Lightweight Sensing and Communication Protocols for Target Enumeration and Aggregation

Qing Fang*　　　　　Feng Zhao[†]　　　　　Leonidas Guibas[‡]

ABSTRACT

The development of lightweight sensing and communication protocols is a key requirement for designing resource constrained sensor networks. This paper introduces a set of efficient protocols and algorithms, DAM, EBAM, and EMLAM, for constructing and maintaining sensor aggregates that collectively monitor target activity in the environment. A sensor aggregate comprises those nodes in a network that satisfy a grouping predicate for a collaborative processing task. The parameters of the predicate depend on the task and its resource requirements. Since the foremost purpose of a sensor network is to selectively gather information about the environment, the formation of appropriate sensor aggregates is crucial for optimally allocating resources to sensing and communication tasks.

This paper makes minimal assumptions about node onboard processing and communication capabilities so as to allow possible implementations on resource-constrained hardware. Factors affecting protocol performance are discussed. The paper presents simulation results showing how the protocol performance varies as key network and task parameters are varied. It also provides probabilistic analyses of network behavior consistent with the simulation results. The protocols have been experimentally validated on a sensor network testbed comprising 25 Berkeley MICA sensor motes.

Categories and Subject Descriptors

C.2.1, C.2.2 [**Computer-Communication Networks**]: Network Architecture and Design – *Distributed Networks, Wireless Communications;* Network Protocols – *Application*

General Terms

Algorithms, Experimentation

Keywords

Distributed algorithms for ad hoc networks, processing and fusion of data in sensor networks, self-configuration in ad hoc networks, applications for ad hoc networks

1. INTRODUCTION

Networked embedded sensing is a promising technology for applications ranging from environmental monitoring to industrial asset management. Wireless sensor networks are characterized by limited onboard energy supply, as well as resources such as storage, communication and processing capabilities. Thus, the power of a sensor network lies in the ability of sensor nodes to pool resources together, optimally direct the resources to tasks at hand, and cooperatively gather and process information while satisfying strict resource constraints.

Riding on Moore's Law, we can expect sensor nodes to become smaller and cheaper, and thus networks of such nodes can be deployed on a much larger scale than was possible up to now. A key challenge we will be facing then is the design of a set of *lightweight*, extremely efficient sensing and communication protocols that use simple communication and low-resolution computation, such as comparing detection amplitudes of nearby sensors, to recover *qualitative information* about the environment or physical phenomenon of interest.

This paper addresses this challenge by developing a set of sensing and communication algorithms for a class of practically important problems for sensor networks: the detection and monitoring of *activity intensity*, for example, estimating the density or number of targets in a particular area, allocating and directing a corresponding amount of resources to the sensing problem. Our algorithms accomplish the recovery of qualitative information about the environment, such as the location of target signal peaks or the estimation of source power levels, without expensive signal processing. They also choreograph the efficient formation and maintenance of sensor groups, or aggregates, for collaborative processing tasks.

Consider, for example, the problem of tracking a moving herd of zebras in wildlife habitat management. An instance of a collaboration region is defined as the set of seismic sensor nodes that can potentially sense the movement of the animals, *i.e.*, those within the propagation range of vibrations from the animal footsteps. We call such a group of sensors an *aggregate*, as they collaboratively perform a specific task. For example, these sensors may collectively estimate the size of the herd from the intensity of the vibrations, or the speed at which the herd travels from the frequency of the signals. As the herd moves to the next region, a new aggregate of sensors will have to be defined, wake

---

* Department of Electrical Engineering, Stanford University, Stanford, CA 94305, E-mail: jqfang@stanford.edu

[†] Palo Alto Research Center (PARC), 3333 Coyote Hill Rd, Palo Alto, CA 94304, E-mail: fz@alum.mit.edu

[‡] Department of Computer Science, Stanford University, Stanford, CA 94305, E-mail: guibas@cs.stanford.edu

up, and start to track the animals, and so on. As one can see from this example, the definition of such collaboration regions depends dynamically on the task objectives and resource constraints. For example, some sensors may be on critical routing paths and their energy reserve is more likely to be quickly depleted than that of others; thus in forming an aggregate these sensors should participate only when the expected gain exceeds a threshold. Moreover, the collaboration regions are dynamically formed and updated, as the physical events of interest, environmental conditions, or network topology change. The definition and maintenance of collaboration regions adaptively is one of the key resource management tasks in sensor network operation.

2. OVERVIEW

This paper develops a number of scalable and efficient protocols and algorithms for low-cost, low-resolution wireless sensor nodes. The first is a lightweight protocol, *Distributed Aggregate Management* (DAM), for forming sensor aggregates for a target monitoring task. The protocol comprises a decision predicate *P* for each node to decide if it should participate in an aggregate and a message exchange scheme *M* about how the grouping predicate is applied to nodes. A node determines if it belongs to an aggregate based on the result of applying the predicate to the data of the node as well as information from other nodes. Aggregates are formed when the process eventually converges. The protocol is developed to support a representative collaborative signal processing task in sensor networks — monitoring distinct targets in a sensor field. Sensors are clustered around each target, or signal peak, using a "downhill" protocol. Sensor aggregates defined by multiple interfering targets are also considered.

However, as targets or other physical phenomena, move and pass by each other in the sensor field, information about the target density may be lost because of the possible under-sampling of the signal field and the lack of target movement history in DAM. For example, three targets may come together, resulting in a single sensor cluster by DAM. When the three targets split into two groups and move apart, at some point two distinct sensor clusters will be formed. However, without possible use of history or signal source information, the sensor network will not be able to determine which sensor cluster corresponds to which target group (the one target or the two targets), and thus be unable to allocate an appropriate amount of resources to the monitoring tasks.

The second algorithm, *Energy-Based Activity Monitoring* (EBAM), solves this problem by considering the energy level of target signal sources. The energy level is estimated by computing the signal impact area, combining a weighted form of the detected target energy at each "impacted" sensor. A nice property of this algorithm is its relative insensitivity to noise and disturbances or perturbation of node positions. However, to convert the energy level into the corresponding target density, we need to assume roughly constant source energy output for the targets. When arbitrary target source amplitudes are allowed there are intrinsically ambiguous situations for target counting. Without source separation, we cannot tell apart three targets of strength 2 at one location, vs. two targets of strength 3 at the same location. Such ambiguities cannot be present when all targets strengths are roughly the same (the case we discuss in EBAM), but also when all target signal strengths are very different.

The third algorithm, *Expectation-Maximization Like Activity Monitoring* (EMLAM), removes the constant and equal target energy level assumption. EMLAM estimates the target positions and signal energy using received signals, and uses the resulting estimates to predict how signals from the targets may be mixed at each sensor. This process is iterated, until the estimate is sufficiently good.

This paper focuses on defining and maintaining sensor clusters for in-network collaborative processing tasks. While the paper does not explicitly address the query processing and associated routing issues, the basic cluster structure computed by our protocols can be used to efficiently route and optimize queries. For example, the tree structure rooted at the leader node of each sensor node cluster in the DAM protocol can be used to aggregate sensor information. By additional embedding of a space indexing tree (such as a quad tree) and aggregating target totals for each internal node of the tree, we can support efficient query of the network for information such as "how many targets are present in an X-Y geographical region?" By exploiting the hierarchical structure of the indexing tree, we can make the cost of such queries roughly proportional to the number of nodes near the perimeter of the region, rather than the number of nodes inside. Similar ideas for supporting half-space queries were discussed in [4].

We have introduced noise in simulations to study the robustness of our algorithms. However, we have not systematically considered the effect of packet loss on the protocol performance, which remains a topic of future study. Our implementation of the DAM protocol on wireless sensor nodes, the Berkeley MICA motes, did experimentally validate the protocol and worked well under a moderate amount of packet loss.

3. RELATED WORK

Directed diffusion [7] is an effective mechanism for coordinating information transport in sensor networks. Diffusion uses a fine-grain data-level publish and subscribe for data sources to advertise data attributes of signals they detect and for data sinks to express data attributes they are interested in. The data source attributes and data sink interest are propagated and meet throughout the network. Routing pathways between the sources and sinks are established as shortest paths in the network connectivity graph.

The DAM protocol can be considered as an example of the next-level up coordination mechanism that defines "coherent" regions of sensors in a network. Unlike directed diffusion where data attributes are first-class objects, DAM makes grouping predicates first-class entities. Directed diffusion forms data routing and aggregation paths, while DAM forms sensor aggregates defined by constraints from tasks, resources, or geometry of a space.

Geographic routing [8, 12] is a mechanism for routing data to a geographic region instead of a destination node specified by an address. The destination region must be specified either as a rectangle or other regular geometric object for computational reasons. DAM can form arbitrarily complex regions as long as the network topology permits. The resulting aggregates can be abstracted as geometric objects for use by geographic routing. On the other hand, geo-routing could implement the information exchange within sensor groups DAM. Our work is also related to information storage mechanisms in sensor networks, such as DIMENSIONS [3] and geographic hash tables [10].

Our work is closest to that of Madden et al. [9] in spirit. Madden et al. discusses the challenges for supporting SQL-style queries

such as AVERAGE, MIN, MAX for distributed sensor networks. A SQL-style database system supports an aggregation function and a grouping predicate. For example, the query "SELECT TRUNC(temp/10), AVERAGE(light) FROM sensors, GROUP BY TRUNC(temp/10), HAVING AVERAGE(light)>50", forms groups of sensors according to the temperature bins, computes average light for each group, and then excludes those groups with light values less than or equal to 50. However, many interesting collaborative signal processing applications must form aggregates specified not just by individual node data, but also by the relations between the data across nodes. In the target monitoring problem, for example, sensor nodes exchange and compare amplitude detection values in order to form groups belonging to each target. In other cases, relations between the data may be determined by the geometry of the sensor locations. As this example shows, the distinction between query processing and collaborative signal processing in sensor networks is not as clear as in centralized data warehousing applications. DAM supports this style of more general grouping by blending the generality of a query schema with problem-dependent in-network signal processing.

A number of approaches to collaborative signal processing have already started to address the formation and management of collaboration regions in several application contexts [6]. For example, to track moving vehicles on street, sensor groups are dynamically formed, with each group responsible for collecting and processing information about one vehicle [14]. By examining a number of such problems, our long-term aim is to abstract such collaboration patterns into a set of generic schemas to support a wide class of applications for sensor networks.
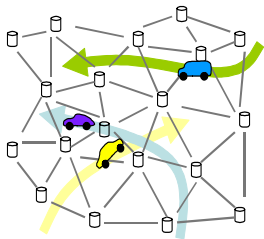
4. MONITORING SCENARIO

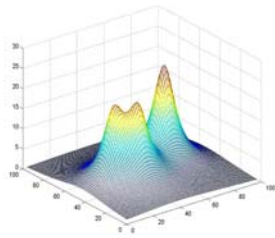**Figure 1.** Target monitoring scenario, showing three targets in a sensor field.

Figure 2. The corresponding target signal amplitude profile over the sensor field. The target counting becomes a peak counting problem.

We consider a task of counting multiple targets in a two-dimensional sensor field as shown in Figure 1. Targets can be stationary or moving at any time, independent of the states of the other targets. In addition, the following assumptions are made about this network:

1) Targets are point sources of (say, acoustic) signals. Target signal amplitude attenuates, as a monotonically decreasing function of the distance from the source, according to an inverse distance squared law (e.g., acoustic signal propagation in free space) or exponentially.

2) Each sensor has a finite sensing range. Sensors can only sense signal amplitude. Signals of multiple targets are summed at each sensor. This is a reasonable assumption for a mixture of acoustic signals from a set of non-coherent sources.

3) Each sensor can communicate wirelessly with other sensors within a fixed radius larger than the mean inter-node distance.

4) Sensors are time synchronized to a global clock.

5) Onboard battery power is the main limiting factor, as well as network bandwidth and latency, and onboard storage and processing.

The task here is to determine the number of targets and the approximate locations of target clusters in the field, forming an initial count and re-computing the count when targets move, enter, or leave the field. Although there may be different ways to solve this problem, we have taken the following approach. For each distinct target, a sensor leader is elected corresponding to the target. When multiple targets stay very close to each other, it is possible that only one leader is elected for this small group of targets, in which case, total power emitted from this group of target may be used in identifying number of targets in this small group. As targets move, new leaders are elected to reflect network changes. Therefore, we can obtain target count by determining the number of leaders elected and the number of targets in each cluster that each leader is elected from. The locations of the leaders and their cluster boundaries also provide information of the approximate locations of the targets. This paper focuses on the leader election process, postponing query processing and information gathering aspects for another study.

5. ALGORITHMS AND PROTOCOLS

Formally, a sensor network is represented as a graph $G(V, E)$, where V are the vertices representing sensor nodes and E edges representing one-hop connectivity in the network. Then, the *counting protocol* has the structure (G, T, P, M), with T the targets, P the grouping predicate, and M messaging schema, as defined earlier. The schema M applies P to nodes in the network, in a pre-specified manner, to compute sensor aggregates $A=\{V_1, …, V_n\}$, where $V_i \subset V$.

Property: The union of $V_1, …, V_n$ is not necessarily equal to V. Furthermore, V_i and V_j are not necessarily disjoint.

Examples: When targets are well separated, sensor aggregates for the targets become islands in the network. In other cases when the targets' influence regions are overlapping and each sensor in a region is able to separate signal components for each target (not assumed by the target counting problem studied in this paper), then the network could maintain overlapping sensor aggregates, one for each target.

The messaging schema M can apply P to nodes in several different ways, depending on the nature of the problem. When P is an equivalence relation on V, then A partitions V, and efficient algorithms for computing the partition already exist. M only needs to apply P to pairs of adjacent nodes in G in order to compute A. This may proceed in a relaxation style, propagating from node to node. Alternatively, one could first find those pairs of adjacent nodes that violate P, delete the corresponding edges in G, and return the graph components as the equivalence classes. For the target counting problem, when the targets' influence regions are overlapping and their boundaries are completely defined by saddle connections in the signal landscape, identifying the boundary nodes first can sometimes partition the nodes more efficiently. However, the protocol we will describe next uses grouping predicate that is not an equivalence relation, for reasons that will be discussed in a later section.

For a protocol to be applicable to large-scale sensor networks with diverse target characteristics (velocity, moving patterns, etc.) and limited network resources, it should have the following properties. First, it should be distributed in nature for scalability. Second, the leader election process should converge quickly to allow fast leader reelection for fast moving objects. Third, the protocol should require a minimal amount of inter-sensor communication, while keeping application semantics intact. Fourth, a reasonable level of fault tolerance should be supported.

As the sensors in this network can only sense amplitude, we need to examine the spatial characteristics of target signals when multiple targets are in close proximity of each other. In Figure 2, the 3-D surface represents target signal amplitude. Three targets are in the region, with two targets near each other and one target well separated from the rest of the group.

The 3-D surface shown in figure 2 forms a signal field landscape, where the height of a point is determined by strength of signal at that point. We can expect that a geometric analysis of this landscape can provide us with information about the signal sources − in this case the targets.

There are several interesting observations.

1)   When the targets' influence areas are well separated, the leader election can be considered as a clustering and cluster leader election problem. Otherwise, it approximates a peak counting problem.
2)   The target signal propagation model has a large impact on target "resolution". The faster the signal attenuates with distance from the source, the easier targets are to be differentiated from their neighbors based on signal amplitude they emit.
3)   Spacing of sensors is also critical in obtaining correct target count. Sensor density has to be high enough so that sampling of target signal amplitude provided by sensors can yield enough information for obtaining correct target counts. On the other hand, close proximity of sensors to each other makes for redundant measurements and wastes resources.

5.1 DAM: Distributed aggregate management
We develop DAM with target monitoring in mind, and with low cost amplitude sensing only sensors. For given signal field landscape such as that shown in figure 2, spatial sampling of this field by sensors at their locations only provides a sampled view of the surface. In this sampled view, we divide the sensors into clusters according to their sensed signal strength, so that there is only one peak per cluster. The purpose of the DAM protocol is to elect local cluster leaders. One peak may represent one target in case this target is well separated from the other targets. It can also represent multiple targets in case multiple targets are close together. Worst of all, it may represent no target in case the peak is generated by noise sources. A detailed description of DAM is given in this section. We will discuss ways to handle noise and target clustering issues in later sections.

Peaks are identified by means of comparing heights of neighboring sensors. To do this in a distributed fashion, information exchanges between neighboring sensors are necessary. If a sensor, after exchanging packets with all its one-hop neighbors, finds that it is higher than all its one-hop neighbors on the signal field landscape, it declares itself a leader.

At start of each protocol period, each sensor broadcasts its "qualification" (sensed signal strength) for leader election to its one-hop neighbors. Its "qualification" will be compared with information the neighbor has had so far, after the packet reaches a neighbor. If its "qualification" warrants the possibility of its becoming a leader, this packet will be passed to neighboring sensors of this neighbor, otherwise, the packet will be dropped. Dropping packets of no value at the earliest possible stage, while maintaining subsequent protocol semantics, serves the purpose of fast convergence of the protocol and minimizes the amount of inter-sensor communication.

Another important aspect of this protocol is enforcing locality. Locality is accomplished by enforcing the packet flow direction to be "downward only" on the signal field landscape. Figure 3 illustrates this idea in 1-D space. There are two clusters, cluster 1 and 2. In each of the clusters, packets can only be passed from sensor to sensor in a downward direction.
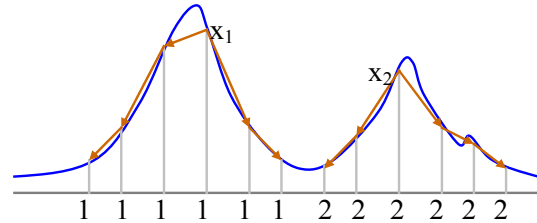

Figure 3. A sampled signal amplitude field. Neighboring nodes inform each other using a downward flooding algorithm to form equivalence classes of node groups corresponding to the support for each peak. Small disturbances can be handled using a δ–tolerance during amplitude comparison.

Before proceeding to the description of the protocol, it is necessary to make some definitions:

Definition 1: Neighbors of a sensor S refer to sensors that are within the transmitting radius of sensor S. *i.e,.* all the sensors that can "hear" sensor S directly.

Definition 2: *ThreshholdElection*, refers to the minimum signal amplitude a sensor has to receive from target(s) for it to participate in the leader election process. This value is up to protocol designer to decide. It must be no smaller than the sensor receiving threshold determined by the noise floor. It can effectively be used to find sensor cluster boundaries for a given *ThreshholdElection*, which is determined by the sensing task.

Definition 3: *Protocol period*, is the time duration of a leader election process. The leader election process runs every protocol period.

Definition 4: A DAM packet, is the only packet generated by a participating sensor each protocol period to broadcast its "qualifications" for leader election. A DAM packet includes the following fields:

MaxPr – Received signal power at packet originating sensor.
MaxID - ID of the packet originating sensor.
TransPr - Received signal power at sensor overlaying the packet.
TransID - ID of the sensor overlaying the packet.

In a sense, (*maxPr*, *maxID*) registers the "qualification" of the sensor that originates the packet, while (*transPr*, *transID*) registers the "qualification" of the sensor who passes on this packet to the sensor that examines these fields.

Definition 5: Sensor state, is a set of parameters a sensor keeps during each *protocol period* in order to process packets from other sensors to elect leaders. Sensor states include the following fields:

maxPrHeard – The highest *MaxPr* recorded in all the packets received by this sensor in current *protocol period*.

leaderID – ID of the cluster leader that this sensor belongs to.

myPr - Received signal power at this sensor.

myID - ID of this sensor.

myParent – ID of the parent node for this sensor.

participating - a boolean indicating if this sensor node currently participates in the leader election process. It is true when sensed signal power level exceeds *ThreshholdElection*. Otherwise, false.

Definition 6: The quantity δ is the threshold for the difference between *myPr* and *transPr* recorded in an incoming packet, beyond which the packet will be dropped. The aim for introducing this threshold is to overcome disturbances which may create lots of tiny spurious peaks on the signal landscape. Introduction of the δ threshold rule can overcome spurious noise with power up to δ above the *threshholdElection* level. More discussion on disturbance rejection can be found in section 6.3.1.

The purpose of the DAM protocol is to elect local cluster leaders and maintain local information about the signal landscape. DAM does not handle information forwarding to nodes querying the network. A leader is the sensor with maximal sensed signal power in its cluster. In other words, a leader is the local peak in the sampled view of our signal landscape. In the process of leader election, a tree structure is also formed with each sensor pointing to one of its one-hop neighbors as its parent, which is the sensor that overlays the packet from the leader to this sensor. Each cluster is thus a tree rooted at its leader and all paths from other nodes to the root are strictly ascending in the signal landscape.

1) At beginning of each protocol period

    If (myPr > threshholdElection) {
        participating = true;
        maxPrHeard = myPr;
        leaderID = myID;
        p = createMyPacket();
        p.maxPr = p.transPr = myPr;
        p.maxID = p.transID = myID;
        broadcast(p);}

2) When a packet *p* is received from a neighbor

    If (p.maxPr > maxPrHeard && p.transPr+δ> myPr) {
        maxPrHeard = p.maxPr;
        leaderID = p.maxID;   //leader at this stage
        myParent = p.transID;   //parent at this stage
        p.transPr = myPr;
        p.transID = myID;
        broadcast(p);}
    else
        drop(p);

The "downward only" semantics is expressed by logic:

If (p.transPr +$\delta \leq$ myPr)

    drop(p);

This protocol is synchronous. Because locality is enforced, packets from different clusters will be dropped before traveling far into another cluster. Meanwhile, because of the use of *ThreshholdElection*, only sensors with readings exceeding this value can participate in leader election process, thus limits packet

exchange to the geographical area around targets. This protocol scales well with size of the sensor network.

DAM provides a way to cluster sensors according to their sensed signal strength: each sensor joins the cluster defined by the highest peak that can reach that sensor through a strictly descending path in the landscape. Depending on inter-target distance, sensor spacing and signal propagation model, one leader may correspond to more than one target, i.e., there may be more than one target in each cluster formed at the end of each protocol period of DAM. Figure 4 shows a snapshot of a simulation run. There are 9 targets in the sensor field. There are 6 sensor clusters formed according to DAM. Some of the clusters indeed have more than one target.
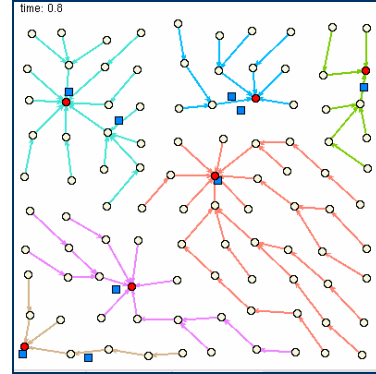

Figure 4. A snap shot of the sensor network we simulated. For the scenario shown above, there are 9 moving targets and 100 sensors in an area of 100m×100m. The 9 solid squares represent moving targets. Circles represent sensors. Solid circles signify that the sensors are elected leaders for the current protocol period. 6 leaders were elected, each with its cluster of sensors. Each cluster is a tree rooted at its elected leader.

5.2 EBAM: Energy-based activity monitoring

To address the under-counting problem, this section presents algorithms to estimate the number of targets within each cluster. After briefly explaining problems with signal energy measurement in a sensor network, we review the concept of Voronoi diagrams from geometric computing and then introduce algorithms and protocols for energy based activity monitoring. Impact of choices of system parameters on protocol performance is also discussed.

EBAM provides a solution to counting targets within each sensor cluster formed as a result of DAM protocol, with the assumption that all targets emit approximately equal amount of power, to be relaxed in Sec. 5.3, and this amount is known to the DAM protocol. The latter assumption could be removed by a network calibration protocol which we do not discuss in this paper.

5.2.1 The Energy Measurement Problem

The intuition of our approach is that, since the height of each point on our signal field landscape represents the signal power level sensed by a sensor at that location, the "volume" of each cluster over this landscape is associated with the total signal power in the cluster region. If we can estimate this "volume" in a meaningful way, we can relate sensor data to the total signal power in the region. Since the single target signal power is known, we can derive bounds on the number of targets from the total signal power sensed in the cluster.

Our sensors, however, can be placed in arbitrary and non-uniform patterns. What the sensed data represent is scattered measurements of the true signal field landscape. No matter which

signal propagation model fits the application in question, all propagation models share a common characteristics, which is that signal power attenuates quickly (attenuation factor 2~5) with distance from the source, especially at points within short distance from the source. This presents a major difficulty in estimating total signal power from scattered sampled data obtained by sensors. Figure 5 shows two signal power profiles for free space propagation model. We can see that attenuation rate is rather high within one meter from the source. Therefore, a little displacement of the sensor location relative to the source can cause the sensed signal power to vary drastically.
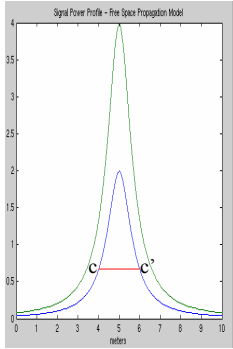

Figure 5. Two signal sources collocated at one point. Lower curve shows signal profile for a single source. Upper curve shows the profile of the superposed signal. CC' is the "*ceiling*".
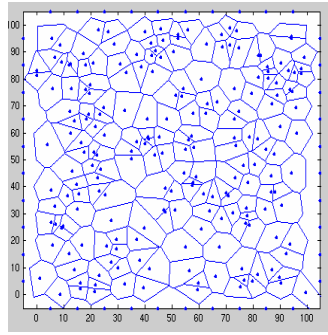
Figure 6. Voronoi cells for 200 sensors in a 100m×100m field. Sensors locations are chosen uniformly at random.

To solve this problem, we introduce a "*ceiling*" (CC' in figure 5) on the amount of power a sensor can contribute to the total signal power of the cluster, thus "desensitizing" the total cluster power value to slight changes of target locations. Therefore, the total power, *Pt*, calculated this way is an indicator related to the true "volume" under the signal landscape over the cluster region. The desired effect is that, when a single target moves across the network, *Pt* remains roughly invariant. It is immediately clear that the *ceiling* level cannot be too low, in which case *Pt* would become a constant factor away from the measurement of the impact area size. On the other hand, *Pt* cannot be too high, for otherwise, the location invariance property will be lost.

However, there is another difficulty in obtaining invariant *Pt* due to uneven sensor density. For example, given two identical signal sources, the first in an area with dense sensor distribution, the second in an area with sparse sensor distribution, it is almost certain that higher Pt will be obtained for the first signal source. This discrepancy arises from the fact that more sensors contribute to *Pt* in the first case. To take sensor density into consideration, we use the area of Voronoi cells associated with each sensor as weight in computing total *Pt* for each cluster.

5.2.2 Algorithms

5.2.2.1 Voronoi Cell
Given a set of points P = {p0, p1, p2, …} in the plane, one can partition the plane into Voronoi Cells (convex polygons) so that the Voronoi Cell about a point p in P consists of all points in the plane which are closer to p than any other point in P. What is of interest to us is that Voronoi Cell area can be used to capture local point density.

Figure 6 shows Voronoi cells for sensors in a sensor field with 200 sensors placed at locations picked uniformly at random.

5.2.2.2 Height of the "Ceiling"
We proceed now to examine the criteria for the choice of the height of the ceiling or, in other words, the upper bound for individual sensor contribution to *Pt*. A free space signal propagation model is used here. The signal profile generated by a single source looks like an elongated bell, as shown in figure 5. With the bell top removed at line cc', we get an elevated plateau with area πR^2, where $R=CC'/2$.

As discussed previously, the goal for having the *ceiling* is to construct an index *Pt*, which remains nearly invariant as a single target moves across different regions of the sensor field. Let us consider a scenario in which all sensors are placed at cross points of a grid with side length d. Figure 7 shows a bird's eye view of such a grid. Circles in this figure represent plateau areas formed by removing the top of the bell shaped signal surface generated by a single target.

Varying locations of the target, we obtain different values for *Pt*. We plot *Pt* against different *R* values for different target locations using MATLAB. The segment where all the lines intersect one another is the acceptable region of *R* for the given network configuration. Using this method, we obtained the relationship between *R* and d as $R = 0.38d \sim 0.45d$

Or in terms of area, $R = 0.38\sqrt{S} \sim 0.45\sqrt{S}$

where *S* is the constant Voronoi Cell size for all sensors. This result is also verified by simulation results. When sensors are not arranged on a grid, the first relation is no longer applicable. Experiments have shown that the second relation, however, still holds. In this case, *R* can be chosen first based on the known target signal power level. The above relation then gives the upper bound for the required cell size. The lower bound is no longer of concern with the use of Voronoi Cell size as weighting factors in obtaining *Pt*.

5.2.2.3 Sensor Density
As discussed above, Voronoi Cell size is closely related to the height of the *ceiling*. Average Voronoi Cell size is related to sensor density when sensors are arranged in an arbitrary fashion, which is the most likely case for real world applications. Average sensor density is an important parameter for the system designer. The total number of sensors and the field area determines average sensor density.

It is easy to understand that a denser sensor distribution provides higher resolution in sensing; however, it also increases system cost. In this section, we relate sensor density to Voronoi Cell size in a sensor field where sensors are dropped uniform at random onto this field. We give a performance bound through a mathematical analysis. Average performance is studied through simulations.

Consider a large sensor field with an area A, in which N sensors are uniform randomly dropped in this field, such that any sensor is equally likely to be at any point in the field independent of other sensors' locations. The sensor density in this area is,

$$\lambda = \frac{N}{A}$$

When λ is small and A is large, the number of targets in a sub-region with area s, has a Poisson(λs) distribution, where s << A.

For any sensor in A, on average, its Voronoi Cell will be a six-sided convex polygon. This is so because the Voronoi Diagram is dual to the Delaunay triangulation, and in any planar triangulation the average degree of a vertex is six or less [2]. Pick an arbitrary sensor Q in the field. Let r1, r2, …r6 be the distance from Q to the nearest sensor, second nearest sensor, and so on. If we assume that all the Voronoi cells in the field have at most 6 sides, which can easily be verified by inspecting figure 6, then

Voronoi cell size $\leq \pi r_6^2$.

Let n = number of points within distance r from Q. Then,

$$P(n \geq 6) = P(r_6 \leq r)$$

$$= 1 - e^{-\lambda \pi r^2}(1 + \lambda \pi r^2 + \frac{(\lambda \pi r^2)^2}{2!} + \frac{(\lambda \pi r^2)^3}{3!} + \frac{(\lambda \pi r^2)^4}{4!} + \frac{(\lambda \pi r^2)^5}{5!})$$

The exponential component is the dominating factor in this probability measure as long as the number of terms in the summation is finite. It shows that the lower bound probability increases exponentially with increasing sensor density. We performed some experiments to find out how sensor density relates to cell size requirement in a real system. A 100m×100m sensor field is used. Sensor density is varied with changing total number of sensors.
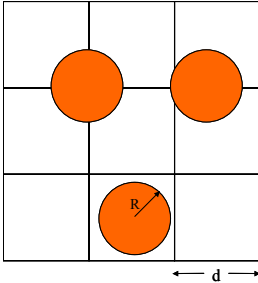

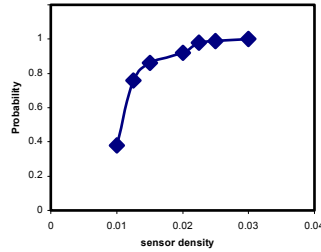**Figure 7.** Illustration of geometrical relations between ceiling area, grid spacing and target locations

F**igure 8.** Probability of Voronoi cell size not exceeding 100m^2 as average sensor density varies

Based on target signal power level and the chosen *ceiling*, 100m^2 is determined to be the upper bound for cell sizes. Figure 8 shows how the probability of each cell size not exceeding 100m^2 changes with different sensor density.

Apparently, to have all cell sizes not exceeding 100m^2, only 100 sensors are needed when they are arranged on square grid, while at least about 250 sensors are needed if we were drop them randomly in the field. We need many more sensors to provide a comparable level of protocol performance when sensors are randomly dropped in the field compared to the case in which sensors are placed on uniform grid.

5.2.3 Protocol

Our work on EBAM up to now has been focused on its computational correctness, which will be discussed in the simulation section. We intentionally ignored networking issues to experiment with the algorithm more efficiently. As for networking behavior, all EBAM requires is a way for obtaining P_t. It can be implemented in many different ways. We here give one possible solution.

First, let us define *ceiling* as a system parameter, which upper bounds the amount of contribution a sensor can make to the summation of all the sensor readings within the same cluster, namely P_t. It is the height of the ceiling as referred previously.

This protocol is built upon DAM protocol described earlier. A report packet is originated by each leaf sensor (that is a leaf on the tree representing the cluster) to report its reading (up to *ceiling)* to its parent. Its parent will merge the packets from its children (calculate subtotal of the reported readings from all its children) and send one report packet further up to its parent. This repeats until the packet reaches the root (leader). Each leader is responsible for calculating P_t and estimating number of target within its cluster.

Disturbance rejection in EBAM will be discussed in Sec. 6.3.2.

5.3 EMLAM: Expectation-maximization like activity monitoring

In the spirit of lightweight sensing, one of our principles in designing both DAM and EBAM is to keep the protocol as simple as possible. The protocols have been designed as simple periodic repetitions of set of operations based on data acquired in the current protocol period. Targets are counted in each period, independent of information from the previous period. However, given the spatial-temporal continuity of targets' movement, it is a logical next step to explore the possibility of exploiting this coherence in target movement. In this section, we introduce new algorithms for intra-cluster target counting using an expectation-maximization (EM) like technique, and analyze the validity of such an approach.

Assume that targets are not clustered together when they enter the sensor field. In other words, each leader node has one target associated with it as result of the DMA protocol sometime in the past. As targets move close to one another, by exchanging information among these sensor leaders, the number of targets can be tracked. However, a problem arises when targets move away from each other, i.e. clusters split. It is difficult to determine how many targets go into one part of a split cluster and how many go into another. We have tried to solve this problem using the algorithm introduced below.

5.3.1 Algorithm

The basic idea is that, if target location and amplitude can be estimated, then we have nearly complete knowledge of the signal field landscape instead of a sampled view of it. In the case where the free space propagation model applies, the set of estimation equations for determining location and target signal power can be solved by the Minimum Mean Square Estimation (MMSE) method [11]. However, when there are other targets nearby, signals emitted from these targets interfere with the estimation results. Consequently, MMSE can render unreliable estimation values. To fend off these interference effects, we apply the principle of expectation maximization in the estimation process. The algorithm is as follows:

- Assume at the beginning, targets are well separated.

- Use a prediction model to estimate new target position at each time step. The simplest prediction model is to let new positions equal to old positions.

- Using the old positions, calculate the percentages of the signal at each sensor node due to each target. Allocate the sensed signals to each target using these percentages.
- Use the separated signals for each target to perform MMSE for that target location.
- Repeat the above till convergence

In our implementation of the algorithm, each cluster leader performs MMSE based on sensed signal power at itself and at its neighbors. When two targets move close enough to each other such that they share one cluster leader, by passing information from the two previous leaders, the new leader is able to know that two clusters are merging into one, therefore two targets are in the region of its cluster. The new leader is then responsible for the joint estimation of the two targets.

5.3.2 Pros and Cons of This Method

This algorithm works very well when targets are reasonably separated. It yields rather accurate location and amplitude estimations for each moving target. The exact lower bound of distance between targets depends on target signal power levels and sensor density. However, when targets are within very close proximity of one another (in our simulation, half the sensor spacing distance), the interference effect derails the MMSE results. This is because the prediction model, as mentioned earlier, cannot be perfect. When interfering sources are very close, slight imperfections in the prediction of the current locations of targets can lead to wrong input data (to MMSE) and in turn to significant deviation from the true sensor readings for the target of interest. The consequence of these imperfect inputs and the fact that MMSE is quite sensitive to disturbances is that our estimation results will be inaccurate. Since the current target locations are used to predict target locations in the future, wrong estimation at current stage can throw the future estimation into doubt. This error propagation effect can eventually be very severe.

6. SIMULATIONS

In this section, we first evaluate the performance of the DAM protocol in NS-2 simulations. These simulations include detailed models of a wireless network MAC and physical layers. After verifying the correctness of DAM and measuring its performance, we then confirmed the viability of our design of EBAM with randomly distributed sensors and significant environment noise. In the interest of computational tractability, we do not model radio details in simulating the behavior of EBAM.

6.1 DAM Simulations

6.1.1 DAM Simulation Setup

By modeling the full MAC layer and physical layer, NS-2 allows evaluation of a system's performance on a bandwidth-limited, contention-prone wireless medium. Our simulations use a preamble based TDMA MAC and a 20-m radio range. We do not consider environmental noise in this part of the simulations, and leave it to section 6.3.

Two new protocol agents (target and sensor) were added at the transport layer. There are several issues related to the simulation setup. We chose TDMA as our MAC protocol for two reasons. First, TDMA is more energy efficient than CSMA based 802.11 MAC. Power consumption is not considered in our study at this stage. Should it be considered as a performance metric in the future, TDMA will certainly be more appropriate for power consumption estimation. Second, in our simulation, target signals are simulated by packet broadcast from each target. Sensing is simulated by sensors receiving these packets. To simulate signal superposition, we have each sensor add up receiving signal power (assuming target signal powers are independent) from all such packets. With the current preamble based TDMA (in NS-2), collision will never occur. This is exactly what we need because we want target signals to be summed up at a sensor instead of being dropped due to collision. We separate each protocol period into sensing and communication phases to avoid the simulation of sensing interfering with communications among sensors.

Free space propagation model is used in our simulation. The target movement pattern is generated with the mobility generator in ns-2. Sensor locations are generated with adding random perturbations in both x and y coordinates to a uniform grid. The random perturbation has a zero mean Gaussian distribution.

6.1.2 DAM Simulation Analysis

DAM is developed with a target monitoring task in mind. DAM provides us with information about target clustering and approximate locations of these clusters. The finer resolution the system can obtain in identifying each target, the more detailed information it provides on targets. In a perfect scenario, DAM can identify one sensor leader for each target in its closest proximity. To find the most important factors affecting system resolution in target monitoring, we first consider a simplified scenario in which sensors are arranged on a grid. We call this type of network non-jittered network. The second type is the jittered layout as described earlier. We first provide an analysis of the non-jittered network. We then present the simulation results of both types of networks and compare them with the mathematical results obtained. The comparison shows that the simulation results are qualitatively consistent with the mathematical analysis. The capability of the protocol to discern distinct targets decreases as the number of targets or average sensor spacing increases.

6.1.2.1 An Case Study for Non-jittered Network with Stationary Targets

We conducted an analysis on the performance of the protocol for the following scenario: Consider a large sensor field with an area A, in which sensors are arranged on a uniform square grid with distance between neighboring nodes being r. Sensors can communicate with their closest neighbors (4 such neighbors for each sensor, except for those on the border of the field). Consider randomly placing N targets in this field, so that any target is equally likely to be at any point in the area. There are N targets in the region. Target density is thus $\lambda = N/A$. When λ is small, the number of targets in a sub-region with area s, where s<<A, has a Poisson(λs) distribution.

Let T be any target, and d be the distance between T and its closest neighbor (another target). Following the property of Poisson distribution, we know that d has cumulative distribution function, $F(d) = 1 - e^{-\pi \lambda d^2}$.

For two target signal peaks to be differentiated, any communication pathway between the two leader nodes must contain a node whose sensed signal amplitude is lower than those of both leaders. Assuming a free space propagation model, from a

geometric analysis, we know that if d > 2.5r, the two neighboring targets are discernible as shown in Figure 9. In others words, this is a sufficient condition and the bound is reasonably tight.
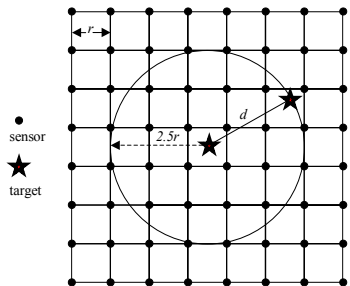

Figure 9. Inter-target distance has to be greater than a few multiples of r in order for the target peaks to be differentiated.

Based on this, the lower bound probability for a target being discernible from its nearest neighbor can be calculated as follows.

$$P(d > 2.5r) = e^{-\pi\lambda(2.5r)^2} = e^{-\pi(2.5r)^2 N/A}$$

This shows that the lower bound detection probability decreases exponentially with increasing number of targets for a given region. Likewise, the probability decreases rapidly as inter-sensor spacing increases. This reinforces our intuition that dense sensor distribution or sparse targets make targets much easier to be differentiated on the average.

When target mobility is considered, target distribution at any instant of time is no longer Poisson. We study relationships between protocol performance and target density, as well as sensor spacing through simulations.

6.1.2.2 Simulations Results
1) Simulations of Non-jittered Networks with Moving Targets

We are interested in the probability of obtaining correct count of targets by DAM alone from an empirical point of view. It would also be interesting to compare our protocol performance in non-jittered networks vs. that in jittered networks. To reduce the variations due to finite sample size, we ran a large number of experiments. The simulation results are shown in Table 1. Each data entry shown is the average of results from about 10 simulation runs, each of 1000 seconds duration (translating to a few 100s protocol periods and count estimate).

As expected, the performance of the protocol, as measured in terms of percentage of correct counts, decreases very rapidly as sensor spacing increases, and decrease quickly as the number of targets increases.

2) Simulations of Jittered Networks with Moving Targets

In this experiment, the sensor locations are generated by jittering the grid locations with a Gaussian noise, i.e., perturbing both x and y coordinates of a grid position with a zero-mean Gaussian, with a standard deviation of 0.1.

The simulation results are shown in Table 1. Note that a few entries near the lower-right corner of the table are not filled in, as they are practically close to zero. The performances for non-jittered and jittered networks are comparable.

Table 1. Performance data
Simulation I - percentage of correct counts for type I network
Simulation II - percentage of correct counts for type II network
λ - number of targets per sq. km

	λ	24.5	51	102	204
r =10	I	0.9312	0.8815	0.7525	0.4531
	II	0.9127	0.8775	0.7243	0.5060
r = 20	I	0.7575	0.6573	0.1302	0.02
	II	0.7190	0.6112	0.1488	0.03
r = 40	I	0.1311	0.01	-	-
	II	0.0991	0	-	-

It is also useful to analyze the effect of radio communication radius, relative to the mean inter-sensor distances, on the correctness of counting. Fixing the communication radius, as inter-sensor distance increases, neighboring sensor nodes may become unable to communicate directly with each other, resulting in possible over-count of targets. When inter-sensor distance decreases, more sensors can radio with each other, leading to possible under-count of close-by targets. The cause of this is the loss of "locality", and hence the resolution of the peak detection. Neither scenario would occur if the radio range could be adjusted so that only closest neighbors can communicate with each other. In general, over-count can be remedied with increasing sensor density or increasing radio range. Under-count can be remedied with decreasing radio range or use of EBAM.

6.2 EBAM Simulations
The detailed NS-2 simulations verify the correctness and robustness of DAM in a more realistic wireless environment. However, they were limited to system sizes on the order of 100 nodes. We now use less detailed simulations to verify our energy based activity monitoring algorithm in much larger systems. We built a special-purpose simulator that assumes packet delivery to neighboring nodes is instantaneous and error-free. This simulator is sufficient because our focus now is first on correctness of our method to obtain power invariant *Pt* with randomly placed sensors, and second on validity of introducing cell areas to factor sensor density into considerations.

6.2.1 EBAM Simulation Setup
We simulated networks with 100, 225, 400 sensors in a 100m×100m area. Sensors within 15m, 15m, 8m ranges are considered one-hop neighbors for each of the three network configurations respectively. EBAM is implemented on top of DAM. The following are system parameters used in our simulation: target signal power = 0.8mW, ceiling = 28μW, threshholdElection = 0.3652μW, and antenna gains = 1.

In our experiments, 9, 12, 20 moving targets in the sensor field are simulated. Target movement patterns are generated with the mobility generator in NS-2. We experimented with network configured such as all sensors are placed on uniform grid, as well as network configured such as each sensor is place at a spot picked uniformly at random and independent of the placement of all the other sensors. Figure 10 shows a sensor network with randomly placed sensors and clusters formed by DAM protocol. Performance metrics are percentage of time obtaining correct count of target and percentage of time missing the correct count by 10% and 20%.
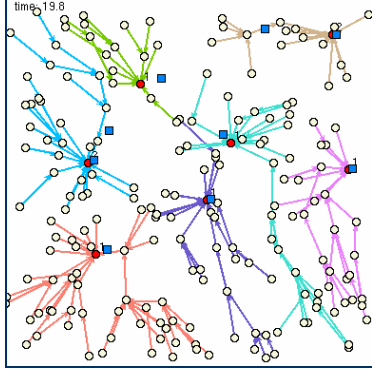
Figure 10. 200 sensors placed uniformly at random in a 100m×100m field. Clusters are formed according to the DAM protocol.

6.2.2 EBAM Simulation Results

In this section, we analyze simulation results for EBAM algorithm and compare performance of EBAM in network with sensors on grid and network with sensors placed uniformly at random. Results are presented in Table 2.

Table 2   Comparison of system with sensor distributed on grid and system with sensor distributed uniform randomly. First column shows percentage of time obtaining correct count (0% error). Second column shows percentage of time obtaining count with 10% error. Third column shows percentage of time obtaining count with 20% error.

	Grid			Random		
Error rate	0%	10%	20%	0%	10%	20%
100 sensor	57%	35%	8%	25.3%	30.1%	13%
225 sensor	61%	37%	2%	48.8%	35.1%	10%
400 sensor	74%	25%	1%	54%	40.2%	5.5%

From the above comparison, it is apparent that EBAM is robust in a network with total randomly distributed sensors. However, it does require many more sensors in this case to provide same level of count accuracy compared to networks with sensors placed on uniform grid. These results are expected because, as discussed in the algorithm section, what matters here is Voronoi Cell size. Smaller cell size leads to finer sampling of the 2-D signal surface as shown in figure 2, therefore, better system performance. When sensors are dropped into the field uniformly at random, it is likely that in some region sensor density will be low, consequently leads to inaccurate count when targets are in these regions.

6.3 Robustness to Disturbances

We've delayed detailed discussions of protocol robustness to environment noise and disturbance until this section because the major mechanism to counter external disturbance is inherent in EBAM. We will give a brief introduction to ways of handling minor disturbances in DAM and EBAM before presenting our simulation results and discussions on dealing with imperfect terrain conditions.

6.3.1 Noise Rejection in DAM

When random noise exceeds the known noise floor level whose impact is minimized with the use of *ThreshholdElection* discussed in DAM protocol, it will have an adverse effect on the protocol performance. The worst type of noise effect due to random environment noise is generating false peaks, which are treated as local leaders in DAM. Our strategy is to use a disturbance rejection threshold δ to test if a candidate is indeed a peak. For example, when a node's amplitude detection is less than δ above

transPr it just received, instead of dropping the packet it received, the node will continue to pass along the packet, thus effectively smoothing out the local spike due to disturbance or noise. The choice of the threshold δ depends on the noisiness of the environment and accuracy requirement of the task, and can be adjusted to reflect changing requirements or resource constraints.

6.3.2 Noise Rejection in EBAM

In EBAM, target count is derived from cluster summation P_t, with *ceiling* limiting contribution from each sensor. In signal processing terms, the summation process acts as a smoothing filter. White noise will have minimal impact after this smoothing operation. It is harder to handle non-white noise since smoothing operation is not as effective as it with white noise. However, the ceiling level at least can limit the degree of noise impact.

6.3.3 Simulation Results on Noise Rejection

We now demonstrate that EBAM is robust in the presence of white noise and DMA has certain resistance to disturbances with reasonably chosen threshold δ.
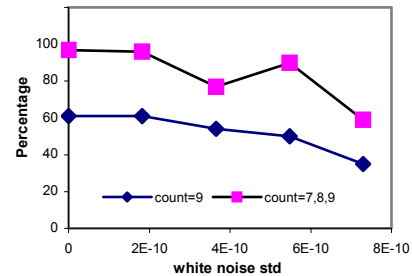

Figure 11. EBAM performance vs. standard deviation of white Guassian noise

We simulated 225 sensors arranged on a grid in a 100m×100m area. Other system parameters remained the same. White Guassian noise $N(0, \sigma)$ is used in this experiment. Maximal σ value is chosen to be twice as large as sensor *ThreshholdElection*. Recall that if random variable x has zero mean Guassian distribution, probability of x exceeds σ is about 15.9% (Q(1)=0.159, where Q(.) is the Q-function for normal distributions). This means that the noise we add to each sensor reading has 15.9% chance of larger than 2×*ThreshholdElection*, which is determined by the level of noise floor. We consider this reasonably models common environment noise.

It is evident in figure 11, that protocol performance degrades with increasing noise power, however in an acceptably moderate way. It is our belief that as long as only amplitude sensing sensors are used, noise will have a significant impact on system performance although good design may reduce this impact to some extent.

6.3.4 Handling Disturbances Due to Imperfect Terrain Conditions

Our study has focused on a 2-D signal propagation model, which also assumes that a sensor can "talk" to any sensor within its communication radius. In real world, it is likely that we are going to encounter situations that do not completely fit such a model.

For example, a scenario in which sensors are placed on an uneven surface may result in non-monotonic decrease in signal strength due to shadowing. This type of disturbances can be treated in a

similar way as that for noise rejection in DAM (section 6.3.1). Undulation of terrain surface results in weaker signal strength at lower points on the surface. This is equivalent to superposing noise of negative amplitudes on the 2-D propagation model at each point on the surface. An absolute lower threshold for elected leaders will also help to eliminate false peaks detected because of false valleys due to shadowing effect.

There may also be blocking objects which can interfere with communication between sensors. In this scenario, two sensors near one target would become two leaders because they could not communicate directly with each other. Dealing with this type of disturbances requires more involved techniques. One solution could be to discover obstacles by finding "communication holes" in the sensor network topology. Once these "communication holes" are identified, some virtual neighbors in terms of Euclidean distance can be identified. Running the proposed protocols taking into consideration of these virtual neighbors (instead of graph neighbors only) could then avoid over counting in this type of scenario. Finding such "communication holes" is part of our ongoing research.

7. EXPERIMENTAL VALIDATION

The DAM protocol has been implemented on a network of Berkeley MICA motes [5]. In our testbed, 25 wireless sensor nodes are laid out on a two-dimensional, jittered grid of 25×25 with a 10'' spacing (Fig.12). Each node in the network communicates wirelessly with 8 nearest neighbors. A base station, at the bottom of the sensor grid, retrieves the cluster/leader information from the nodes in the network through the multi-hop BLESS protocol, and displays it on the GUI of a PC (Fig. 13).
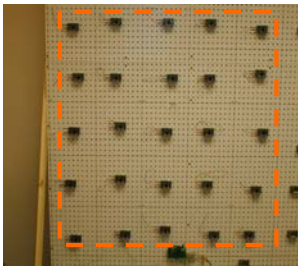

Figure12. A wireless sensor network testbed, built from Berkeley MICA motes, for experiments with the DAM protocol

Figure13. Sensing field (top) and GUI display (bottom) showing status of DAM leader election.

Light sensors were used for the experiment since the chosen target was a light source. The light sensors were chosen over the sound sensors due to their greater robustness to ambient noise (which in our case was harder to control for sound). The light source is produced with using a projector to image a MATLAB-generated movie of two moving spots onto the sensor board. The light intensity attenuation of each spot follows a 2-D Gaussian distribution, with the peak at the center of the spot.

Figure 14 shows two snapshots of the DAM output. One the left, the two targets are well-separated, with a distinct leader node elected for each target. On the right, the two targets are overlapping and result in under sampling of the signal field, with only one leader node elected for the targets. The results

experimentally validated the basic structure of the DAM protocol. Additional performance analysis of the protocol, through both simulation and testbed experiments, is a topic of future research.
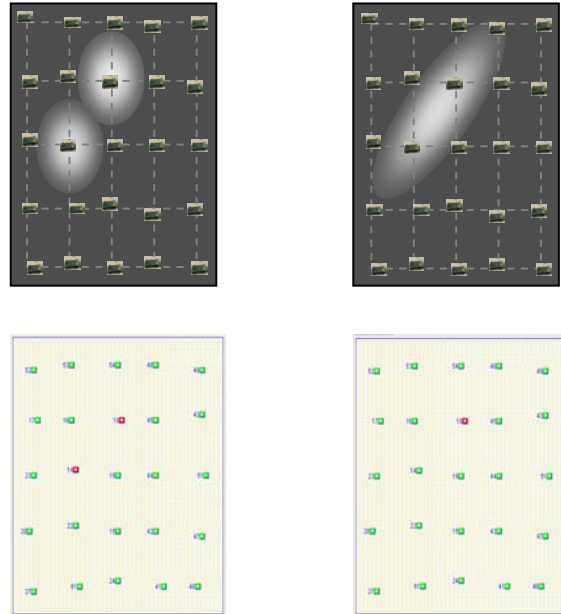

Figure 14. DAM protocol results. Left: leader node election for well-separated targets. Right: Leader election for overlapping targets.

8. DISCUSSION

We have intentionally kept our approach to the target counting problem as generic as possible, in order to bring out the central issue of aggregate management. For example, if two targets with overlapping influence regions have significant differences in signal signatures that the sensors can exploit, then the target counting problems can be simplified by using a classifier to determine the source of the signal components. However, this would require additional processing capabilities on the sensor node. By making minimal assumptions on the sensor node processing and communication capabilities, our protocols are more amendable to implementations on resource-limited hardware such as Berkeley motes [5].

An alternative to the counting protocol we discussed is for each node to locally determine if it is near a peak, using perhaps derivatives of the amplitude information. However, this local approach would be more sensitive to noise. The communication radius in DAM defines the minimal amplitude features the network can distinguish, and hence somewhat smoothes out small disturbances. Moreover, DAM computes sensor aggregates that can support additional collaborative processing tasks.

The target count information needs to be aggregated and extracted to answer a query [1]. Assume the user query originates from a node on the edge of the network. Our protocol needs to be extended to set up the query routing path(s) to appropriate regions, aggregate the results from each region, and route the query results to the node originating the query, all optimized with respect to constraints from tasks and resources. In our leader based protocol, the query aggregation is simply to aggregate the cluster leaders in the network. As pointed out in Sec. 2, the data we compute can be used to answer target counting queries for

geometric regions and not just for the entire domain, using a spatial indexing tree and pre-computing partial results. There is obviously a trade-off here that may be interesting to develop, where less preprocessing can be traded off against more sensing and communication at the time of the query. The optimal balance will depend on the cost of the preprocessing against the rate at which the queries come in.

For the moving target case, the correctness of the query aggregation would be more difficult to guarantee because of communication delays. If the communication is slower relative to the motion of the targets, stale peak detection might be aggregated, resulting in incorrect count. This remains as a future topic for us to address.

Computation and communication in terms of aggregates are not new. Spatial Aggregation Language (SAL), for example, explicitly treats aggregates as first class objects, and provides a set of operators to transform the aggregates [13]. The challenge for ad hoc wireless sensor networks is to develop a set of scalable, lightweight, resource-aware sensing and communication protocols that can efficiently form and maintain aggregates for collaborative processing tasks. The algorithms developed in this paper represent interesting instances of this class of new protocols and solve the nontrivial collaborative processing task of monitoring target activities. Our analysis and implementation results have already demonstrated the effectiveness of these algorithms.

9. ACKNOWLEDGMENTS

10. REFERENCES

[1] P. Bonnet, J. E. Gehrke, and P. Seshadri. "Querying the Physical World." IEEE Personal Comm., 7(5):10-15, October 2000.

[2] M. de Berg, M. van Kreveld, M. Overmars, and O. Schwarzkopf. Computational Geometry: Algorithms and Applications. 2nd Edition. Springer Verlag, 2000.

[3] D. Ganesan, D. Estrin, "DIMENSIONS: Why Do We Need A New Data Handling Architecture for Sensor Networks?" First workshop on Hot Topics in Networks, October 2002.

[4] L.J. Guibas, "Sensing, tracking, and reasoning with relations." IEEE Signal Processing Magazine, March 2002.

[5] J. Hill, R. Szewczyk, A. Woo, S. Hollar, D. Culler, and K. Pister. "System architecture directions for network sensors," in Proc. 9th International Conf. on Architectural Support for Programming Languages and Operating Systems (ASPLOS-IX) Cambridge, MA, Nov. 2000.

[6] IEEE Signal Processing Magazine special issue on Collaborative Signal and Information Processing for Microsensor Networks, S. Kumar, F. Zhao, D. Shepherd (eds.), vol. 19, no. 2, March 2002.

[7] C. Intanagonwiwat, R. Govindan, and D. Estrin, "Directed Diffusion: A Scalable and Robust Communication Paradigm for Sensor Networks," In Proc. 6th Int'l Con. on Mobile Computing and Networks (MobiCOM 2000), Boston, MA, Aug. 2000.

[8] B. Karp, H.T. Kung, GPSR: "Greedy Perimeter Stateless Routing Protocols for Wireless Networks." In Proc. 6th Int'l Con. on Mobile Computing and Networks (MobiCOM 2000), Boston, MA, Aug. 2000.

[9] S. Madden, R. Szewczyk, M.J. Franklin, and D. Culler. "Supporting aggregate queries over ad-hoc wireless sensor networks." In Proc. 4th IEEE Workshop on Mobile Computing Systems & Applications, June 2002.

[10] S. Ratnasamy, B. Karp, Li.Yin, F. Yu, D. Estrin, R. Govindan, S. Shenker, "GHT: A Geographic Hash Table for Data-Centric Storage", In Proc. 1st ACM Int'l Workshop on Wireless Sensor Networks and Applications, Sept. 2002.

[11] A. Savvides, C.C. Han, and M.B. Strivastava. "Dynamic Fine-Grained Localization in Ad-Hoc Networks of Sensors." In Proc. 7th Int'l Conf. on Mobile Computing and Networking (MobiCOM), 2001.

[12] Y. Yu, R. Govindan and D. Estrin, "Geographical and Energy Aware Routing: A Recursive Data Dissemination Protocol for Wireless Sensor Networks." UCLA Computer Science Department Technical Report UCLA/CSD-TR-01-0023, May 2001.

[13] F. Zhao, C. Bailey-Kellogg, and M. Fromherz, "Physics-Based Encapsulation in Embedded Software for Distributed Sensing and Control Applications." Proceedings of the IEEE, 91(1):40-63, Jan. 2003.

[14] F. Zhao, J. Shin, J. Reich, "Information-Driven Dynamic Sensor Collaboration for Tracking Applications." IEEE Signal Processing Magazine, March 2002.