

A Reduced Cover Approach to Energy Efficient Tracking using Wireless Sensor Networks

Mohamed K. Watfa** and Sesh Commuri

Abstract- In this paper, we present a novel approach for tracking a dynamic phenomenon. One of the central issues in sensor networks is energy efficient target tracking, where the goal is to monitor the path of a moving target using a minimum subset of sensor nodes while meeting the specified quality of service (QoS). Unlike other tracking methods that are based on computationally complex clustering techniques, our strategy is based on finding a reduced cover of the whole region and then subdividing the reduced cover into sub covers based on the target's location. The behavior of the proposed tracking algorithm is analyzed through simulation and the excellent performance is illustrated. We study the tradeoff involved in the energy efficient tracking of the target and compare the performance of our distributed tracking algorithms with other popular strategies. The gain in energy savings come at the expense of reduced quality of tracking. Our algorithms guarantee the robustness and accuracy of tracking as well as the extension of the overall system lifetime.

Keywords: *Ad-Hoc and Sensor Networks, Tracking, 3D Coverage, Energy Savings, Reduced Cover.*

I. INTRODUCTION

A Wireless Sensor Network (WSN) is a computational network of many, spatially distributed sensing devices that can be used to monitor conditions at different locations, such as temperature, sound, vibration, pressure, motion or pollutants. Usually these devices are small and inexpensive and are produced and deployed in large numbers, and their resources in terms of energy, memory, computational speed and bandwidth are severely constrained.

WSNs have been under development for many years and are gaining widespread use as technology improves, prices drop, and new applications are developed. "Microelectromechanical Systems (MEMS)" technology [1] made it possible to fit sensors into a smaller volume with more power and with less production costs. Smart disposable micro sensors can be deployed on the ground, in the air, under water, on bodies, in vehicles, and inside buildings. Sensor networks are playing an important role in bridging the gap between the physical world and the virtual information world [2].

Tracking, which involves identifying an object by its particular sensor signature and determining its path over a period of time, is one of the applications that can benefit from exploiting the characteristics of wireless sensor networks. The inherent parallelism of distributed sensors makes it possible to track multiple objects simultaneously, while the relatively low cost and ease of deployment enable the use of sensor network based tracking systems in remote or inaccessible locations, and when they need to be deployed on short notice. Algorithms for wireless sensor networks must have low communication overhead, rely as much as possible on local information, adapt to failures and changes in network conditions, and produce results in a timely fashion. Given the requirements to minimize the power, it is desirable to select the bare essential number of sensors dedicated for the task while all other sensors should preferably be in the hibernation or off state. Tracking mobile targets is an important application of sensor networks for both military and defense systems. Even though target tracking has been widely studied for sensor networks with large nodes and distributed tracking algorithms are available [3-7], tracking in ad hoc networks with micro sensors poses different challenges due to communication, processing and energy constraints. In particular, the sensors should collaborate and share data to exploit the benefits of sensor data fusion, but this should be done without sending data requests to and collecting data from all sensors, thus overloading the network and using up the energy supply.

Target tracking is considered a canonical application for wireless sensor networks, and work in this area has been motivated in large part by DARPA programs. Zhao et al. present the information driven sensor querying (IDSQ) mechanism in [6, 7]. IDSQ is a sensor-to-sensor leader handoff based scheme in which at any given time there is a leader sensor node which makes the decisions about which sensors should be selectively turned on in order to obtain the best information about the target. Liu et al.[8] develop a dual-space approach to tracking targets which also enables selective activation of sensors based on which nodes the

*Submitted to ICWN 06

**Corresponding author and Presenter

This work has been supported by U.S. Department of Defense, Army Research Office through grant # DAAD 19-03-1-0142.

Mohamed K. Watfa is with the School of Electrical and Computer Engineering at the University of Oklahoma in Norman, OK 73019. He can be reached at (405) 325-4302 (tel), (405) 325-3442 (fax); or at mwatfa@ou.edu (by email).

Sesh Commuri is an Associate Professor in the School of Electrical and Computer Engineering at the University of Oklahoma in Norman, OK 73019. His email id is scommuri@ou.edu

target is likely to approach next. Along these lines, Ramanathan et al. advocate a location-centric approach to performing collaborative sensing and target tracking in [9]. The idea is to develop programming abstractions that provide addressing and communication between localized geographic regions within the network rather than individual nodes. This makes localized selective-activation strategies simpler to implement. Brooks et al. present self-organized distributed target tracking techniques with prediction based on Pheromones, Bayesian, and Extended Kalman Filter techniques [10, 11]. The implementation and testing of a real distributed sensor network collaborative tracking algorithm in a military context is described in [12].

In this paper, results from our previous work [13-14] are used to develop a distributed tracking algorithm using wireless sensor networks. As shown in the simulation results, our algorithm outperforms the other tracking algorithms in terms of tracking error and energy efficiency. The rest of the paper is organized as follows. The challenges associated with tracking using wireless sensor networks are presented in section 2. The tracking problem is formulated in section 3. In section 4, an algorithm for tracking a dynamic phenomenon is addressed. Numerical simulation results that validate the proposed algorithms are presented in section 5 and the conclusions are summarized in section 6.

II. PERFORMANCE ISSUES

Though certain types of energy harvesting are conceivable, energy efficiency will be a key goal for the foreseeable future. This requirement pervades all aspects of the system's design, and drives most of the other requirements.

In target tracking using wireless sensor networks, an important requirement of the sensor network is that the required data be disseminated to the proper end users. In some cases, there are fairly strict time requirements on this communication. For example, the detection of a poisonous gas intrusion in a surveillance network should be immediately communicated to the authority so that action can be taken. We describe the various issues associated with sensor networks that need to be addressed by any protocol being developed for application in sensor networks. We will outline some key design challenges for any proposed tracking algorithm in the domain of wireless sensor networks:

1. *Large number of sensors:* Networks of 10,000 or even 100,000 nodes are envisioned, so scalability is a major issue. Nodes may fail and new nodes may join the network. In the light of target tracking, the coordination function should scale with the size of the network, the number of targets to be tracked.
2. *Low energy use:* Since in many applications the sensor nodes will be placed in a remote area, service of a node may not be possible. In this case, the lifetime of a node may be determined by the battery life, thereby requiring the minimization of energy expenditure.
3. *Network self-organization:* Given the large number of nodes and their potential placement in hostile locations, it is essential that the network be able to self-organize; manual configuration is not feasible. Individual nodes may become disconnected from the rest of the network, but a high degree of connectivity must be maintained. Sensor nodes should be capable of organizing themselves into a network and achieving the desired

objective in the absence of any human intervention or fixed patterns in the deployment.

4. *Collaborative signal processing:* The end goal is detection/estimation of some events of interest, and not just communications. To improve the detection/estimation performance, it is often quite useful to fuse data from multiple sensors.
5. *Distributed processing:* While a centralized architecture is theoretically optimal and also conceptually simple [5], it is not suitable in a large scale area because of the limited communication bandwidth of the wireless sensors. Moreover, failure of the fixed superior node may imply failure of the whole system.
6. *Tracking accuracy:* To be effective, the tracking system should be accurate and the likelihood of missing a target should be low.
7. *Computation and communication costs:* Any protocol being developed for sensor networks should keep in mind the costs associated with computations and communication. With current technology, the cost of computation locally is lower than that of communication in a power constrained scenario. As a consequence, emphasis should be put on minimizing the communication requirements.
8. *Uncertainty:* The exact positions of the nodes can not be known, so any position estimate of the target being tracked will be affected.
9. *Multi-modality sensor network:* The sensor have the abilities to sense the environment in various modalities , process the information , and forward it to a certain node for further processing. Compared to the single-modality sensor network that can only provide partial information of the environment, a multi-modality network can obtain more complete descriptions of the monitored environment through combining the fused data from various sensors with different capabilities and strengths.
10. *Time synchronization:* Time synchronization is a critical piece of infrastructure for any distributed system. Distributed, wireless sensor networks make particularly extensive use of synchronized time: for example, to integrate a time-series of proximity detections into a velocity estimate; to measure the time-of-flight of sound for localizing its source; to distribute a beamforming array; or to suppress redundant messages by recognizing that they describe duplicate detections of the same event by different sensors.

The impact of these performance issues on the design of a tracking algorithm is addressed in the next sections. A distributed energy efficient tracking algorithm is presented. The algorithm specifically aims at minimizing the number of active nodes necessary to track a dynamic phenomenon while achieving a high level of tracking accuracy.

III. PROBLEM STATEMENT

An emerging application area for sensor networks is intelligent surveillance and tracking. Sensors are randomly distributed in an area to be monitored. The ultimate goal is to detect, track and predict the movement of an intruder target and alert the sensors which are close to the predicted path of the target. However, minimizing the power consumed should be the most important design goal. Energy is a precious resource in sensor networks because every single unit wasted of energy directly effects the lifetime of the network.

In a tracking scenario, information obtained from nodes far away from the region of activity is of little or no use. For dense WSNs, information obtained from some sensors close to the region of activity might be redundant. An obvious way to save energy is to switch on only a subset of the sensor nodes. In most sensor activation strategies, energy savings come at the expense of a reduction in the quality of tracking. Depending on the information provided by a small subset of the sensor nodes would result in an increased uncertainty and thus the tracking accuracy would decline. In this paper, results from our previous work [13-14] will aid us in developing distributed accurate, and energy efficient tracking algorithms. Our algorithms are distributed since they don't depend on a centralized controller. They are energy efficient since they require that a small subset of sensor nodes to be active at a specific time. Finally, the algorithms are accurate since they activate a minimum subset of sensor nodes to guarantee an accurate estimation of the target location. The tracking algorithm is aimed at addressing the various challenges outlined in section 2.

We will start by defining the notion of a sensing region of a sensor node. Then, we provide definitions for both a reduced cover and a border cover. These definitions will aid us devise the distributed tracking algorithm.

Definition 1: Let $Y = \{y \in \mathbb{R}^3 / O_i(y) > \delta\}$. The **sensing region** of sensor S_i located at $X_i \in \mathbb{R}^3$ is defined as $A_i = \{y \in Y / \|y - X_i\| \leq R_s\}$, where $\|\cdot\|$ is the Euclidean distance between y and X_i .

In the case of 2D, the sensing region is assumed to be a disk of radius R_s . The sensing boundary (circle) of sensor S_i , in this case, is denoted by Cir_i .

Definition 2: A collection of sensor nodes $C_F = \{S_1, \dots, S_n\}$ is said to fully cover the region \mathbf{R} if and only if $\mathbf{R} \subseteq A_1 \cup A_2 \dots \cup A_n$.

Definition 3: A cover C_{RF} of \mathbf{R} with sensor nodes S_1, S_2, \dots, S_n each with sensing radius R_s and sensing regions A_1, A_2, \dots, A_n is **reduced** if no proper subset of C_{RF} is a cover of \mathbf{R} .

Definition 4: A set of sensor nodes C_{RB} is said to be a **reduced boundary cover** of a region \mathbf{R} if $\forall p \in B(\mathbf{R}), p \in S_i$ for some $S_i \in C_{Border, Reduced}$ and no proper subset of C_{RB} is a boundary cover of \mathbf{R} . i.e. $C_{RB} - S_i$, for any $S_i \in C_{RB}$ is not a boundary cover of \mathbf{R} .

Using definition 4, a region is said to be boundary covered if and only if an intruder is always detected as it crosses the border of the region.

Definition 5: A **boundary sensor node** S_b is a sensor node that is on the boundary cover. A sensor S_{nb} is said to be a **non-boundary sensor node** if $\forall p \in C_{nb}, p \in A_i$ for some $A_i \neq A_{nb}$ i.e. its border circle is completely covered by other sensors.

IV. DISTRIBUTED ALGORITHM

A. Algorithms

The objective of the intrusion detection algorithm is to achieve a static arrangement of the sensor nodes that minimizes the probability of undetected penetration of the intruder. The border coverage problem was studied in our previous work where we provided algorithms to select an optimal border cover of a rectangular region of interest.

We will start by proving that if we had a reduced cover of a region of interest, the border cover is a proper subset of the reduced cover.

A sensor S_i is not on the boundary of coverage (or coverage hole H) if and only if its sensing boundary circle Cir_i is completely covered by its neighboring sensors i.e. $S_i \notin B(H) \Leftrightarrow \forall p \in Cir_i, p \in A_j, \text{ for some } S_j \in S$.

Now, we can define the boundary of coverage (or coverage hole H) in terms of a set of sensor nodes.

Theorem: The set of border sensors are always a proper subset of the reduced cover set i.e. $C_{border} \subset C_{full}$

Proof: Suppose we have a redundant sensor S_x which will be deactivated according to the reduced cover algorithm. If we can prove that by deactivating S_x , none of the remaining non-border sensors will be transformed to border sensors then the final reduced cover set of the given region will always include the border cover set. Suppose S_x intersects sensors $S_y, S_z, S_w \dots$ and $S_y, S_z, S_w \dots$ are non-border sensors before S_x is deactivated. Since S_x is redundant, then the intersecting regions of S_x and $S_y, S_z, S_w \dots$ is clearly covered by other sensors. In figure xx, consider the intersecting Arc AB inside S_x (covered by S_x). Deactivating S_x , interior arc AB is still covered by S_x 's neighboring nodes. According to the definition of a non-border sensor, all the points on the border circle are covered and by deactivating S_x , all the points on the border circles are still covered. So, none of the non-border sensors are transformed to border sensors if a redundant sensor is deactivated so $C_{border} \subset C_{full}$.

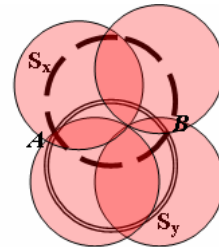


Figure 1: S_x (dashed circle) is a redundant sensor. Deactivating it won't transform its intersecting non border sensor S_y (double line circle) to a border sensor node.

In real world, a sensor network is completely asleep for a long time. When some interesting event happens, only a limited zone of the network that is close to the event is kept in its fully active state. The active zone should be centered at the current location of a target phenomenon that is being tracked; and, of course, the zone should move through the network along with the target. Nodes that are not within sensing range of the event are outside of the zone, and therefore do not waste energy. Optimally, the zone should move such that a phenomenon of interest is always kept inside of the zone. The zone is a circular region where the center of this zone is the border sensor node which had detected the target. The radius of the zone depends on two factors: the maximum speed of the intruder and the maximum time needed to calculate a reduced cover. The key to our algorithm is that there is no central controller i.e. each node will decide autonomously to be active or not in order to track the target.

The sensor nodes in the network can be in three different modes:

- 1-Full Active Mode: A node is capable of both sensing and communicating with neighboring nodes.
- 2-Light Active Mode: A node can only communicate with neighbors.
- 3-Sleep Mode: A node is inactive.

Our algorithms depend on the idea of "Divide and Conquer" which is basically selecting a reduced sensor cover for the region of interest, and if the phenomenon is moving, a new reduced sensor cover is established for the moving zone and so on. Every time the target is about to leave a zone, a border sensor node detects it and a new zone is created with the border sensor node as its center. A set of new sensor nodes within the circular zone and that belong to the reduced cover are activated. In order to save more energy, an enhancement to the algorithm would be adding prediction techniques where only a subset of the reduced cover nodes within the zone is activated depending on the predicted location of the target. Both approaches are discussed in details in the following sections.

Next, we will provide a number of algorithms that will aid us in developing a distributed tracking algorithm.

Algorithm 1 (*Reduced Cover Algorithm*)

PROBLEM

Given a dense deployment of sensor nodes, find a minimum subset of active nodes that guarantee full coverage of \mathbf{R} .

SOLUTION

Our works in [13, 14] indicate that a sensor node S_0 is completely covered if all the intersection points $C_i \cap C_j$ are covered by some sensor $S_l, l \neq i, j = 1..n$ where S_i and S_j are neighboring sensor nodes of S_0 and $C_i = A_0 \cap A_i$. Therefore, to check if S_0 is completely covered; one has to first find all the circles obtained by the intersection of $S_0 \cap S_k, k = 1..n$. For each C_k , find all the intersection points. If all these intersection points are covered by some

sensor node, then the circles C_k are covered which implies that S_0 is covered and can be deactivated.

The reduced cover algorithm is executed in the following way:

To check if S_i is covered, for every pair of nodes S_j, S_k in $N(i)$ do the following:

- a) Find C_{ij} the circle got by the intersection of the coverage surface of S_i, S_j .
- b) Find C_{ik} the intersection circle of S_i, S_k .
- c) Find the intersection points $C_{ij} \cap C_{ik}$.
- d) If the intersection points are all covered, i.e. $C_{ij} \cap C_{ik} \in A_l, S_l \in N(i), l \neq i, j, k$, then deactivate S_i .

It is well known that the coverage problem in WSNs is NP-hard. The computational complexity of the algorithm developed in this section is $O(N^3)$ where

$N = \left(\max_{i=1}^n |N(i)| \right)$ is the maximum number of nodes in the neighbor set of any sensor in the network.

Algorithm 2 (*Sub Reduced Cover Algorithm*)

PROBLEM

Given a reduced cover set of a region \mathbf{R} , deduce the reduced cover of a sub region \mathbf{R}_{sub} of \mathbf{R} .

SOLUTION

Each node $S_j(x_j, y_j, z_j)$ that is part of the reduced cover set will receive an ALLERT message that contains the coordinates of the center of the sub region $S_i(x_i, y_i, z_i)$ and the maximum speed of the intruder. S_j will check if it lies within the sub region i.e. it will check if $d(S_i, S_j) \leq V_{max} \cdot t_{max}$ and decides to be part of the reduced Sub Cover.

Algorithm 3 (*Boundary Cover Algorithm*)

PROBLEM

Given a reduced cover set of a region \mathbf{R} , deduce the border cover.

SOLUTION

Each node $S_j(x_j, y_j, z_j)$ that is part of the reduced cover set will receive an ALLERT message that contains the coordinates of the center of the sub region $S_i(x_i, y_i, z_i)$ and the maximum speed of the intruder. S_j will check if its sensing region intersects the boundary lines of the sub region i.e. it will check if $r_{sub} - R_s \leq d(S_i, S_j) \leq r_{sub} + R_s$ and decides to be part of the reduced Sub Cover.

Algorithm 4 (*Prediction Algorithm*)

In the linear prediction (LP) model, also known as the autoregressive (AR) model, the next location $X(n)$ is approximated by a linear combination of k past locations.

We are then looking for a vector 'a' of k coefficients, k being the order of the LP model. Provided that the 'a' is estimated, the predicted value is computed simply by FIR filtering of the k past samples with the coefficients using $X(n) = \sum_{i=1}^k a_i X(n-i)$. To keep the calculation simple

and the communication overhead low, the prediction model we use is only based on the target's moving speed and its direction of movement using the previous and current position of the target to predict the next location. The previous position of the target $X(t-a) = (x(t-a), y(t-a))$ and the current position of the target $X(t) = (x(t), y(t))$ are used to estimate the velocity and the direction of the movement. The velocity is given by $v = \frac{d(X(t), X(t-a))}{a}$ while the direction

is $\theta = \cos^{-1} \frac{x(t) - x(t-a)}{d(X(t), X(t-a))}$. The next position of the

target can be predicted by $x(t+a) = x(t) + vt \cos \theta$ and $y(t+a) = y(t) + vt \sin \theta$.

B. Tracking Algorithm

Energy efficient tracking of a target involves different steps:

Phase 1:

- Find a reduced cover of the region of interest.
- Deduce the border cover of the region of interest.

Phase 2:

- Detect the presence of the target.
- Broadcast the coordinates of the border sensor node and activate the necessary Sub reduced cover (Deduce the Sub border Cover).
- Move the sub region accordingly.

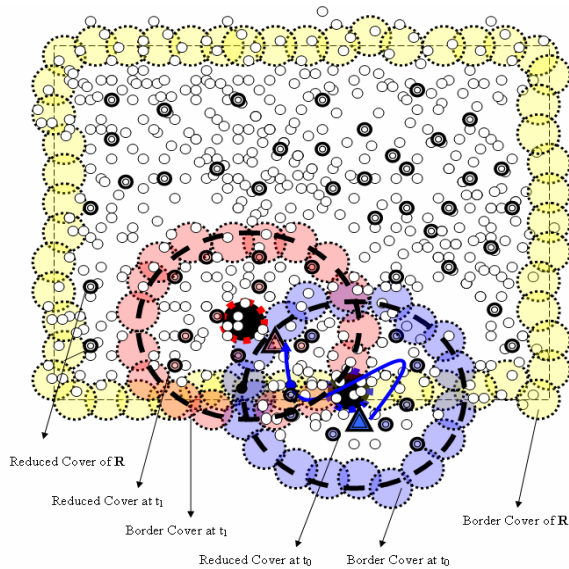


Figure 2: A snapshot of the distributed tracking algorithm in action.

The distributed tracking algorithm works by assigning a role for each sensor node. The initialization phase basically activates a border cover of the region of interest i.e. all the

sensor nodes on the border of the region of interest are active. When a target is detected by a particular border sensor node S_i , then S_i is selected as the center node and broadcasts its coordinates to the reduced cover nodes in order to activate a subset of the reduced cover that will cover the sub circular zone of center S_i . The new sub border cover is deduced and as soon as a border sensor node detects the intruder, the same steps are repeated. This procedure guarantees the tracking of the target at all times since the radius of the circular zone depends on the maximum speed of the target and on the maximum time it takes to form the reduced cover of the sub region. A snapshot of the tracking algorithm is depicted in Figure 2.

The flowchart of the processing performed at any given node S_i located at X_i to allow distributed target tracking is provided in Figure 3:

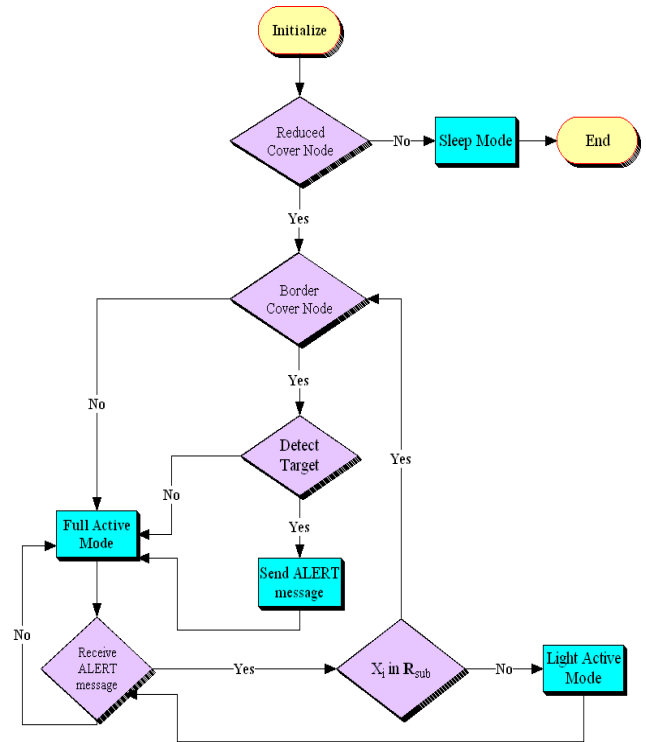


Figure 3: flowchart of the processing performed at any given node using the tracking algorithm.

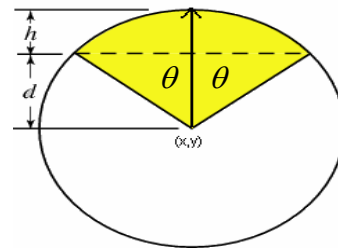


Figure 4: Circular sector of central angle 2θ .

B. Tracking Algorithm with prediction

Using the prediction algorithm, a sensor node estimates the targets next location moving with a velocity V and direction θ . Since the ultimate goal is conserve energy while achieving the necessary tracking performance, only a subset of the sensor nodes within a circle of radius $R=V.T$

and center X is activated. Since the direction of motion is known, any sensor node that belongs to the reduced cover and is within a circular sector of central angle $\alpha = 2\theta$ is activated. So, instead of activating all the sensor nodes that lie within a circle of radius R and center X , we only activate a subset of these sensor nodes that are with a circular sector of central angle 2θ where θ is the direction of the movement of the target (see Figure 4).

The distributed tracking algorithm could be further optimized by encapsulating prediction techniques (Algorithm 4) and is performed using the following steps:

Step 1: Border Sensor node S_i detects target.

Step 2: S_i broadcasts an ALERT message containing its location (x_i, y_i) , the maximum velocity V_{max} , and the predicted direction of the moving target θ .

Step 3: Any sensor node S_j that receives the ALERT message, checks if it belongs to the reduced cover of the circular zone of center S_i . If so, it checks if the Euclidean distance between S_i and S_j is less than the radius of the tracking circular zone R . If that is the case, it checks if it is within the circular sector of central angle 2θ . To do so, it checks if ϕ , the angle between the straight line connecting $S_i - S_j$ and the predicted direction of the target, is less than θ . If that is true, it decides to be active.

The flow chart of the processing at each sensor node using the predictive tracking algorithm is presented in Figure 5:

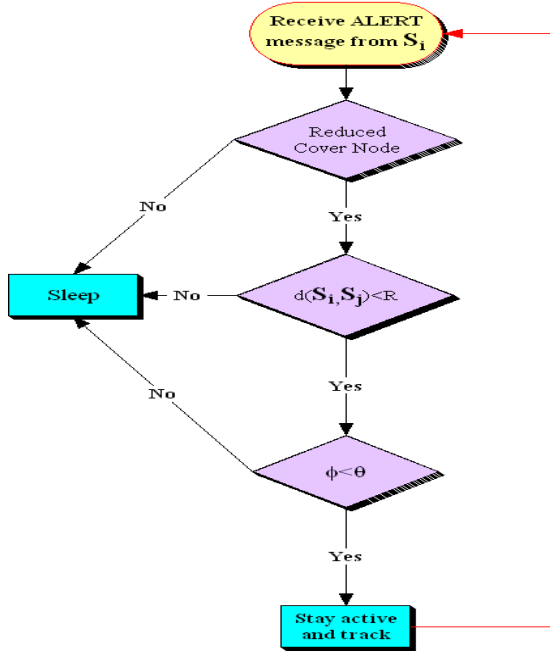


Figure 5: The flowchart of the processing at a sensor node using predictive reduced cover based tracking.

C. Performance Measures

We consider a sensor network consisting of n nodes deployed in some operational area, operating for a total time duration t . There is a single target moving through the area.

We assume that all sensors in the network are binary detectors with a fixed sensing range R_s . In other words, at each instant, each sensor returns a '1' if the target is present within a distance S of that sensor, and a '0' otherwise. Given this simple sensor model, we take the centroid of the locations of all detecting sensors as an estimate of the target's location at any given time t_i .

There are k sensors at locations $X_i(t_0) = (x_i(t_0), y_i(t_0))$ detecting the target at time t_0 . Then the estimated location of the target is $X_{Target}(t_0) = (x_{Target}(t_0), y_{Target}(t_0))$ where

$$x_{Target}(t_0) = \sum_{i=1}^k x_i(t_0), \quad y_{Target}(t_0) = \sum_{i=1}^k y_i(t_0)$$

The two performance measures of interest to us in evaluating different tracking strategies are the coverage life time of the whole system (energy expenditure), and a measure of the tracking quality (accuracy), which reflects the uncertainty in the target's location. The two metrics are presented next.

Performance Measure 1 (System Lifetime)

We evaluate the system life time. The metrics used in evaluating system lifetime is the *coverage lifetime* of the region to be monitored. The overall coverage lifetime is the continuous operational time of the system before the coverage drops below its specified threshold (for example 0.9). Assuming that each sensor node has a limited energy supply (300 Joules) and when it runs out of energy it is deactivated. The power consumption of Tx (transmit), Rx (receive), Idle and Sleeping modes are 1400mW, 1000mW, 830mW, 130mW respectively. As time passes, sensor nodes will be deactivated due to lack of energy and will leave some coverage holes in the border of the region.

Performance Measure 2 (Tracking Error)

We use the Euclidean distance between the estimated and actual locations of the target to measure the tracking error.

The tracking error at any given time t_i is:

$e(t_i) = d(X(t_i), X_{Target}(t_i))$ where $X(t_i)$ is the actual position of the target at time t_i and $X_{Target}(t_i)$ is the estimated position of target using the distributed tracking algorithm.

We denote the time average error over the total time t as

$$E = \frac{1}{t} \int_0^t e(t) dt .$$

V. SIMULATION RESULTS

In this section, the theoretical results in sections 3-5 are validated through experimental simulations. The metrics developed in section 4 are compared on different tracking strategies. The strategies that will be compared are:

- **Basic Strategy (BS):** In this strategy, all the sensor nodes are in full active mode. Obviously this strategy offers the worst in terms of total coverage life time. However, it offers the best results in terms of tracking accuracy

and thus serves as a baseline for comparison with other developed strategies.

- **Reduced Cover Strategy (RCS):** In this strategy, the cover redundant sensor nodes are deactivated using algorithm 1 and all the remaining sensor nodes that are in the reduced cover track the object.
- **Basic Selective Predictive Strategy (BSPS):** In this strategy, only a small subset of all the nodes is in tracking mode at any given point of time. They also predict the “next” position of the target and hand over tracking to nodes best placed to track the target in the “next” position. The rest of the nodes are in communication mode and can switch to tracking mode on being alerted by signals from tracking nodes. All the sensor nodes within a circle of specified radius centered on the predicted target location are in full active mode.
- **Reduced Cover Selective Tracking Strategy (RCSTS):** In this strategy, only a subset of the reduced cover is activated to guarantee that the target is detected at all time (see section 4).
- **Reduced Cover Selective Predictive Strategy (RCSPS):** This strategy is an enhancement to RCSTS where only a subset of those activated using RCSTS are activated depending on the targets next predicted location (see section 4).

We simulated a virtual large network of sensor nodes deployed on a 10x10 region of interest. A total of 1000 nodes were deployed. Different trajectories of the target were considered.

First, the optimum coverage algorithm described in section 3 is used to find the reduced cover of region 10x10x10 units when sensor nodes are randomly deployed. The nodes have a sensing radius of 1 unit and different numbers of nodes are randomly deployed in this region using a random distribution. It can be seen that the size of the reduced cover is almost the same as the number of deployed sensor nodes is increased, which indicates that our algorithms are scalable. Starting with 1000 deployed sensor nodes, the necessary number of nodes to be active in order to fully cover the region of interest is about 82 sensor nodes, resulting in a great deal of energy savings.

The performance of the network over time was also studied to determine the benefits of using a reduced cover tracking algorithm. This is done by assuming that each sensor node has a limited energy supply of 300 Joules and is deactivated when the available energy is used up. The performance is evaluated in terms of *coverage lifetime*. It can be seen in both the cases that the overall coverage drops over time as the available energy is used in processing the queries. Using the reduced network, it is seen that the resultant cover over time is significantly better. This is because each node in the reduced network has fewer neighbors and as a result has more efficient communications and less energy expenditure per query. This improvement in the coverage lifetime comes at a cost. The algorithm for obtaining the reduced network requires the communication between a node and its neighbors and as a result a portion of energy is used up during the initialization stage of the network. This causes early onset of degradation and loss of cover. This, however, can be addressed by incorporating *self healing* in the WSN [13].

Using the basic predictive strategy BSPS, there is significant improvement on the amount of energy consumed since only a subset of the nodes are active depending on the next location of the target. Comparison between the 3 algorithms is presented in Figure 7(a). Using the two

proposed tracking algorithms (RCSTS and RCSPS), we notice that we can further improve on the energy savings and thus increasing the overall cover life time of the system as depicted in Figure 7(b). As we increase the number of deployed sensor nodes, the results are very similar however, we note the using BSPS decreases its performance since all the sensor nodes within a specified radius are activated while using the reduced cover approach, the number of sensor nodes to be activated would almost be the same since we first calculate the reduced cover of the region and then activate the necessary subset in order to track the target. The results are shown in Figure 7(c) and Figure 7(d).

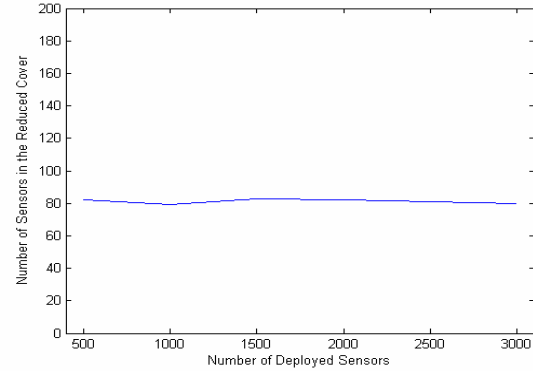
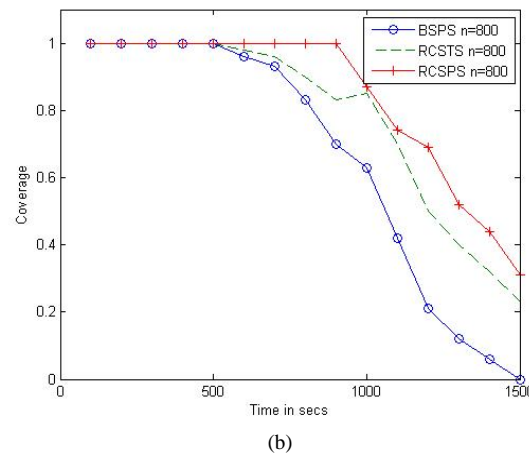
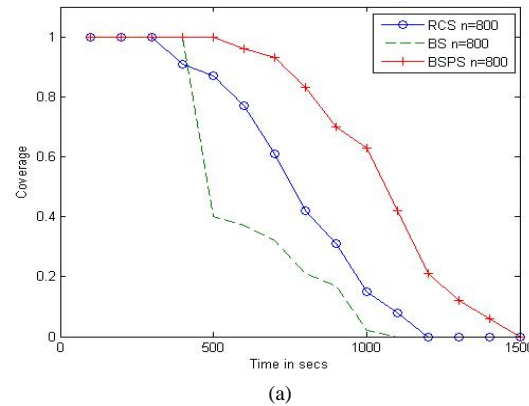


Figure 6: Different number of deployed sensor nodes with sensing radius of 1 unit resulting in different sizes of the reduced cover of a 10x10 units region.



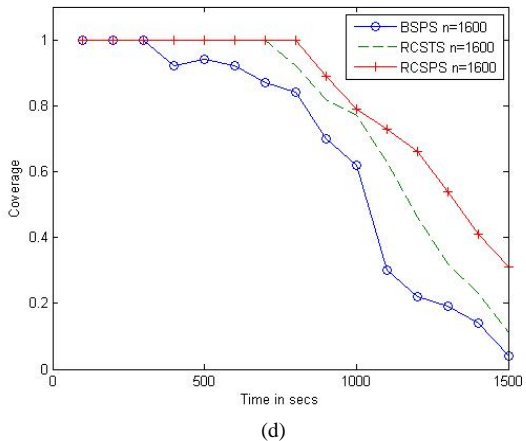
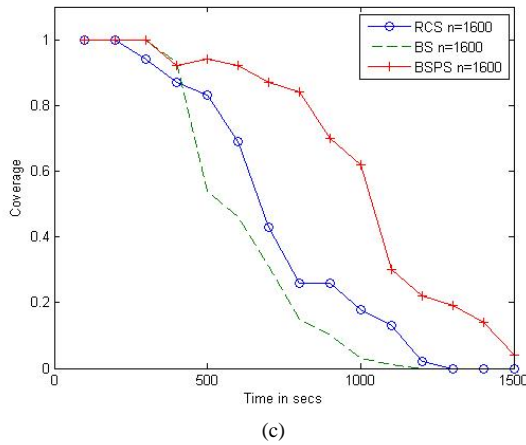


Figure 7: (a),(c) The coverage life time of the network as time passes using 3 different algorithms BS, RCS, BSPS when the number of deployed nodes is 800 and 1600 respectively. (b), (d) The coverage life time of the network as time passes using 3 different algorithms BS, RCSTS, RCSPS when the number of deployed nodes is 800 and 1600 respectively.

The next experiment is more concerned with the accuracy of our tracking algorithm and compares the tracking error metric discussed in section x for the different tracking algorithms. We notice that in terms of tracking error, with no surprise, BS outperforms all the others. However, as the sensing radius of each sensor node is increased, all the other algorithms converge to a negligible tracking error. The results are shown in Figure 8.

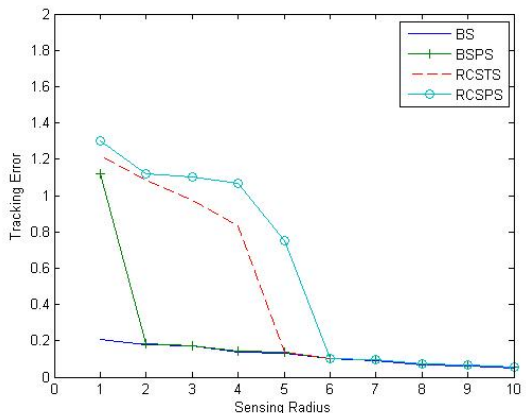


Figure 8: The resulting tracking error as we increase the sensing radius of each sensor node using 4 different algorithms: BS, BS, BS, RCSTS, and RCSPS.

The simulation results show us the energy-quality trade offs between the different tracking algorithms. Our algorithms minimize the number of active nodes while guaranteeing the moving target is detected at all times.

VI. CONCLUSIONS

In this paper, distributed tracking algorithms using wireless sensor networks were proposed. Theoretical as well as experimental results were developed. Unlike previous work in this area, our algorithms make use of a minimal subset of sensor nodes in order to track a target which minimizes the overall energy consumption and therefore the lifetime of the network. Our work in this paper is a continuation of our previous work where 3D full coverage and border coverage problem were analyzed. Our future work includes more sophisticated sensor models.

REFERENCES

- [1] B. Warneke, K.S.J. Pister, "MEMS for Distributed Wireless Sensor Networks," *Int'l Conf on Electronics, Circuits and Systems*, Croatia, September 2002.
- [2] K. Sohrabi, J. Gao, V. Ailawadhi and G. J. Pottie, "Protocols for self-Organization of a Wireless Sensor Network," *IEEE Personal Communication*, October 2000.
- [3] C. Meesookho, S. Narayanan, and C.S. Raghavendra, *Collaborative Classification Applications in Sensor Networks, Sensor Array and Multichannel Signal Processing Workshop*, August 2002
- [4] D. Li, K.D. Wong, Y.H. Hu, and A.M. Sayeed, *Detection, classification, and tracking of targets, IEEE Signal Processing Magazine*, March 2002
- [5] Q. Fang, F. Zhao, and L. Guibas, *Counting Targets: Building and Managing Aggregates in Wireless Sensor Networks*, Xerox Palo Alto Research Center (PARC) Technical Report, June 2002
- [6] M. Chu, H. Haussecker, and F. Zhao, *Scalable Information-Driven Sensor Querying and Routing for ad hoc Heterogeneous Sensor Networks*, Xerox Palo Alto Research Center (PARC), July 2001
- [7] F. Zhao, J. Shin, J. Reich, "Information-Driven Dynamic Sensor Tracking Applications." *IEEE Signal Processing Magazine*, March 2002.
- [8] J. Liu, P. Cheung, L. Guibas, and F. Zhao, *A Dual-Space Approach to Tracking and Sensor Management in Wireless Sensor Networks*, Palo Alto Research Center, Technical Report P2002-10077, March 2002
- [9] P. Ramanathan, *Location-centric Approach for Collaborative Target Detection, Classification, and Tracking, IEEE CAS Workshop on Wireless Communication and Networking*, September 2002
- [10] R. Brooks, and C. Griffin, "Traffic model evaluation of ad hoc target tracking algorithms," *Journal of High Performance Computer Applications*, Accepted, 2002.
- [11] R. Brooks, C. Griffin, and D. S. Friedlander, "Self-Organized distributed sensor network entity tracking," *International Journal of High Performance Computer Applications*, special issue on Sensor Networks, vol. 16, no. 3, Fall 2002
- [12] J. Moore, T. Keiser, R. R. Brooks, S. Phoha, D. Friedlander, J. Koch, A. Reggio, and N. Jacobson, "Tracking Targets with Self-Organizing Distributed Ground Sensors," 2003 IEEE Aerospace Conference, Invited Paper, Accepted for publication, November 2002.
- [13] Watfa M., Commuri S., "An Energy Efficient and Self-Healing 3-Dimensional Sensor Cover", *International Journal of Ad Hoc and Ubiquitous Computing (IJAHUC)*, 2006.
- [14] Watfa M., Commuri S., "Optimal 3-Dimensional Sensor Deployment Strategy", *IEEE Consumer Communications and Networking Conference(CCNC06)*, Las Vegas, NV, USA, January 2006