Purposeful Mobility in Tactical Sensor Networks

Guohong Cao, G. Kesidis, T. La Porta, Bin Yao,* and Shashi Phoha
Department of Computer Science & Engineering
The Pennsylvania State University
University Park, PA 16802

Abstract

Adding mobility to sensor networks can significantly increase the capability of the sensor network by making it resilient to failures, reactive to events, and able to support disparate missions with a common set of sensors. Mobility in sensor networks may be controllable, and hence be used to help achieve the network's missions. That is, mobility may be "purposeful" instead of being treated as an uncontrollable external stimulus to which the ad-hoc networks must respond. To make use of the purposeful mobility, we propose techniques for mobility assisted sensing and routing considering the computation complexity, network connectivity, energy consumption of both communications and movement, and the network lifetime. We also define utility functions that can capture the benefits of the movement from the perspective of all missions, and maximize the capability of the network.

1 Introduction

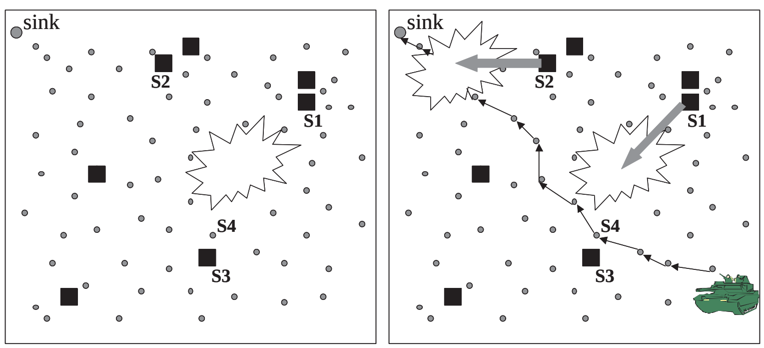
Recent advances [1, 30] in hardware design are enabling low-cost sensors that have sophisticated sensing, communication, and computation capabilities, to accomplish multiple, disparate missions [15]. These sensors communicate via radio transmitters/receivers to form a multihop wireless network, i.e., a distributed wireless sensor network [14]. Sensor networks can automate information gathering and processing, and therefore can support many applications (missions) such as target tracking, perimeter defense, homestead monitoring, and intelligent transportation.

As sensors become widely deployed, multiple missions, each with different requirements, may share common sensors to achieve their goals. Each mission may have its own requirements for the type of data being reported, the sampling rate, accuracy, and location of the sampling. As a single sensor network needs to support different sets of missions under different conditions, the requirement on physical sensor locations becomes dynamic. For example, in target tracking missions, enough sensors should be deployed along the track of the target, whereas in perimeter defense, the requirement is to have adequate sensors along a prescribed perimeter. This dynamic requirement on sensor locations cannot be easily met by deploying a large number of sensors, since provisioning for all possible combinations of mission requirements may not be economically feasible. More importantly, precise sensor deployment may not be possible, especially in a hostile environment, where sensors are subject to power depletion, failures, malicious attacks, and may

*Bin Yao is with Bell Labs, Lucent Technologies.

change their physical locations due to external force. Therefore, a fixed sensor network has limitations when applied to support multiple missions or when the network conditions change. In such cases, mobile sensors are essential.

Figure 1 illustrates an example of using mobile sensors. Initially, fixed sensors indicated by round dots, are randomly distributed. Several mobile sensors represented by squares, are also deployed in the network. The mission of the network is to monitor the perimeter of a field, and track any target that enters the field. This general mission has specific instances in airport control, home security, military operations, etc. During network operation, certain sensors fail due to energy depletion or are destroyed by an external force (fire, bomb), creating *coverage holes* which are not covered by any sensor as shown in Figure 1(a). Since there are enough sensors along the perimeter of interest to fulfill the initial mission of the sensor network, i.e., perimeter monitoring, no adjustment is necessary at this time. Suppose a target enters the field; the mission of the sensor network is to track the target. Since the track taken by the target may be arbitrary, it is desirable to fill the coverage holes. This can be achieved by using the mobile sensors as shown in Figure 1(b). As the target approaches, the network also constructs a data dissemination (routing) path from the sensor monitoring the target to the sink. If some other sensor failure creates a network partition, moving S2 can fix the routing problem as shown in Figure 1(b).



(a) (b)

Figure 1: A generic sensor network with mobile sensors to satisfy different mission requirements

From this example, we can see that mobility may significantly increase the capability of the sensor network by making it resilient to failures, reactive to events, and able to support disparate missions with a common set of sensors. However, there has not been a great deal of work on adding mobility to sensor networks, referred to as mobile sensor networks. Mobile sensor network is different from ad hoc network. Although both support mobility, mobility in sensor networks may be controllable, and hence be used to help achieve the network's missions. That is, mobility may be "purposeful" instead of being treated as an uncontrollable external stimulus to which the ad-hoc communication network must respond.

In this chapter, we propose the following solutions to address purposeful mobility in mobile sensor networks.

- **Mobility assisted sensing:** As the mission changes, or to achieve the mission when the network condition (e.g., sensor failure) changes, some mobile sensors must be relocated. The network needs to locate redundant mobile sensors and derive a scenario to move them to the target location considering the computation complexity, movement distance, communication overhead, and the impact of the mobility on other concurrently running missions.

- **Mobility assisted data dissemination (routing):** When sensors move in reaction to an event, a mission change, or failure, they may create network partitions, undesirable routes, or cause other disruptions. It is a challenge to invoke sensor mobility to improve network communications considering issues such as network connectivity, energy consumption of both communications and movement, and the network lifetime.

- **Integrated mobility management for sensing and routing:** In addition to moving nodes to fulfill the sensing or communication requirements of a single mission, it is essential to analyze the impact of mobility on all missions sharing the sensors for sensing or communication, and it is a challenge to define utility functions that can capture the benefits of the movement from the perspective of all missions, and maximize the capability of the network.

The rest of the chapter is organized as follows. In the next section, we discuss previous work and our solution for mobility assisted sensing. Sections 3 presents our work on mobility assisted routing. In Section 4, we present a problem formulation and the proposed solution for mobility integration. Section 5 concludes the chapter.

2 Mobility Assisted Sensing

In this section, we first give a brief review of the related work on sensor deployment, and then present the challenges and the proposed solution on relocating mobile sensors. Finally, we present techniques to find coverage holes, which are used to invoke sensor relocation.

2.1 Related Work

Since mobility assisted sensing is still a new area, there is not much work in the literature. The closet work is sensor deployment. Previous work on sensor placement [8, 28, 27] largely addressed random and sequential deployment. In a pure random scheme, many more sensors are required than the optimal number to achieve high coverage of a target area. Due to the existence of wind and obstacles, some areas may never be covered no matter how many sensors are dropped. Moreover, during in-building toxic-leaks [13], chemical sensors must be placed inside a building from the outside. In these scenarios, it is necessary to make use of mobile sensors [36, 35], which can move to the correct places to provide the required coverage. Based on the work from [32], mobile sensors have already been a reality. Their mobile sensor prototype, called Robomote, is smaller than 0.000047m^3 at a cost of less than 150 dollars in parts. Robomote also has some capability of avoiding obstacles when moved to the designated location.

There have been some research efforts on deploying mobile sensors, but most of them are based on centralized approaches. The work in [38] assumes that a powerful cluster head is able to know the current location and determine the target location of the mobile sensors. However, such a central server may not be available in most cases, and this approach suffers from single point failure problem. Sensor deployment has also been addressed in the field of robotics [13], where sensors are deployed one by one, utilizing the location information obtained from the previous deployment. This method has strong assumptions on the initial sensor placement in order to guarantee the communication between the deployed and undeployed sensors, and it does not work in case of network partition. Since sensors are deployed one by one, the long deployment time can significantly increase the network initialization time.

In our previous work [36], assuming that all sensors are mobile, we proposed three distributed algorithms for controlling the movement of sensors that are initially randomly placed to get high coverage. The algorithms use Voronoi diagrams to detect coverage holes. In one algorithm, VOR, sensors migrate towards holes. In the second, VEC, sensors move away from each other to achieve a uniform distribution. In the third, Minimax, sensors move towards their local center. Although mobile sensors can be used to improve the sensing coverage, their costs may be high. To achieve a good balance between sensor cost and sensor coverage, we proposed a bidding protocol [35] to assist the movement of mobile sensors when a mix of mobile and static sensors are used. In this protocol, mobile sensors act as the hole healing server, the base price of whose service is the size of the hole generated if they leave. Static sensors detect coverage holes and bid mobile sensor based on the hole size. Mobile sensors accept the highest bid and move to heal the hole if the bid is larger than its base price. In this way, mobile sensors always move to heal the largest holes and increase the coverage.

2.2 The Challenges of Sensor Relocation

The motion capability of sensor nodes can also be used for purposes other than sensor deployment. For example, in case of a sensor failure or node malfunction, other sensors can move to replace the role of the failed node. As an event (i.e., fire, chemical spill, incoming target) occurs, more sensors should relocate to the area of the event to achieve a better coverage. Compared with sensor deployment, *sensor relocation* which relocates mobile sensors from one place to another place, has many challenges. First, sensor relocation has strict time constraint. Sensor deployment is done before the network is in use, but sensor relocation is on demand and should be finished in a short time. For example, if the sensor monitoring a security-sensitive area dies, another sensor should take the responsibility as soon as possible; otherwise, some security policy may be violated. Second, relocation should not affect other missions currently supported by the sensor network, which means that the relocation should minimize its effect on the current topology. Finally, since physical movement costs much more energy than computation and communication, the moving sensor may suffer. As some nodes die due to low battery power, other nodes need to move again and cost more power. To be fair to each sensor and to prolong the network lifetime, it is important to balance the tradeoffs between minimizing the total energy consumption and maximizing the minimum remaining energy of the mobile sensors. Sensor relocation has been mentioned in [3], which focuses on finding the target locations of the mobile sensors based on their current locations and the locations of the sensed events. However, they did not address the challenges of finding the relocation path under time, topology, and energy constraints.

Due to these new challenges, our deployment protocols [36, 35] cannot be directly applied for senor relocation. For example, if the area covered by a failed sensor does not have redundant sensors, moving neighbor sensors may create new holes in that area. To heal these new holes, more sensors need to be involved. This process continues until some area having redundant sensors is reached. During this process, sensors may move back and forth and waste lots of energy. Based on this observation, we propose to first find the location of the redundant sensors, and then design an efficient relocation schedule for them to move to the target area (destination).

2.3 Finding the Redundant Sensors

Using flooding to find the redundant sensors may significantly increase the message overhead. Techniques based on publisher/subscriber [10] are designed for distributed systems or wired networks, and may not be applied to sensor networks due to high overhead. To reduce the message overhead, solutions similar to TTDD [37] can be used. In TTDD, the target field is divided into grids, and each has a grid head. The grid head is responsible for disseminating the sensing data to other grid heads. To find the interested data, the sink floods the query, which will be served by the grid head that has the sensing data. Since the data needs to be flooded to the whole network, although only grid heads, it still has significant overhead.

We apply the quorum concept [6, 11, 4] to reduce the message overhead. A quorum is a set of grids, and any two quorums must have an intersection grid. If the grid with redundant sensors advertises to sensors in its quorum, any destination grid head can obtain this information by sending a request to the sensors in its quorum. A simple quorum can be constructed by choosing the grids in a row and a column. Suppose N is the number of grids in the network. By using this quorum based system, the message overhead can be reduced from $O(N)$ to $O(\sqrt{N})$ [6].

By organizing grids as quorums, each advertisement and each request can be sent to a quorum of grids. Due to the intersection property of quorums, there must be a grid which is the intersection of the advertisement and the request. The grid head will be able to match the request to the advertisement. A simple quorum can be constructed by choosing the nodes in a row and a column. Instead of flooding the network with advertisements or requests, the request and the advertisement are only sent to nodes in a row or column. For example, as shown in Figure 2, suppose grid (0,3) has redundant sensors, it only sends the advertisement to grids in a row ((0,3), (1,3), (2,3), (3,3), (4,3)) and a column ((0,4), (0,3), (0,2), (0,1), (0,0)). When grid (3,0) is looking for redundant sensors, it only needs to send a request to grids in a row ((0,0), (1,0), (2,0), (3,0), (4,0)) and a column ((3,4), (3,3), (3,2), (3,1), (3,0)). The intersection node (0,0) will be able to match the request to the advertisement. Suppose N is the number of grids in the network. By using this quorum based system, the message overhead can be reduced from $O(N)$ to $O(\sqrt{N})$. Although the message overhead is very low compared to flooding, we can further reduce the message overhead by observing the specialty of our problem.

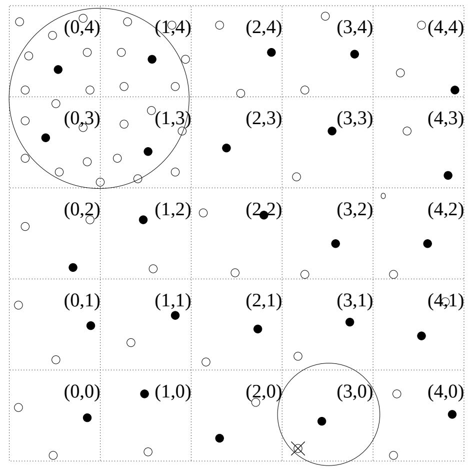


Figure 2: The system model

We can further reduce the message complexity by using the geographic information in sensor networks. For example, we can specify that an advertisement must be sent to grids in one column (advertisement quorum), and a request must be sent to grids in one row (request quorum). Since there is always an intersection grid between any column and row, the grid head of that intersection grid will be able to match the request to the advertisement. Still using the example of Figure 2, Grids (0,4), (1,4), (0,3) and (1,3) have redundant sensors, while grid (3,0) needs more sensors. The grid head of (1,3) propagates its redundant sensor information through its supply quorum ((1,4), (1,3), (1,2), (1,1), (1,0)). The grid head in grid (3,0) searches its demand quorum ((0,0), (1,0), (2,0), (3,0), (4,0)). Grid (1,0) can reply the information about redundant sensors. Compared to using the quorum in the last example, using grid-quorum cuts the message by half.

2.4 Relocating Sensors to the Target Location

After obtaining the information about where the redundant sensors are, the grid head needs to determine how to relocate them. At one extreme, sensors can move to the destination directly. Although this solution can minimize the moving distance, the redeployment time may be long especially when the destination is far away from the source. Furthermore, the sensor moved through a long distance may consume too much energy. If the sensor dies shortly after its movement, this movement is wasted and another sensor has to be found and relocated.

We use a *cascaded movement* to address the problem. The idea is to find some cascading (intermediate) sensors, and involve them into the relocation to reduce the delay and balance the power consumption. For example, as shown in Figure 3, assume S_0 fails and S_3 is the redundant sensor. S_3 can move to S_2, and S_2 moves to S_1, and S_1 moves to S_0. Since the sensors can first exchange communication messages (i.e., logically move), and ask all relevant sensors to (physically) move at the same time, the relocation time is much shorter. However, the total physical moving distance of this approach may be longer, and it is a challenge to make sure that the sensor coverage is maintained during the sensor movement.

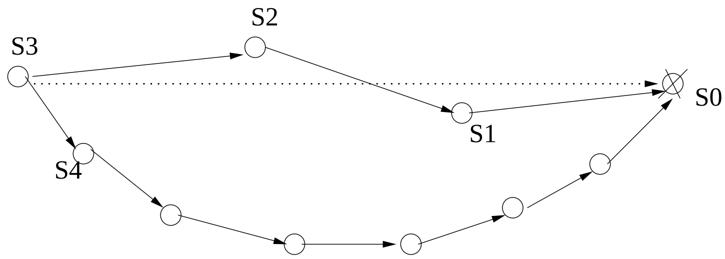


Figure 3: Sensor relocation

Generally speaking, we have three objectives when determining the relocation schedule: *minimize the relocation time, minimize the total energy consumption, and maximize the minimum remaining energy*. Relocation time is mission related and each cascading node has a time constraint. For example, as shown in Figure 3, for S_2, if it moves to S_1 before S_3 moves toward S_2, there may appear a new coverage hole around the area covered by S_2. Based on the mission requirement, this may or may not be allowed. From the energy point of view, maximizing the minimum remaining energy at all nodes after the relocation can prolong the network life time since no individual sensor is penalized, but there is a tradeoff between minimizing the total energy consumption and maximizing the minimum remaining energy, as illustrated by the two possible paths (going through S_2 and going through S_4) in Figure 3.

Previous work on power-aware routing addressed the tradeoff based on the current power level [19, 33]. When the remaining power is high, minimizing the total energy is more important; otherwise, maximizing the minimum remaining energy is more important since the overuse of individual sensor at this time may deplete their energy and consequently result in disconnections. To achieve a better tradeoff, Li *et. al* [23] designed a z-min algorithm, which calculates the route maximizing the minimum remaining power within those paths whose total energy consumption is less than z times the minimum energy consumption. However, our problem is much more complicated since cascading sensors also have time constraints.

One solution can be based on minimizing the difference between the total energy consumption and the minimum remaining power. Different from ad hoc routing protocols [19, 33], we may be able to minimize this difference when the sensors are regularly distributed. In this case, dynamic programming techniques can be used to minimize the difference between the total energy consumption and the minimum remaining energy, with the relocation time constraint.

To implement the scheduling algorithm in a distributed way, broadcasting can be used. Each possible cascading sensor broadcasts its decision of the next best sensor to the destination. A sensor can either wait for some amount of time (a system threshold) before broadcasting, or make its own decision and rebroadcast the updated version if the previous is wrong. To reduce the frequency of rebroadcasting, geographic information can be used. With this information, the sensors can be sequenced according to their distance to the redundant sensor. A sensor only broadcasts its decision after it receives the decisions from its neighbor sensors which are located in a search area, which has a high probability to encompass all the cascading nodes.

2.5 Sensor Network Diagnosis

In this section, we will develop techniques to detect coverage holes, which can be used to invoke the sensor relocation mechanisms.

2.5.1 Related Work

Exchanging messages between neighbors has the potential of quickly detecting a sensor failure, but may suffer from high false positive. Results in system level diagnosis has been used to address this problem in ad-hoc wireless networks [5]. The broadcast nature of the communication network has been exploited to efficiently implement a comparison-based diagnosis protocol. However, every node is eventually notified of the status of every other node, which is not necessary in sensor networks. A less complex approach uses two tier time out values [7]. The shorter time out is used to suspect failure of neighboring nodes, while a longer time out is used to reduce false positives with input from other neighbors. These techniques fundamentally rely on the participation of all neighbors. If some neighbors of the active node are in the low-power sleep mode, these protocols may not work well.

Low overhead failure detection can be achieved when the topology information is available. In a protocol by Staddon *et al.* [16], a base station continuously learns the topology of the network. It periodically probes nodes along a pre-established tree structure. If a subtree fails to send back a response, the root node of the subtree is determined to be dead. The status of its children in the subtree is obtained by routing additional probe messages around the dead node. This protocol depends on continuous update of the topology information. Therefore, it may not work well with missions where notification is sent to the sink infrequently, for example only when an exception occurs.

2.5.2 Techniques for Sensor Network Diagnosis

The techniques mentioned in related works do not take dynamic mission requirements into account, and focus mainly on the status of the sensors. In reality, the existence of a coverage hole depends on both sensor status and mission requirement. We observe that mission requirement change is initiated by the sink (or other control nodes) and the change can be significant. On the other hand, change in sensor status is a local event and typically has localized impact. We therefore believe two different approaches are necessary: a coverage hole estimation algorithm initiated by the sink whenever mission requirement changes, and a sensor status monitoring algorithm that executes continuously but dynamically adapts to mission requirement changes.

Coverage Hole Estimation: We first examine how to quickly estimate the existence and locations of coverage holes given a mission requirement. Each sensor is assumed to know its own location through some localization services, as well as its sensing areas for each parameter of interest to the mission. One important factor to consider is the detection speed. As mobile sensors have finite speed, early detection is critical. Another criteria is false positive rate; that is, if a coverage hole is detected, there should be a high probability that the reported hole actually exists.

The detection of a coverage hole relies on aggregate information from multiple sensors. One naive solution is to have the sink collect information from every sensor and perform a local check. This technique can create communication "hot spots" around the sink and therefore is not desirable. More importantly, a false positive may occur if the coverage information from a sensor is lost. We use a *mission directed data aggregation* technique to address this problem, where the mission requirement information is used to achieve effective data aggregation. Clearly, the sink can partition the network into a set of continuous non-overlapping areas satisfying the following property: if two areas are adjacent, the parameters being measured in these areas cannot be identical. This information can be used to build an in-network aggregation structure and to efficiently aggregate data from individual sensors. One observation is that we can exploit the sensing result difference of one particular parameter, especially when sensors are identical. Suppose two parameters p_1 and p_2, need to be monitored in the same area A. The maximum sensing distance of a sensor is d_1 for p_1, and d_2 for p_2, with $d_1 < d_2$. If we can determine there is no coverage hole in A when p_1 is measured, we can safely conclude that there is no coverage hole for p_2 either. Similarly, if there is a coverage hole in A when p_2 is measured, it is guaranteed that coverage hole will exist when p_1 is measured.

A promising approach to reduce false positive stems from the following observations: coverage holes and covered areas are non-overlapping areas, and continuous areas can be effectively aggregated. We can then estimate both coverage holes and covered areas during the same data collection and analysis process, with one of them being more aggressively estimated and the other more conservatively estimated. As a coverage hole and a covered area can not overlap, we can use the estimation on covered area to reduce false positives on coverage holes.

Sensor Status Analysis: The second approach is to devise a sensor diagnosis technique that can quickly detect a sensor failure and decide whether such failure will result in a new coverage hole. Deciding whether a sensor failure will result in a coverage hole requires information on both sensor status and mission requirements. This can be achieved by using in-network distributed diagnosis that captures distributed mission requirements as well as locally aggregated sensor status. Existing work on network diagnosis [22, 24] can be extended to develop protocols to construct such diagnosis structures, and to adapt it according to both mission requirements and sensor status change.

For a given sensor distribution and a set of mission requirements, not every sensor failure will result in a coverage hole. Therefore, the diagnosis technique will concentrate on sensors whose failure result in coverage holes. This technique can result in significant savings in terms of communication and computation overhead.

Co-Design Issues: Although these two techniques described above are inherently different, they are used to solve the same problem and need to co-exist in the same network. A typical scenario would be as follows. The solution to coverage hole estimation will be executed by the sink whenever the mission requirement changes; the sensor diagnosis technique is executed continuously, but should dynamically adapt to the area of interest to the mission. The coverage hole estimation algorithm needs to collect and analyze information from individual sensors and estimate the coverage holes. During this process, the protocol may need to maintain a certain in-network data structure to perform effective data filtering and aggregation. It is conceivable such infrastructure could be utilized by the sensor diagnosis protocol to quickly adapt its diagnosis and estimation behavior.

3 Mobility Assisted Routing

In this section, we focus on issues related to mobility for routing. When considering routing, sensors may be cast in one of three roles. First, sensors may be acting solely as relays to transfer data from a source to a sink. In this case they will move to form an energy efficient route. Second, sensors may be acting solely to gather data, and therefore their movement is dictated by their sensing requirements as discussed in Section 2. In this case, the routing protocols do not control mobility, but must react to it. Third, sensors may be assigned both sensing and relaying responsibilities, simultaneously. In this case, mobility must consider both sensing and routing. We discuss the first two cases in this section, and the third case in the next section.

In the first subsection below we discuss mobility algorithms that can minimize the energy consumed due to communication in the relay case, and extend it to a scenario in which the sensors are also assigned a sensing mission. In the second subsection we discuss priority based schemes designed to operate in the presence of high network volatility, the type of which may occur when sensors are highly mobile because of diverse sensing requirements.

3.1 Mobility for Routing Using a Distributed Annealing Strategy

Suppose that, as shown in Figure 1, certain sensor nodes are assigned target-tracking tasks while others are assigned tasks supporting communication, i.e., relaying the tracking data to the data-sinks of the network. For the relay nodes, the goals of mobility are to create routing paths between sources and sinks and to maximize the life-time of the network by moving to positions at which the required transmission power for the tracking data flows is minimized.

Although many routing protocols [19, 33, 34] for ad hoc networks can achieve similar goals, they are not optimized for a different environment that we consider here. Mobile sensor networks are a special kind of ad-hoc communication network in which all of the nodes have a communal mission. In some cases, mobility may be controllable and can, therefore, be used to help achieve the network's missions. That is, mobility may be "purposeful" instead of being treated as an uncontrollable external stimulus to which the ad-hoc communication network must respond. In other cases, sensor mobility may be random; e.g., a group of sensors diffusing through the air, or sensors moving to *scan* a large area [18], where scanning is a special case of mobility for sensing. Even when mobility is controllable, it may be desirable to make it partially random in order to deal with a lack of information locally due to the distributed nature of the network [20].

In this section, we propose a preliminary distributed/decentralized motion decision framework for the relay nodes based on the simulated annealing optimization algorithm (see, e.g., [21]) assuming that nodes can localize their proximal neighbors [26]. Our specific objective is to incrementally find the node positions that minimize the total required transmission power for all the active flows in the network while suitably penalizing for the energy cost of motion in order to find these positions, i.e., the mobility energy costs were amortized over the savings in communication power.

More specifically, let $V(x, r)$ be the total power required from the network to transmit the F flows using routes r when the intermediate nodes are in positions x; the optimal choice of routes at position x is

$$R(x) \quad \in \quad \arg\min_{r \in \mathbf{R}(x)} V(x, r)$$

where $\mathbf{R}(x)$ is the set of feasible routes connecting those nodes when in positions x. $R(x)$ is the objective of a distributed routing algorithm (c.f., the following subsections) operating at a much faster time-scale than that of the motion of the nodes. The amount of power required to maintain a link can be incorporated into the link metrics used for establishing routes.

Under a deterministic greedy mobility strategy, each node moves to a position at which it expects to minimize its energy costs for transmission. This approach may not achieve optimal energy efficiency because local minima may occur. To overcome this characteristic, we restrict motion to a lattice. Then under our annealing motion strategy, node k (currently at x_k) selects a neighboring position z, *at random*, and accepts the move to z according to a "heat bath" probability:

$$\min\{1, \exp(-\beta \Delta_k V(x, z))\}$$

where β is interpreted as inverse annealing temperature. Intuitively, the higher the "temperature," the more random motion that will occur. We chose a lattice over other random mechanisms because it tends to minimize the total energy costs in the network [20].

Given the β parameter, it is possible to tune this algorithm to match the requirements of the pure relaying scenario, or a scenario that includes sensors scanning and relaying simultaneously. When scanning, sensors will move throughout a field to gather information. In these cases, the random motion for routing may coincide with the motion for scanning. When assuming intermediate nodes are solely performing relaying tasks, and stationary nodes are the data sources and sinks, the annealing algorithm can be allowed to "cool" (β increased) to fix the relay nodes in optimal positions. However, if, for example, the tracking nodes themselves move, or the tracking tasking is dynamic, cooling would make the relay network less responsive to this change. Such change is part of more general "volatility" in networking conditions that may be experienced by the relay nodes. We discuss these cases in Section 4.

We conducted a simulation study on a mobile sensor network, see the results depicted in Figure 4. In one set of simulations, no scanning task is set, but in the other, a node moves to scan with equal probability that it makes an "annealing" move for the purposes of relaying data. The figures clearly indicate that tasking scanning resulted in increased power for communication and motion but increased scanning performance where the last figure represents the total number of points visited over a sixty second sliding time-window.

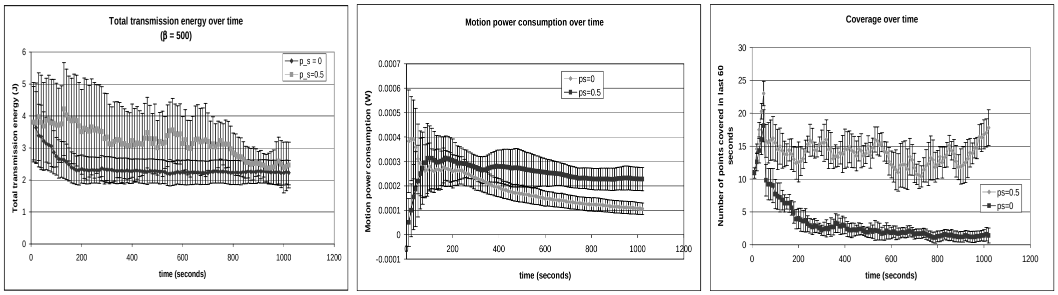


Figure 4: Communication Power, Motion Power and Scanning Coverage

3.2 Robust (Multipath) Priority Based Routing

It is natural to assume that because sensors are performing different missions, the data gathered from these sensors will have different requirements in terms of latency when being relayed through the network. Among the flows, high-priority flows include those for tracking traffic, responses to specific queries, the queries themselves, and control (routing) traffic. The goal of the routing protocol for these flows is to minimize energy consumption given a delay constraint. Lower-priority routing flows include passive surveillance traffic and tracking traffic for low-priority targets. The goal of the routing protocol for these flows is to minimize energy consumption. We develop priority based routing protocols that jointly manage both delay and energy concerns. The principle challenges of such protocols is to reliably route in a highly volatile topology with minimal overhead. Therefore, these protocols will be suitable for cases in which sensors are highly mobile to achieve sensing missions.

Our general approach will attempt to give primary importance to energy-efficiency by routing through a good energy path and use priority scheduling to reduce delay for the priority traffic. Non-priority traffic is not starved under the assumption that priority traffic is bursty and light. Nodes can be notified of the "true" energy resources and delay through each neighbor by a link capacity metric that incorporates the information about queue backlog (related to queuing delay via Little's formula).

We explore suitable routing algorithms based on both "swarm intelligence" and ant-colony meta-heuristics, e.g., Ant-Colony-Based Routing Algorithm (ARA) [12] and Termite [31]. ARA consists of three phases: route discovery, route maintenance, and route failure handling. In the route discovery phase, new routes between nodes are discovered with the use of a forward-and-backward ants, similar to AntNet. Routes are maintained by subsequent data packets, i.e., as the data traverse the network, node pheromone values are modified so that their paths are "reinforced." Also, as in nature, pheromone values decay with time in the absence of such reinforcement. Routing (link) failures, usually caused by node mobility, are detected through missing acknowledgments. When a node detects a routing error, the pheromone value associated with the "missing link" is set to 0. In [2], in addition to forward-and-backward ants, "uniform" ants are introduced to cope with highly mobile nodes.

Both energy and delay issues are considered in [29]. Only delay quantities, however, are considered when computing the pheromone values and forwarding probabilities. The dissipated energy of a node after each ant passes through is calculated by

$$\Delta E_{ij} = \frac{K}{(D_{ij})^2}$$

where K is the amount of energy to transmit the ant over a single unit distance, and D_{ij} is the Euclidean distance between node i and j[25]. The residual node energy at time t is computed by:

$$E_i(t) = E_i(t-1) - \sum_j \Delta E_{ij}$$

When a node's energy level simply drops below a pre-specified threshold value, the node is removed from the sensor network and alternative routes are found.

The proposed algorithm is based on the ANT mechanism algorithm, and uses energy and delay metrics to perform updates of pheromone levels. We modify the packet header to contain both energy and delay information so that a separate pheromone level will be maintained for each traffic type. Two types of pheromone-based routing algorithms will be developed. In the first framework, packet headers are assumed to have two fields used for routing: one to indicate bottleneck residual energy of a path (to be used for minimizing the energy costs) and the other being a hop count (to minimize delay). In the second framework, packet headers have fields that track the minimum residual energy of the nodes that relay them (as in the first algorithm), and fields that track the cumulative delay based on backlog information of queued packets destined to the packet's source. So, when a packet reaches its destination, it contains the minimum residual energy and the cumulative queuing delay of its route back to its destination.

To reiterate, such pheromone-based approaches [12, 31] are appropriate for highly volatile networking conditions. Such approaches do not exclusively use optimal routes but are highly responsive to changing network topology. Under more "stable" networking conditions, existing protocols like AODV [9] and DSR [17] would yield superior performance/overhead trade-offs, especially when the routing protocol is augmented with *planned* mobility information. Therefore, we will also consider multiprotocol routing in heterogeneous networking environments in which different regions of the network employ the most appropriate routing protocol under the circumstances.

4 Integrated Mobility for Sensing and Routing

In this section we formulate the problem for integrated mobility. Our goal is to maximize the value generated by the sensor network over time. For example, if multiple missions provide conflicting requirements to the network, higher value is placed on fulfilling the higher priority missions. Likewise, the longer a network remains active, the more value it will generate. Therefore, we develop algorithms and protocols to meet the needs of the composite of the highest value missions while maximizing network lifetime by conserving energy.

We define X_k as a vector representing the location of all sensor in the network during time period k, and t_k as the time that the network remains in this configuration. We define m_j as mission j. The value generated by each sensor i per unit time for configuration k is represented by $v_{i,j}^{k}(X_k)$. Further, we have

$$v_{i,j}^{k}(X_k) = u_c(X_k)s_i(m_j, X_k)$$

where $s_i(m_j, X_k)$ is the value of sensor i performing mission j while in configuration k, and $u_c(X_k)$ is either 1 or 0. $s_i(m_j, X_k)$, a function specific to each mission, will tend to be higher for more valuable missions when the sensors is optimally placed: as the sensor moves from its optimal position and its accuracy is compromised, or is assigned less critical missions, its value will decrease until it reaches 0. $u_c(X_k)$ is 1 if the sensor is able to communicate its data to the sink in a timely fashion; otherwise it is 0. This jointly captures the importance of sensing and communicating.

The overall value of the network for configuration k is

$$V_k = t_k(\sum_{i,j} v_{i,j}^{k}(X_k))$$

The overall value of a network is given by $V = \sum_{k=1}^{K} V_k$ where K is the total number of configurations over the lifetime T of the network. To maximize V, we must complete as many missions as possible, which implies conserving energy so that network lifetime is extended. The energy cost of configuration k is

$$C_k = M(X_{k-1} \rightarrow X_k) + E(X_k)t_k$$

where the first term is the cost of moving from configuration $k-1$ and the second is the cost of sensing and communicating in the new configuration k. Clearly, T and N depend on the C_k. We can see that there is a trade-off between the energy expended to realize a configuration, and the energy spent while in the configuration; a critical component to evaluating this trade-off is the time, t_k, spent in the configuration.

There are several interesting challenges to consider when designing algorithms to manage mobility to jointly accommodate sensing and routing. Algorithms that use strict priorities for sensing may not achieve maximum overall value, for example in cases in which the highest priority task requires exclusive use of sensors, thus allowing no other missions to be accomplished. Sequentially considering missions suffers from possible high latency for relocating sensors. Certain missions may be essential, i.e., they must be performed; this must be accounted for when designing algorithms. Algorithms must be carefully designed to account for the impact of location on sensing and relaying; for instance, optimal sensor placement for sensing for one mission may preclude communication, and hence completion of a second mission. When designing these algorithms, we must consider the mobility algorithms discussed in Sections 2 and 3, and possible extensions. For example, as discussed in Section 3, the “temperature” of the annealing algorithm may be modified to make sensors more or less reactive.

5 Conclusions

Adding mobility to sensor networks can significantly increase the capability of the sensor network by making it resilient to failures, reactive to events, and able to support disparate missions with a common set of sensors. However, there has not been a great deal of work on adding mobility to sensor networks. In this chapter, we addressed three closely intertwined issues to support mobility in sensor networks. First, we proposed solutions to relocate sensors in response to an event or failure, considering the computation complexity, movement distance, relocation time, communication overhead, and the impact of the mobility on other concurrently running missions. Second, we developed mobility assisted routing protocols to improve network communications considering issues such as network connectivity, energy consumption of both communications and movement, and the network lifetime. Finally, we defined utility functions that can capture the benefits of the movement from the perspective of all missions, and maximize the capability of the network.

References

[1] I. Akyildiz, W. Su, Y. Sankarasubramaniam, and E.Cayirci, "Wireless Sensor Networks: A Survey," *Computer Networks*, vol. 38, no. 4, March 2002.

[2] J. Baras and H. Mehta, "A probabilistic emergent routing algorithm for mobile ad hoc networks," in *Workshop on Modeling and Optimization in Mobile, Ad Hoc and Wireless Networks*, Sophia-Antipolis, France, Mar. 2003.

[3] Z. Butler and D. Rus, "Event-based motion control for mobile sensor networks," *IEEE Pervasive Computing*, vol. 2, no. 4, pp. 34–43, Oct-Dec 2003.

[4] G. Cao and M. Singhal, "A Delay-Optimal Quorum-Based Mutual Exclusion Algorithm for Distributed Systems," *IEEE Transactions on Parallel and Distributed Systems*, vol. 12, no. 12, pp. 1256–1268, Dec. 2001.

[5] S. Chessa and P. Santi, "Comparison based system-level fault diagnosis in ad-hoc networks," in *Proceedings of the 20th IEEE Symposium on Reliable Distributed Systems*, Oct. 2001.

[6] S.Y. Cheung, M.H. Ammar, and M. Ahamad, "The Grid Protocol: A high performance scheme for maintaining Replicated data," *IEEE Trans. knowl. Data Eng.*, June 1992.

[7] Chih-fan Hsin , Mingyan Liu, "A Distributed Monitoring Mechanism for Wireless Sensor Networks," in *Proceedings of the ACM workshop on Wireless security*, Sept. 2002, pp. p.57–66.

[8] T. Clouqueur, V. Phipatanasuphorn, P. Ramanathan and K. k. Saluja, "Sensor Deployment Strategy for Target Detection," *First ACM International Workshop on Wireless Sensor Networks and Applications*, 2002.

[9] S. Das, C. Perkins, and E. Royer, "Performance Comparison of Two On-Demand Routing Protocols for Ad Hoc Networks," *IEEE Infocom*, pp. 3–12, 2000.

[10] P. Eugster, P. Felber, R. Guerraoui, and A. Kermarrec, "The Many Faces of Publish/Subscribe," *ACM Computing Surveys*, vol. 35, no. 2, pp. 114–131, June 2003.

[11] H. Garcia and D. Barbara, "How to Assign Votes in a Distributed System," *Journal of ACM*, May 1985.

[12] M. Gunes, U. Sorges, and I. Bouazizi, "ARA - the ant-colony based routing algorithm for MANETs," in *Proceedings of International Conference on Parallel Processing Workshops(ICPPW)*, Vancouver, B.C., Canada, Aug. 2002.

[13] A. Howard, M. J. Mataric and G. S. Sukhatme, "Mobile Sensor Networks Deployment using Potential Fields: A Distributed, Scalable Solution to the Area Coverage Problem," *the 6th International Symposium on Distributed Autonomous Robotics Systems*, June 2002.

[14] C. Intanagonwiwat, R. Govindan, and D. Estrin, "Directed Diffusion: A Scalable and Robust Communication Paradigm for Sensor Networks," *Proceedings of the Sixth Annual ACM/IEEE International Conference on Mobile Computing and Networking (Mobicom)*, Aug 2000.

[15] J. M. Kahn, R.H. Katz, K.S.J. Pister, "Next Centry Challenges, Mobile Networking for Smart Dust," in *Proceeding of MOBICOM*, March 1999, pp. 271–278.

[16] J. Staddon, D. Balfanz, and G. Durfee, "Efficient Tracing of Failed Nodes in Sensor Networks," in *Proceedings of the First ACM International Workshop on Wireless Sensor Networks and Applications*, Sept. 2002, p. 122130.

[17] D. Johnson and D. Maltz, "Dynamic Source Routing in Ad Hoc Wireless Networks," *Mobile Computing, Kluwer*, pp. 153–181, 1996.

[18] J.M. Kahn, R.H. Katz, and K.S.J. Pister, "Mobile networking for smart dust," in *Proc. ACM/IEEE Int. Conf. on Mobile Computing and Networking, Seattle*, 1999.

[19] K. Kar, M. Kodialam, T. Lakshman, and L. Tassiulas, "Routing for Network Capacity Maximization in Energy-constrained Ad-hoc Networks," *IEEE infocom*, 2003.

[20] G. Kesidis and R. Rao, "Mobility management of ad-hoc sensor networks using distributed annealing," in *CSE Dept, PSU, Technical Report CSE03-017*, Sept. 2003.

[21] G. Kesidis and E. Wong, "Optimal acceptance probability for simulated annealing," *Stochastics and Stochastics Reports*, vol. Vol. 29, pp. pp. 221–226, 1990.

[22] L. Li, M. Thottan, B. Yao, S. Paul, "Distributed Network Monitoring with Bounded Link Utilization in IP Networks," in *Proceedings of IEEE Infocom*, Mar. 2003.

[23] Q. Li, J. Aslam and D. Rus, "Online Power-aware Routing in Wireless Ad-hoc Networks," *Proceedings of the Seventh Annual International Conference on Mobile Computing and Networking*, 2001.

[24] M. Thottan, Li Li, Bin Yao, Vahab S. Mirrokni and S. Paul, "Distributed Network Monitoring For Evolving IP Networks," in *Processdings of International Conference on Distributed Computing Systems*, Mar. 2004.

[25] M. Mauve, J. Widmer, and H. Hartenstein, "A survey on position-based routing in mobile ad hoc networks," *IEEE Network Magazine*, vol. 15, no. 6, pp. 30–39, Nov. 2001.

[26] M. Mauve, J. Widmer, and H. Hartenstein, "A survey on position-based routing in mobile ad-hoc networks," *IEEE Network Magazine*, vol. 15(6), pp. 30–39, Nov. 2001.

[27] S. Meguerdichian, F. Koushanfar, G. Qu and M. Potkonjak, "Exposure In Wireless Ad-Hoc Sensor Networks," *Mobicom*, 2001.

[28] S. Meguerdichian, F. Koushanfar, M. Potkonjak and M. B. Srivastav a, "Coverage Problems in Wireless Ad-hoc Sensor Network," *IEEE INFOCOM'01*, April 2001.

[29] R. Muraleedharan and L. Osadciw, "Balancing the performance of a sensor network using an ant system," in *37th Annual Conference on Information Sciences and Systems*, Baltimore, MD, USA, Mar. 2003.

[30] G. Pottie and W. Kaiser, "Wireless Integrated Network Sensors," *Communications of the ACM*, vol. 43, no. 5, May 2000.

[31] M. Roth and S. Wicker, "Termite: Emergent ad-hoc networking," in *Proceedings of the 2nd Mediterranean Workshop on Ad-Hoc Networks*, Mahdia, Tunisia, June 2003.

[32] G. Sibley, M. Rahimi, and G. Sukhatme, "Robomote: A Tiny Mobile Robot Platform for Large-Scale Ad-hoc Senosr Networks," *Proc. of Int'l Conf. on Robotics and Automation*, September 2002.

[33] S. Singh, M. Woo and C. S. Raghavendra, "Power-aware Routing in Mobile Ad Hoc Networks," *Proceedings of the 4th Annual International Conference on Mobile Computing and Networking*, 1998.

[34] C.-K. Toh, *Ad Hoc Mobile Wireless Networks*, Prentice Hall, 2002.

[35] G. Wang, G. Cao, and T. La Porta, "A Bidding Protocol for Deploying Mobile Sensors," *IEEE International Conference on Network Protocols (ICNP)*, pp. 315–324, November 2003.

[36] G. Wang, G. Cao, and T. La Porta, "Movement-Assisted Sensor Deployment," *IEEE INFOCOM*, March 2004.

[37] F. Ye, H. Luo, J. Cheng, S. Lu, and L. Zhang, "A Two-Tier Data Dissemination Model for Large-scale Wireless Sensor Networks," *ACM MOBICOM*, pp. 148–159, Sept. 2002.

[38] Y. Zou and K. Chakrabarty, "Sensor deployment and target localization based on virtual forces," *INFOCOM*, 2003.