embedded systems are well documented in [8].

Several research activities, e.g., AntNet [4], have proposed Ant Colony Optimization (ACO) for routing in packet-switched networks. For ad hoc routing, a few proposals have already emerged, such as ARA [31], PERA [23], and AntHocNet [5]. ARA and PERA are quite similar to a reactive ad hoc routing protocol, e.g., AODV. On the contrary, AntHocNet is a hybrid (both proactive and reactive) multi-path ad hoc routing protocol and consists of two main processes: stigmergic learning and diffusion. During stigmergic learning, nodes send out ant-like agents (similar to RREQ control packets in AODV) which sample and reinforce good paths to the destination. Routing information is kept in an array of stigmergic variables, called "pheromone tables." ARA and PERA share the same concept, but in AntHocNet, this mechanism is further supported by a diffusion process that spreads this information to other agents. Packets are routed under the probabilistic guidance of the learned pheromone tables. In Phero-Trail, we store the location information in such stigmergic variables to form a pheromone trail and thus, to support efficient location update and retrieval.

7. CONCLUSION AND FUTURE WORK

In this paper, we studied geographic location services to enable geographic routing in a SEA Swarm. We showed that maintaining location information in a 2D plane is optimal in the environment under consideration. Given this, we designed a novel bio-inspired location service called a Phero-Trail location service protocol where a trajectory of a mobile sink is projected to the nodes in the 2D upper convex hull for location update and retrieval. Under reasonable assumptions, we showed that the performance of Phero-Trail is comparable with the hierarchical schemes even without using geographic hashing.

There are several interesting avenues for future work on this subject. First, we will adapt various location services protocols such as XYLS [36] and GLS [30] in the SEA Swarm scenario and compare the performance with Phero-Trail. Second, we will evaluate the performance of Phero-Trail with various system configurations such as the number of sensors/sinks, the speed of sensors/sinks, the deployment area size (including various depths), and the search pattern of mobile sinks. Third, the SEA swarm scenario may experience network partitioning. We will study methods of overcoming short- and long-term disconnections using Delay Tolerant Networking techniques [21]. Fourth, we will use mobility prediction methods to reduce the overhead of maintaining the common reference point in hierarchical schemes [41]. This enhanced hierarchical scheme will be compared with Phero-Trail. Finally, we will investigate the issues of balancing loads and energy consumption, because the upper hull is extensively utilized than other layers. The problem can be mitigated by dynamically controlling the depth of mobile sensors, or gracefully switching the depth of a 2D plane over time. We will investigate the trade-offs.

8. REFERENCES

[1] O. Babaoglu, G. Canright, A. Deutsch, G. A. D. Caro, F. Ducatelle, L. M. Gambardella, N. Ganguly, M. Jelasity, R. Montemanni, A. Montresor, and T. Urnes. Design patterns from biology for distributed computing. *ACM Trans. Auton. Adapt. Syst.*, 1(1):26–66, 2006.

[2] L. Badia, M. Mastrogiovanni, C. Petrioli, S. Stefanakos, and M. Zorzi. An Optimization Framework for Joint Sensor Deployment, Link Scheduling and Routing in Underwater Sensor Networks. In *WUWNet'06*, Los Angeles, CA, Sep. 2006.

[3] S. Basagni, I. Chlamtac, V. R. Syrotiuk, and B. A. Woodward. A Distance Routing Effect Algorithm for Mobility (DREAM). In *MOBICOM'98*, Dallas, TX, 1998.

[4] G. D. Caro and M. Dorigo. AntNet: Distributed Stigmergetic Control for Communications Networks. *Artificial Life*, 5(2):137–172, 1999.

[5] G. D. Caro, F. Ducatelle, and L. M. Gambardella. AntHocNet: An Adaptive Nature-Inspired Algorithm for Routing in Mobile Ad Hoc Networks. *Telecommunications (ETT), Special Issue on Self Organization in Mobile Networking*, 16(2):137–172, 2005.

[6] 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*, Phoenix, AZ, Apr. 2008.

[7] S. M. Das, H. Pucha, and Y. C. Hu. On the Scalability of Rendezvous-based Location Services for Geographic Wireless Ad Hoc Routing. *Elsevier Computer Networks*, 51(13):3693–3714, 2007.

[8] F. Dressler. Benefits of Bio-inspired Technologies for Networked Embedded Systems: An Overview. In *Dagstuhl Seminar 06031 on Organic Computing - Controlled Emergence*, Dagstuhl, Germany, Jan. 2006.

[9] H. Dubois-Ferriere, M. Grossglauser, and M. Vetterli. Age Matters: Efficient Route Discovery in Mobile Ad Hoc Networks using Encounter Ages. In *ACM MOBIHOC*, Annapollis, MD, USA, Jun. 2003.

[10] M. Erol, L. F. M. Vieira, and M. Gerla. AUV-Aided Localization for Underwater Sensor Networks. In *WASA'07*, Chicago, IL, Aug. 2007.

[11] M. Erol, L. F. M. Vieira, and M. Gerla. Localization with Dive'N'Rise (DNR) Beacons for Underwater Acoustic Sensor Networks. In *WUWNet'07*, Montreal, Quebec, Canada, Sept. 2007.

[12] R. Flury and R. Wattenhofer. MLS: An Efficient Location Service for Mobile Ad Hoc Networks. In *MobiHoc'06*, Florence, Italy, May 2006.

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

[14] R. Friedman and G. Kliot. Location Services in Wireless Ad Hoc and Hybrid Networks: A Survey. Technical Report CS-2006-10, Technion - Israel Institute of Technology, Apr. 2006.

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

[16] M. Grossglauser and M. Vetterli. Locating Nodes with EASE: Last Encounter Routing in Ad Hoc Networks through Mobility Diffusion. In *INFOCOM'03*, San Francisco, CA, May 2003.

[17] P. Gupta and P. R. Kumar. The Capacity of Wireless Networks. *IEEE Transactions on Information Theory*, 46(2), 2000.

[18] Z. J. Haas and B. Liang. Ad Hoc Mobility Management with Uniform Quorum Systems. *IEEE Transaction on Networking*, 7(2):228–240, 1999.

[19] C. Intanagonwiwat, R. Govindan, and D. Estrin. Directed Diffusion: a Scalable and Robust Communication Paradigm for Sensor Networks. In *ACM MOBICOM'00*, Boston, MA, USA, 2000.

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

[21] S. Jain, K. Fall, and R. Patra. Routing in a Delay Tolerant Networking. In *SIGCOMM'04*, Portland, OR, Aug.-Sep. 2004.

[22] B.-H. Jeon, P.-M. Lee, J.-H. Li, S.-W. Hong, Y.-G. Kim, and

J. Lee. Multivariable Optimal Control of an Autonomous Underwater Vehicle for Steering and Diving Control in Variable Speed. In *OCEANS'03*, San Diego, CA, Sept. 2003.

[23] H. M. John S. Baras. A Probabilistic Emergent Routing Algorithm for Mobile Ad Hoc Networks. In *WiOpt'03*, INRIA Sophia-Antipolis, France, 2003.

[24] H. Kalosha, A. Nayak, S. Ruhrup, and I. Stojmenovi. Select-and-Protest-Based Beaconless Georouting with Guaranteed Delivery in Wireless Sensor Networks. In *INFOCOM'08*, Phoenix, AZ, Apr. 2008.

[25] A. Keshavarz-Haddad, V. Ribeiro, and R. Riedi. Broadcast Capacity in Multihop Wireless Networks. In *Mobicom'06*, Los Angeles, CA, Sept. 2006.

[26] Y.-B. Ko and N. H. Vaidya. Location-Aided Routing (LAR) in Mobile Ad Hoc Networks. In *MOBICOM'98*, Dallas, TX, 1998.

[27] 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*, Atlantic City, NJ, Oct. 2005.

[28] 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*, Sardinia, Italy, June 2006.

[29] Y.-Z. Lee, M. Gerla, J. Chen, J. Chen, B. Zhou, and A. Caruso. *DSR: The Dynamic Source Routing Protocol for Multihop Wireless Ad Hoc Networks*, pages 357–366. Springer Boston, 2006.

[30] J. Li, J. Jannotti, D. S. J. D. Couto, D. R. Karger, and R. Morris. A Scalable Location Service for Geographic Ad Hoc Routing. In *MOBICOM'00*, Boston, MA, Aug. 2000.

[31] I. B. Mesut Güneşćň, Udo Sorges. ARA – The Ant-Colony Based Routing Algorithm for MANETs. In *ICPPW'02*, 2002.

[32] D. Niculescu and B. Nath. Trajectory Based Forwarding and Its Applications. In *MOBICOM'03*, San Diego, CA, 2003.

[33] S. Ratnasamy, B. Karp, S. Shenker, D. Estrin, R. Govindan, L. Yin, and F. Yu. Data-Centric Storage in Sensornets with GHT, A Geographic Hash Table. *Springer Mobile Networks and Applications*, 8(4):427–442, 2003.

[34] R. Sarkar, X. Zhu, and J. Gao. Double Rulings for Information Brokerage in Sensor Networks. In *MOBICOM'06*, Los Angeles, CA, Sept. 2006.

[35] M. Stojanovic. On the Relationship Between Capacity and Distance in an Underwater Acoustic Communication Channel. In *WUWNet'06*, Los Angeles, CA, Sep. 2006.

[36] I. Stojmenovic, D. Liu, and X. Jia. A Scalable Quorum based Location Service in Ad Hoc and Sensor Networks. In *MASS'06*, Vancouver, Canada, Oct. 2006.

[37] D. J. T. Sumpter. The Principles of Collective Animal Behaviour. *Philosophical Transactions of the Royal Society of London: Series B*, 361:5–22, Nov. 2005.

[38] G. Viswanathan, S. V. Buldyrev, S. Havlin, M. G. E. da Luz, E. P. Raposo, and H. E. Stanley. Optimizing the Success of Random Searches. *Nature*, 401:911–914, 1999.

[39] C. Westphal. Little Tom Thumb Went Straight Home. In *INFOCOM'07*, Anchorage, AK, May 2007.

[40] Y. Yu, G.-H. Lu, and Z.-L. Zhang. Enhancing Location Service Scalability with HIGH-GRADE. In *MASS'04*, Fort Lauderdale, FL, Oct. 2004.

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