Co-Grid: an Efficient Coverage Maintenance Protocol for Distributed Sensor Networks

Guoliang Xing; Chenyang Lu; Robert Pless
Department of Computer Science and Engineering
Washington University in St. Louis
St. Louis, MO, USA
{xing,lu,pless}@cse.wustl.edu

Joseph A. O'Sullivan
Department of Electrical Engineering
Washington University in St. Louis
St. Louis, MO, USA
jao@ee.wustl.edu

ABSTRACT

Wireless sensor networks often face the critical challenge of sustaining long-term operation on limited battery energy. Coverage maintenance protocols can effectively prolong network lifetime by maintaining sufficient sensing coverage over a region using a small number of active nodes while scheduling the others to sleep. We present a novel distributed coverage maintenance protocol called the *Coordinating Grid (Co-Grid)*. In contrast to existing coverage maintenance protocols which are based on simpler detection models, Co-Grid adopts a distributed detection model based on data fusion that is more consistent with many distributed sensing applications. Co-Grid organizes the network into coordinating fusion groups located on overlapping virtual grids. Through coordination among neighboring fusion groups, Co-Grid can achieve comparable number of active nodes as a centralized algorithm, while reducing the network (re-)configuration time by orders of magnitude. Co-Grid is especially suitable for large and energy-constrained sensor networks that require quick (re-)configuration in response to node failures and environmental changes. We validate our claims by both theoretical analysis and simulations.

Categories and Subject Descriptors

C.3 [**Special-purpose and Application-based Systems**]: Real-time and embedded systems; C.2.2 [**Computer-communication Networks**]: Network Protocols—*Applications*

General Terms

Algorithms, Design, Experimentation

Keywords

Sensor Networks, Energy Conservation, Coverage, Data Fusion, Distributed Detection

IPSN'04, April 26–27, 2004, Berkeley, California, USA.

1. INTRODUCTION

Many wireless sensor networks face the critical challenge of sustaining long-term operation on limited battery energy [11]. Coverage maintenance has been proposed as a promising approach to prolong network lifetime. A coverage maintenance protocol provides required sensing coverage over a geographic region by activating a subset of nodes, while scheduling the others to sleep, to conserve energy. Unfortunately, although a number of coverage maintenance protocols have been developed [2, 14, 18, 19, 20], they are often designed based on deterministic detection models (such as the disc model) that are not applicable to real-world distributed sensing applications. For example, existing protocols assume that each sensor performs sensing independently. This assumption is invalidated by many sensor network applications which rely on data fusion to improve sensing performance.

In this paper, we investigate coverage maintenance based on a probabilistic distributed detection model that allows efficient data fusion from multiple nodes. We characterize coverage by the minimum event detection probability in a region and the system false alarm rate from the active nodes in a network. In our model, the event detection probability and false alarm rate are computed based on an existing data fusion algorithm that correlates detection decisions from multiple nodes. While the adopted fusion algorithm is not new, this coverage formulation provides a basis for bridging the gap between coverage maintenance protocols and distributed detection algorithms.

Distributed detection problem has been shown to be computationally difficult and even NP-complete in some cases [15]. In general, the computational cost for determining the detection probability from a large number of nodes is high due to the need to consider the combination of detection decisions from multiple nodes. Consequently, the key challenge to developing a coverage maintenance protocol that is based on realistic distributed detection model is to (re-)configure the network coverage within a short time while reducing the number of active nodes in order to prolong the network lifetime. A coverage configuration (or *configuration* for abbreviation) of a sensor network is characterized by a set of active nodes that can maintain the coverage of the network. Reconfiguration is needed at runtime when the current active nodes fail to cover the network. Quick reconfiguration is particularly important in large and dynamic sensor networks subject to node failure and changing application requirements. Unfortunately, the need to reduce the

number of active nodes and the network coverage configuration time can conflict with each other. For example, while a centralized algorithm that treats the whole network as a single fusion group can result in a small number of active nodes, it often requires an extremely long time to configure a large network. On the other hand, although a protocol can significantly reduce the configuration time by dividing the network into separate fusion groups that can configure themselves in parallel, it may result in excessive number of active nodes due to the lack of coordination among neighboring fusion groups.

The main contribution of this paper is the design of a novel distributed coverage maintenance protocol called the *Coordinating Grid (Co-Grid)* protocol that meets both key requirements. Co-Grid organizes the network into fusion groups located on overlapping virtual grids. Through effective coordination among neighboring fusion groups, Co-Grid can achieve comparable number of active nodes as the centralized algorithm. Furthermore, the coverage configuration time of Co-Grid is competitive with the protocol based on separate fusion groups. We also present a theoretical analysis of the efficiency of the Co-Grid protocol. Despite the inter-dependency among fusion groups, our analysis shows that Co-Grid can achieve a high degree of parallelism in the coverage configuration process. This property enables Co-Grid to accomplish configuration quickly and scale to large networks. The advantages of Co-Grid are validated in simulation experiments.

In the rest of this paper, we first discuss related work in Section 2. We then present our coverage formulation in Section 3, followed by a description of a specific distributed detection model used in this paper. The design and analysis of our coverage maintenance protocols are presented in Sections 5 and 6, respectively. Finally, we conclude the paper after presenting the simulation results.

2. RELATED WORK

Recent work on coverage maintenance protocols has demonstrated their promise in energy conservation in wireless sensor networks [2, 14, 18, 19, 20]. Several event detection models have been adopted by existing protocols. In the disc model, a sensor is assumed to have a perfectly circular *sensing range* within which any event can be *sensed* or *detected*. Although this disc model allows a geometric treatment of coverage problem, it does not capture the stochastic nature of signals. The exposure formulations introduced in [12] incorporated a signal decay function. However, they do not model multi-sensor data fusion based on established distributed detection algorithms [17].

There is a vast literature on distributed detection using multiple sensors [1, 7, 10, 16, 17]. Distributed detection in capacity-constrained sensor networks is studied in [9]. "Censoring sensors" is proposed in [13] as an energy-efficient scheme for distributed detection in sensor networks by censoring sensor observations based on Local Likelihood Ratio (LLR). The energy consumption is reduced by only transmitting the observations that exceed LLR to the fusion center. The deployment cost needed to achieve the desired detection performance for path traversal is studied in [3]. A centralized algorithm is proposed in [4] to place sensors at grid points of a region to provide desired detection performance. Although computational cost is not a concern in the offline algorithms proposed by these works, it plays an important role in the online coverage configuration algorithms we study in this paper. [6] showed that the organization of clusters has a significant effect on the overall network detection performance. A framework for various collaborative signal processing tasks is discussed in [8], in which the local processing is enabled by dividing the sensor field into "space-time" cells. These projects are not concerned with coverage maintenance protocols that control sensors' duty cycles for energy conservation.

3. PROBLEM FORMULATION

In this paper, we define *coverage* of a sensor network based on a probabilistic detection model. A point p is *covered* by a sensor network if the probability that a target, located at p, is detected by the active nodes is above threshold β and the system false alarm rate is below threshold α. A geographic region is covered by a sensor network if *all* the points in this region are covered. Formally, the coverage requirement of a region A is defined as:

$$(\forall(x,y) \in A, P_D(x,y) \geq \beta) \wedge (P_F \leq \alpha) \quad (1)$$

where $P_D(x,y)$ and P_F represent the detection probability of a target located at (x,y) [1] and the system false alarm rate, respectively.

This probabilistic coverage formulation captures the requirements of many detection-based applications in sensor networks. In addition, it is also useful for other types of sensing applications. For example, a coverage maintenance protocol based on this detection model can be used in a surveillance application. The network can execute the protocol to maintain sufficient detection probability. Once a target is detected, the sleeping nodes are woken up to execute more sophisticated sensing tasks such as intruder tracking.

In this paper, we focus on the design of distributed network protocols that can provide the required coverage over a region by activating a small number of nodes within a short time. Note that while this paper does not focus on the design of data fusion algorithms, our coverage maintenance protocols can be extended to incorporate different data fusion algorithms.

4. DETECTION MODEL

In this section, we present a distributed detection model that can be combined with our coverage maintenance protocol. We assume that each node in a sensor network belongs to one or more *fusion groups*. Each fusion group has a node serving as its *fusion center*. Each node in the fusion group measures the signal and makes its own *local* decision on whether a target is present or not. Then the local decisions of individual nodes are transmitted to the fusion center. The fusion center uses a fusion rule to reach a global decision based on the local decisions.

4.1 Signal Model

In a large-scale sensor network, the system detection performance depends on the spatial distribution of nodes. To capture the correlation between the spatial property of a sensor network and the system detection performance, we

[1]For convenience, $P_D(x,y)$ is referred to as the *detection probability at* (x,y) hereafter.

introduce spatial signal decay parameters into the Neyman-Pearson detector model [17]. Each node detects the targets of interest by measuring the sound power with its acoustic sensor. The power of the acoustic signal emitted by a target decays over the distance of propagation. Specifically, for a target located at point (x, y), we assume that the signal power measured by node i located at position (x_i, y_i) can be described by the following equation:

$$e(x_i, y_i) = \frac{be_0}{d((x,y),(x_i,y_i))^a} \quad (2)$$

where $d((x,y),(x_i,y_i))$ is the distance between the target and node i. e_0 represents the initial power of the signal emitted by the target. We assume that the location of a potential target is not known to the nodes in advance. a and b are attenuation factors determined by propagation properties of sound signals.

A node makes a decision on whether a target is present or not based on its measurement. The noise in the measurement of a node is modeled as a Gaussian distribution with zero-mean and the target signal is modeled as a Gaussian distribution with nonzero-mean. We assume that when the target is present, the mean of the target signal observed at node i equals the square root of the signal power. Thus the task of detection at node i is to test the following two hypotheses:

$$H^0 : \ p(z_i|H^0) = \frac{1}{\sqrt{2\pi}\sigma} \exp\left(-\frac{z_i^2}{2\sigma^2}\right) \quad (3)$$

$$H^1 : \ p(z_i|H^1) = \frac{1}{\sqrt{2\pi}\sigma} \exp\left(-\frac{(z_i - \sqrt{\frac{be_0}{d((x,y),(x_i,y_i))^a}})^2}{2\sigma^2}\right) \quad (4)$$

where z_i represents the measurement at node i. H^0 and H^1 represent the hypothesis that the target is absent and present, respectively. When the signal power decays with distance, the difference between the means of the two hypothesis decreases accordingly, resulting in worse detection performance.

4.2 Decision Rules

We assume that node 0 serves as the fusion center in a fusion group and all other nodes $(1 \sim n)$ in the fusion group send their decisions to node 0 which makes a global decision. In this section, we investigate the decision rule at node i $(1 \le i \le n)$ and the fusion center, respectively.

At node i $(1 \le i \le n)$, the optimal decision rule is LRT (Likelihood Ratio Test)[17]:

$$\frac{P(z_i|H^1)}{P(z_i|H^0)} \underset{u_i=0}{\overset{u_i=1}{\gtrless}} \lambda_i \quad (5)$$

where u_i represents the decision (0 or 1) of detection at node i. Using (3) and (4), the LRT can be transformed to a test on node measurement z_i and a *decision threshold* λ_i^*, *i.e.*, node i decides on one (a target is detected) if its measurement z_i is greater than λ_i^*, otherwise it decides on zero (no target is detected).

For node i $(1 \le i \le n)$, the *false alarm rate*, denoted by P_{Fi}, represents the probability that node i decides on one while no target is present. The *detection probability*, denoted by P_{Di}, represents the probability that a target located at (x, y) is detected by node i. The false alarm rate (detection probability) of node i $(1 \le i \le n)$ is referred to as the *local* false alarm rate (detection probability) while that of the fusion center is referred to as the *system* false alarm rate (detection probability). The local false alarm rate and detection probability are given by:

$$P_{Fi} = \int_{\lambda_i^*}^{\infty} p(z_i|H^0)dz_i \quad (6)$$

$$= Q\left(\frac{\lambda_i^*}{\sigma}\right) \quad (7)$$

$$P_{Di}(x,y) = \int_{\lambda_i^*}^{\infty} p(z_i|H^1)dz_i \quad (8)$$

$$= Q\left(Q^{-1}(P_{Fi}) - \frac{\sqrt{\frac{be_0}{d((x,y),(x_i,y_i))^a}}}{\sigma}\right) \quad (9)$$

$Q(x)$ is given by $\frac{1}{\sqrt{\pi}}\int_x^{\infty} e^{-t^2/2}dt$. λ_i^* can be solved from (7) when P_{Fi} is known. From (9), we can see that the detection probability of node i depends on the local false alarm rate and the distance to the target.

In this paper we assume that the fusion center uses the majority rule[2]. That is, when the number of ones is larger than that of zeros in the local decisions, fusion center decides that a target is present. The system detection probability at location (x, y) (denoted by $P_D(x, y)$) and the false alarm rate (denoted by P_F) can be expressed as follows:

$$P_D(x,y) = \sum_{|S_1|>|S_0|} \prod_{i\in S_0} (1 - P_{Di}(x,y)) \prod_{j\in S_1} P_{Dj}(x,y) \quad (10)$$

$$P_F = \sum_{|S_1|>|S_0|} \prod_{i\in S_0} (1 - P_{Fi}) \prod_{j\in S_1} P_{Fj} \quad (11)$$

where S_0 and S_1 represent the set of nodes whose decisions are zeros and ones, respectively. We can see that the number of addends in (10) equals the number of node combinations in which more than half of nodes decide one. Thus the complexity of computing $P_D(x, y)$ is $O(2^n)$ (where n is the number of nodes). According to the definition of detection probability in (10), the minimal detection probability in region A is:

$$P_{Dmin} = \min_{(x,y)\in A} \sum_{|S_1|>|S_0|} \prod_{i\in S_0} (1 - P_{Di}(x,y)) \prod_{j\in S_1} P_{Dj}(x,y) \quad (12)$$

We assume that all nodes have the same local false alarm rate α_0. From (1) and (11), we have:

$$\sum_{|S_1|>|S_0|} \prod_{i\in S_0} (1 - \alpha_0) \prod_{j\in S_1} \alpha_0 \le \alpha \quad (13)$$

Since α and the number of nodes are known, the maximal value of local false alarm rate at each node can be solved from (13) and will be used by each node in order to achieve the highest system detection probability $P_D(x, y)$. Then the *decision threshold* λ_i^* on the measurement of node i $(1 \le i \le n)$ can be solved from (7).

5. DESIGN OF COVERAGE MAINTENANCE PROTOCOLS

[2]The LRT (Likelihood Ratio Test) at fusion centers is dependent on the target location, and hence is not applicable here.

In this section we present the design of three coverage maintenance protocols. The first protocol, *Central*, employs a centralized algorithm that treats the whole region as a single fusion group. To reduce the coverage configuration time, we further design two distributed algorithms based on "virtual grid"s. The *Separate Grid (Se-Grid)* protocol divides the region into separate grids and all nodes in each grid form a fusion group. Fusion centers perform coverage configuration for their respective grids independently of each other. The *Coordinating Grid (Co-Grid)* protocol organizes the region into overlapping grids that coordinate with each other to achieve coverage.

5.1 Centralized Coverage Maintenance Protocol

In the Central protocol one node is elected among all nodes in the region A to serve as the fusion center. In the coverage configuration phase, the fusion center decides which nodes should remain active and compute their local false alarm rate such that the coverage requirement (1) is met. Initially all nodes are marked as *sleep* by the fusion center. In each iteration of the algorithm, a node is marked as *active*. Given the system false alarm rate threshold α and the number of current active nodes, the fusion center computes a local false alarm rate for active nodes by (13). Using the active nodes' locations and the local false alarm rate, the fusion center finds the location (x_{min}, y_{min}) in region A that has the minimal detection probability P_{Dmin}. If P_{Dmin} is less than β, the fusion center finds the node closest to point (x_{min}, y_{min}) among all sleeping nodes and marks it as *active*. This process repeats until the minimal detection probability P_{Dmin} in region A is greater than β. Then, the fusion center sends a list of active node IDs and the local false rate it computed to all nodes in region A. If a node finds its ID in the list, it remains active and sets its *decision threshold* according to the local false alarm rate using (7). Otherwise it goes to sleep and wakes up periodically to check whether it should activate itself (by listening to messages from the fusion center). The pseudo code for the coverage configuration algorithm of Central is shown as follows:

/*α and β are the thresholds on the system false alarm rate and detection probability, respectively*/
Central(α,β)
begin
  Initialize table *actNodes* that stores the IDs and locations of active nodes to $\emptyset$;

  $P_{Dmin} = 0$;

  Randomly pick a sleeping node and put in *actNodes*;

  /*Initialize the local false alarm rate to the system false alarm rate*/
  $\alpha_0 = \alpha$;

  while ($P_{Dmin} < \beta$)

    if all nodes in $G(i,j)$ are *active*
      return FAILURE;
    fi

    /*If more than one point has detection probability P_{Dmin}, randomly pick one as (x_{min}, y_{min})*/
    Find the point $(x_{min}, y_{min}) \in A$ that has minimal detection probability P_{Dmin} based on α_0 and the active node locations in *actNodes* according to (9) and (12);

    Put the sleeping node closest to point (x_{min}, y_{min}) in *actNodes*;

    /*Adjust the local false alarm rate to satisfy the constraint on the system false alarm rate*/
    Re-compute α_0 based on α using (13);
  end

  broadcast the active node table *actNodes* and α_0 to all nodes;
end

In each iteration of Central, the fusion center computes the minimal detection probability P_{Dmin} in the deployment region. From (12), we can see that the optimal solution of P_{Dmin} is computationally difficult to obtain and only the numerical solution exists. We compute the approximate solution of P_{Dmin} as follows. The region is divided into a matrix of small square patches and the target is assumed to only appear at the corners of the patches (referred to as *sample points*). The detection probability associated with each sample point is then computed using (10) and the minimum detection probability is obtained. As discussed in Section 4.2, the complexity of computing the detection probability of a point is $O(2^n)$. Central may incur high computational cost and unacceptable coverage configuration time when the number of active nodes is large. Furthermore, distant nodes make irrelevant detection decisions due to signal decay. Consequently, fusing the decisions from all the nodes distributed in the region may not improve the overall detection performance. We evaluate this effect experimentally in Section 7.

5.2 Coverage Maintenance Protocol based on Separate Grids

To facilitate parallel processing, we propose the Se-Grid protocol. In Se-Grid, the deployment region of a sensor network is divided into a matrix of identical grids. Each grid is labeled as $G(i,j)$ where (i,j) is the grid index. No grids overlap with each other. The nodes in each grid form a fusion group that executes the Central protocol within its own grid. That is, each fusion group is responsible for covering its own grid by activating nodes within the grid.

Se-Grid can effectively reduce the configuration time because the grids in the region are configured in parallel. However, since Se-Grid restricts decision fusion within each grid, a node cannot contribute to the decision fusion of a neighboring fusion group even if it is close to the grid's boundary. Furthermore, the nodes close to grid boundary are more likely to be activated. This is because the detection probability of nodes decreases quickly with distance, and hence the vicinity of grid boundary usually has lower probability than other locations. Therefore, Se-Grid may activate redundant nodes on both sides of a grid boundary.

5.3 Coverage Maintenance Protocol with Inter-grid Coordination

Since the problem of Se-Grid is mainly caused by the lack of collaboration among adjacent fusion centers, we design the Co-Grid protocol that provides efficient inter-grid coordination. In Co-Grid, the network deployment region con-

sists of overlapping grids. Each grid is composed of four identical sub-grids and each sub-grid belongs to up to four grids. Figure 1 illustrates nine overlapping grids composed of 16 sub-grids. The fusion center of each grid is located at the center of the grid and denoted by a small black circle. Each sub-grid is labeled as $s(i)$ and each grid is labeled as $G(i,j)$. For example, grid $G(1,2)$ consists of four sub-grids $s(5)$, $s(6)$, $s(9)$ and $s(10)$. $s(10)$ (shaded in Figure 1) belongs to four overlapping grids $G(1,2)$, $G(1,3)$, $G(2,2)$, $G(2,3)$. We say two fusion centers are *adjacent* if their grids overlap, *i.e.*, share one or more sub-grids. A fusion center can have up to eight adjacent fusion centers. Since each node belongs to multiple grids, it can contribute to the detection at multiple fusion centers around it. Therefore this algorithm can potentially result in less active nodes by reducing the redundancy in neighboring grids.

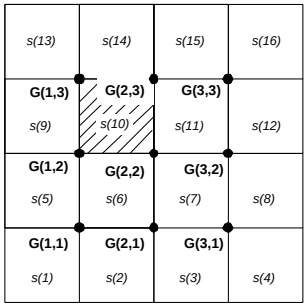

Figure 1: Overlapping Grid Layout

In each iteration of Co-Grid, similar to Se-Grid, a fusion center computes the detection probability of each sample point in a grid and activates the node closest to the sample point with the minimum detection probability until the minimum detection probability in the grid is above threshold β. However, the overlapping grid layout introduces additional complexity in the design due to the dependency among overlapping grids. Since each sample point p in the region belongs to up to four grids (referred to as the *master grids* of point p), we only need to guarantee that the *joint* detection probability from p's master grids is no lower than β. However, the joint detection probability is difficult to compute because the master grids of p share nodes with each other and hence their detection probabilities are not independent. Instead of enforcing the joint probability, Co-Grid guarantees the coverage of point p by ensuring that at least one of its master grids has a detection probability at p that is no lower than β. That is, Co-Grid uses the maximum of the detection probabilities computed by all the master grids of p as an approximation to the joint detection probability at p. For example, to compute the detection probability of a sample point in sub-grid $s(10)$ in Figure 1, the fusion center of $G(2,2)$ computes the maximum of the point's detection probabilities computed by the fusion centers in $G(1,2)$, $G(1,3)$ and $G(2,3)$.

To find the point with the minimum detection probability in a grid, a fusion center running Co-Grid needs to compute every sampling point's detection probability, which is the maximum of all detection probabilities computed by the point's mater grids. To reduce the inter-grid communication, instead of communicating detection probabilities of sample points among adjacent fusion centers, we let each fusion center compute the detection probabilities on behalf of its adjacent fusion centers. A fusion center keeps a local false alarm rate and a list of locations of active nodes for each adjacent fusion center. The procedure performed by the fusion center in grid $G(i,j)$ to compute the minimal detection probability $P_{Dmin}(i,j)$ of grid $G(i,j)$ can be formulated as follows:

$$P_{Dmin}(i,j) = \min_{(x,y)\in G(i,j)} \max_{(x,y)\in G(m,n)} P_D(x,y,m,n) \quad (14)$$

where $P_D(x,y,m,n)$ represents the sample point (x,y)'s detection probability in its master grid $G(m,n)$ ($G(m,n)$ is an adjacent grid of $G(i,j)$). Note each sample point has up to four master grids and all sample points in a sub-grid share the same master grids. $P_D(x,y,m,n)$ is computed according to (10) and (13) using the locations of active nodes and the local false alarm rate of grid $G(m,n)$.

At runtime, when the fusion center in $G(i,j)$ activates a node in its grid, the local false alarm rate of active nodes in $G(i,j)$ needs to be re-computed using (13) to satisfy the constraint on the system false alarm rate. This may potentially result in changes in the detection probabilities of all sample points in the grid. Due to sharing of sub-grids, up to eight adjacent fusion centers need to know the updated local alarm rate and locations of active nodes in $G(i,j)$ before they can activate any new node. Therefore, activating a new node in a grid may invalidate the ongoing processes of activating any other new node in its adjacent grids. To resolve the contention among adjacent grids, we consider the following two approaches. 1) The fusion center in $G(i,j)$ notifies its adjacent fusion centers both before it starts and after it completes the process of finding a new active node. Because the location of the new active node is unknown before the process completes, all the adjacent fusion centers have to wait until they receive the result from the fusion center in $G(i,j)$. While this "locking" strategy sequentializes all the computations of adjacent fusion centers, it is pessimistic and may unnecessarily reduce the efficiency of Co-Grid (detailed analysis is presented in Section 6). 2) Each fusion center performs the process of activating new nodes independently. Whenever the fusion center in $G(i,j)$ activates a new node, it advertises the local false alarm rate and the locations of current active nodes to its adjacent fusion centers, which cancel their current computations and restart with the updated parameters received from the fusion center of $G(i,j)$. This approach maximizes the parallelism of adjacent fusion centers and is adopted by Co-Grid.

The pseudo-code of the Co-Grid protocol at fusion center $G(i,j)$ is shown as follows:

```
/*Global definitions*/
struct Grid {
    /*IDs and locations of the active nodes in the grid*/
    list actNodes;
    /*Local false alarm rate of the active nodes*/
    double α0;}

/*Information of grid G(i,j)*/
struct Grid self;

/*Information of adjacent grids*/
struct Grid adjGrids[8];

/*α and β are the thresholds on the system false
alarm rate and detection probability*/
ActivateNode(α,β)
begin
    if all nodes in G(i,j) are active
```

```
        return FAILURE;
    fi

    if  self.actNodes is empty
        Randomly pick a sleeping node and put
        in self.actNodes;
    fi

    do
        /*Adjust the local false alarm rate α0 to satisfy
        the constraint on the system false alarm rate*/
        Compute self.α0 using self.actNodes
        according to (13);

        /*If more than one point has detection probability
        PDmin, randomly pick one as (xmin, ymin)*/
        Find the point (xmin, ymin) ∈ G(i, j) that has the
        detection probability PDmin according to (14)
        using self and adjGrids;

        if PDmin ≥ β    return SUCCESS;

        Put the sleeping node closest to point (xmin, ymin)
        in self.actNodes;

        /*Advertise the change of state*/
        send an Update message including self.α0 and
        self.actNodes to all adjacent fusion centers;
    while(PDmin < β)
return SUCCESS;
end

/* The following function is called whenever the
fusion center receives an Update message from
an adjacent fusion center. */
UpdateParameters
begin
    Stop the execution of ActivateNode;
    Update adjGrids array with received information;
    Call ActivateNode(α,β);
end
```

After the process of coverage maintenance completes, each fusion center sends the list of active nodes and the local false alarm rate it computed to all nodes in its grids. If a node finds itself in the list of active nodes, it remains active and sets a decision threshold according to the local false alarm rate it received (see (7)). Since each node belongs to up to four master grids, an active node may have up to four decision thresholds. During detection phase, an active node periodically compares its measurement with each decision threshold and sends a decision (0 or 1) to the corresponding fusion center. Hence an active node needs to send up to four decision messages in each detection period.

6. ANALYSIS OF THE DEGREE OF PARALLEL CONFIGURATION

In Se-Grid, each fusion center can turn on new nodes independently from each other. However, this is not the case for Co-Grid due to the inter-dependencies among overlapping grids. As described in Section 5.3, when a fusion center adds a new active node, all the adjacent fusion centers have to re-start their computation from scratch. We define *effective computation* as the computation in a fusion center that will lead to the addition of a new active node. Any two adjacent fusion centers cannot perform effective computation at the same time. We define the *degree of parallel configuration (DPC)* as the total number of fusion centers that can perform effective computations simultaneously in the whole network. Clearly DPC has a significant impact on the coverage configuration time of the whole network. The DPC of Se-Grid is the total number of grids because all the grids can configure themselves in parallel. In contrast, the DPC of Central is only 1. However, it is less straightforward to quantify the parallelism of Co-Grid due to the inter-grid dependencies.

In order to understand the cost of inter-grid coordination on configuration time, we now analyze the degree of parallel configuration under Co-Grid. We model the network as a graph (referred to as *parallelism graph*) where each fusion center is a vertex and an edge exists between two adjacent fusion centers to represent the fact that any two adjacent fusion centers cannot perform effective computation simultaneously. The fusion centers that can proceed simultaneously form an *independent set*[3] of the parallelism graph. *Maximal independent set* of a graph is a subset of vertices such that there is no edge between any pair of vertices in the set and no more vertices can be added without making it a non-independent set. It is clear that the best-case and worst-case DPCs are equal to the maximal and minimal cardinality of the maximal independent sets of the parallelism graph, respectively.

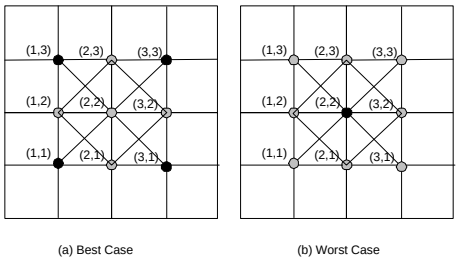

Figure 2: Degree of Parallel Configuration

Figure 2 shows two possible states of the network at configuration time. The network is composed of 9 overlapping grids. One fusion center is located at each grid center. Figure 2(a) shows the maximal possible degree of parallel configuration where each of the four fusion centers denoted by black circles can activate a node simultaneously. That is, in the best case the DPC of the network is 4 under Co-Grid. On the other hand, Figure 2(b) shows the worst-case DPC is 1 where only fusion center $(2, 2)$ activates a node while all other fusion centers cannot proceed.

The best-case DPC of Co-Grid equals the maximal number of grids that do not overlap in the region and can be as good as Se-Grid. Figure 3 shows the worst-case DPC of Co-Grid and the DPC of Se-Grid under different grid width in a $120 \times 120\ m^2$ region. The worst-case DPC of Co-Grid is obtained by computing the minimal cardinality of the maximal independent sets in the corresponding parallelism graph.

When the network size approaches infinite, it can be shown that the worst-case and best-case DPCs of Co-Grid are $n/9$ and $n/4$, respectively, where n is the total number of grids. Since Co-Grid has four times as many grids as Se-Grid for the same grid size, the lower-bound on the ratio between the DPCs of Co-Grid and Se-Grid approaches 4/9 for large regions. This result indicates that the DPC of Co-Grid in-

[3]An independent set of a graph is a subset of the vertices such that no any two vertices in the subset is connected by an edge.

creases about proportionally with the number of grids and hence can scale well in large networks in term of configuration time.

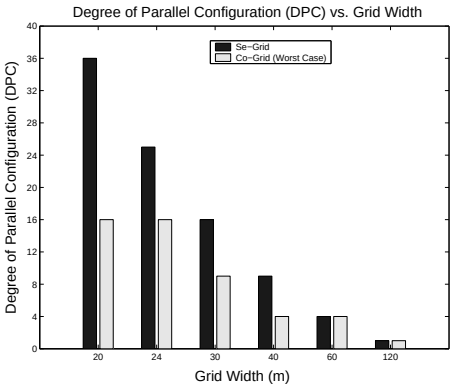

Figure 3: Degree of Parallel Configuration

7. PERFORMANCE EVALUATION

In this section we present the simulation results using Matlab. In addition to our protocols presented in Section 5, we implemented a protocol called Random as the baseline for performance comparison. Random works similarly to Se-Grid except that a fusion center always *randomly* activates a new node in each iteration until the desired detection probability is achieved. Comparing our protocols against Random allows us to study the effectiveness of the greedy strategy that activates the node closest to the point with minimum detection probability.

In each experiment, 2000 nodes are randomly deployed in a 120×120 m^2 region. The thresholds on the system detection probability and false alarm rate are 90% and 5% respectively. All results in this section are averages of five runs. The attenuation factors in the signal decay model (see (2)) are set to $b = 1$, $a = 2$ and the initial signal power of the target (e_0) is set to 200. The variance σ of the noise and node measurement distribution is set to 1.

Since the complexity of computing the detection probability of a point using (10) is $O(2^n)$ (n is the number of active nodes), the simulations are extremely time consuming when the number of active nodes is large. In order to finish such experiments, we employed the Monte Carlo method to approximate the detection probability when the number of active nodes is larger than 17. We note that the long time needed for computing the detection probability on a PC platform again demonstrates the infeasibility of the centralized protocol even on a medium-sized sensor network in which nodes are typically equipped with much slower processors.

In the rest of this section, we first compare the number of nodes activated by each protocol, and then study the coverage configuration time required by each protocol.

7.1 Number of Active Sensors

Figures 4(a) and (b) show the distribution of the nodes activated by Se-Grid and Co-Grid, respectively. The region is divided into 30×30 m^2 grids and the dotted lines donate the boundaries of grids. We can see that many of the nodes activated by Se-Grid are concentrated in the vicinity of the grid corners. In contrast, Co-Grid activates less nodes through efficient coordination and node-sharing among grids. Furthermore, the nodes activated by Co-Grid are distributed more uniformly in the region.

Figure 5 shows the number of nodes activated by the protocols versus the grid width. Note that when the grid width equals $120m$, Se-Grid becomes Central since there is only one grid in the region. We can see that both Co-Grid and Se-Grid outperform Random, which shows that the greedy activation strategy is very effective. Co-Grid outperforms Random and Se-Grid under all grid widths and performs similarly to Central when grid width is larger than $30m$. When the grid width increases, all protocols perform better,

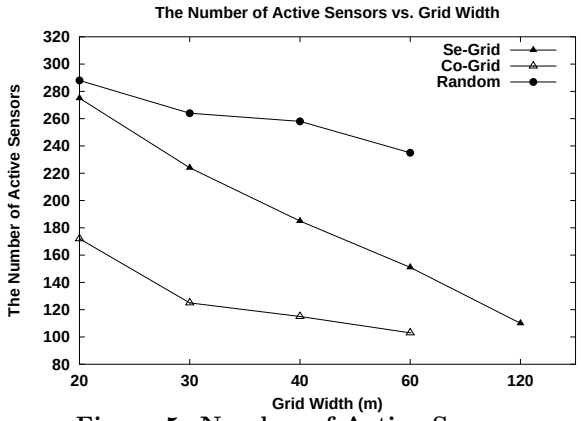

Figure 5: Number of Active Sensors

resulting in less active nodes. This is because, the decrease in the number of grids results in better decision fusion in each grid. When the grid width increases, the number of grids decreases accordingly. This leads to more active nodes in each grid, which potentially results in higher local false alarm rate at each node due to the constraint on the system false alarm rate (see (13)). From (7), higher local false alarm rate always results in higher *decision threshold* and local detection probability at each node. That is, the nodes are more "sensitive" to a target. Consequently, higher overall system detection probability is achieved through fusing the "better" decisions from all nodes. However, from the curve of Co-Grid, we can see the benefit of using larger grids diminishes as the grid width increases. This is because the decisions from the nodes far away from each other have little correlation due to signal decay and cannot contribute to the overall system detection probability.

7.2 Coverage Configuration Time

The coverage configuration time is an important performance metric for coverage maintenance protocols. From (10), the complexity of computing the detection probability of a point is $O(2^n)$ (n is the number of active nodes). Due to the limited processing capacity of nodes, the computational cost will be prohibitively high when the number of active nodes in the network is large. As discussed in Section 6, the overall system performance can be improved by the parallel processing performed in different grids. In this section, we measure the configuration time of different coverage configuration protocols.

The configuration time is decided by the total amount of computation the network must perform and the degree of parallel configuration. The former is affected by the number of activated nodes. We have shown in Section 6 that the degree of parallel configuration of Co-Grid is lower than (but

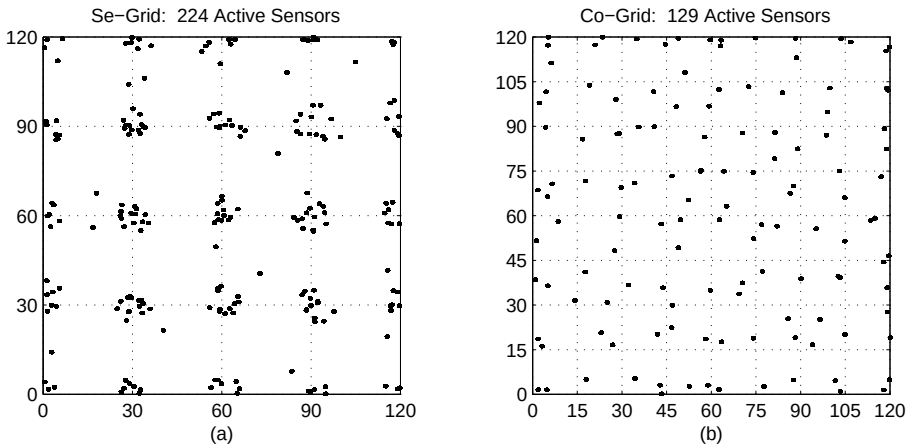

Figure 4: Active Sensors

competitive with) Se-Grid. In this section we analyze the configuration time based on our simulation results.

Since the distributed coverage configuration protocols are implemented in Matlab as sequential programs, the wall-clock time taken by each simulation does not reflect the actual configuration time. In addition, the simulation time is not representative of real systems because it is run in the Matlab/PC environment instead of node devices and also because the Monte Carlo method was used in some simulations. Therefore, instead of depending on the measured simulation time, we estimate the configuration time as follows.

From Section 5, we see that the computational complexity of adding a new active node is $O(2^n)$ in both Se-Grid and Co-Grid, where n is the number of the nodes already activated in the same grid. We assume that the computation time for adding a new active node equals 2^n time unit.

For Se-Grid, the configuration time of each grid is simply the sum of the computation time for each activated node, and the configuration time of the whole network is the maximum configuration time among all fusion groups. However, as discussed in Section 6, estimating the configuration time for Co-Grid is more complicated due to the coordination among adjacent fusion centers. To simulate the distributed execution of fusion centers, at each step, the main simulation program randomly chooses a fusion center to add a new active node to its grid. The chosen fusion center is assumed to have finished its current iteration earlier than all its adjacent fusion centers. In a real system, the addition of a new node by the fusion center will force all of its adjacent fusion centers to restart their computations. To speed up the simulations, our simulator only executes the completed computations that cause the addition of new nodes. The canceled computations are not executed. Note that they do not affect the results on active nodes or the estimation of the total configuration time, because they occur in parallel with the completed computations. To help compute the configuration time, each grid $G(i,j)$ keeps a counter $t(i,j)$ called the *earliest release time*. This counter represents the earliest time when a grid can start effective computation that results in the addition of a new active node. The earliest release times of all grids are initialized to 0 at the beginning of the simulation. After grid $G(i,j)$ adds a new active node, it increases its earliest release time $t(i,j)$ by 2^n, where n is the number of nodes in $G(i,j)$ that are activated before the new node is added. In addition, the main program also advances the earliest release times of $G(i,j)$'s adjacent grids to $t(i,j)$, if their current earliest release times are less than $t(i,j)$. This is because no adjacent grid of $G(i,j)$ can start any useful computation that would lead to the addition of a new node, until $G(i,j)$ finishes its computation and adds a node. After the coverage configuration process finishes, the configuration time of the whole network is the maximum earliest release time among all fusion centers.

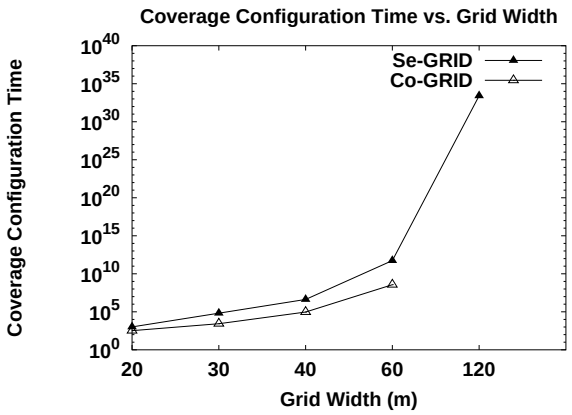

Figure 6: Coverage Configuration Time

Figure 6 shows the configuration time of different coverage maintenance protocols under different grid width in a 120×120 m^2 region. We should note that although the time unit of the configuration time shown in the figure is virtual, it does not affect the relative performance of different protocols. Although Co-Grid is shown to have a lower degree of parallel configuration than Se-Grid in Section 6, we can see that its configuration time is always shorter than Co-Grid. This is because Co-Grid activates fewer nodes than Se-Grid, which results in significantly less amount of com-

putation. When grid width is 120 m, there is only one grid in the region and Se-Grid becomes Central. We can see that Central requires a configuration time that is more than 10^{24} times that of Co-Grid with a grid width of $60m$ that equals half of the region width. Assuming that the time unit is 10 us, it will take Central $\sim 2.6 \times 10^{28}$ sec (or $\sim 3 \times 10^{23}$ days) to configure a network, while it will take Co-Grid only $\sim 3.6 \times 10^{3}$ sec (1 hour). This result clearly demonstrates that Co-Grid is significantly more scalable than Central.

8. IMPORTANT OBSERVATIONS

In this section, we discuss some important observations we made during this work. Addressing these issues in detail is left as future work.

Reducing Communication Cost in Detection: Since the time in which a sensor network operates in detection phase is much longer than in coverage configuration phase, the communication cost of the detection phase plays a more important role in the energy consumption of the network. During the detection phase, each active node needs to send its detection decision periodically to the fusion center(s). To reduce the amount of communication among active nodes and fusion centers, the active nodes and the fusion center in a grid can be organized into a spanning tree rooted at the fusion center. Every active node periodically sends a decision message encoding its own decision and all the decisions received from its children to its parent in the tree. In Co-Grid, each active node needs to communicate with up to four fusion centers and hence the spanning trees in adjacent grids may overlap. In such a case, a more efficient network structure needs to be studied to reduce the amount of communication among active nodes and fusion centers. The payload of the decision messages of the Central protocol is larger than that of other protocols since the decisions of all active nodes need to be transmitted to a single fusion center in the network.

Integrated Connectivity and Coverage Configuration: From Section 7, we see that the number of active nodes decreases as the grid width increases. The active nodes produced by a coverage maintenance protocol may be too sparse to remain connected when the grid size is large. In such a case, more nodes need to be activated to maintain the network connectivity in a grid. The problem of integrated coverage and connectivity configuration has been addressed in [18].

Location-based Fusion Rule: In this paper, the grid size has been shown to play an important role in the detection performance of fusion groups because the detection ability of sensors decays with distance. This result suggests that a decision fusion strategy may provide better performance by taking the locations of the sensors into consideration. For example, fusing the decisions of the sensors in the vicinity of a target may achieve competitive or better detection performance than fusing the decisions of all sensors distributed in a large grid. Although the majority fusion rule is implemented in Co-Grid in this paper, more efficient fusion rules (*e.g.*,[5]) that consider the locations of sensors can be easily incorporated into the Co-Grid framework.

Reducing Computational Cost: As shown in this paper, the coverage configuration based on a data fusion model usually has a high computational cost in a large network. Hence the tradeoff between the quality of the results (*e.g.*, the number of active nodes) and the configuration cost needs to be studied to achieve desirable coverage configuration time without incuring high energy consumption at runtime. For example, when the number of active nodes is large, the detection probability can be estimated by approximation techniques that take polynomial time, such as the Monte Carlo method discussed in Section 7. Another observation is that the procedure for computing the detection probability from a large number of nodes can be easily parallelized (*e.g.*, through customized reconfigurable hardware) because the computations for different combinations of node input are independent of each other.

9. CONCLUSION

In this paper, we introduced the Co-Grid protocol that maintains probabilistic detection guarantees over a region. Co-Grid is designed based on a distributed detection model that considers data fusion among multiple nodes. This distinguishes Co-Grid from existing protocols that are based on simpler detection models. Our theoretical analysis and simulation results demonstrate that Co-Grid not only competes well against the centralized protocol in terms of the number of active nodes, but also consistently outperforms the protocol based on separate grids in terms of the configuration time.

10. ACKNOWLEDGMENTS

This work is funded, in part, by the NSF under an ITR grant CCR-0325529. We thank the anonymous reviewers for their valuable feedback.

11. REFERENCES

[1] Z. Chair and P. Varshney. Optimal data fusion in multiple sensor detection systems. *IEEE Trans. Aerospace Electron. Syst.*, Jan. 1990.

[2] K. Chakrabarty, S. S. Iyengar, H. Qi, and E. Cho. Grid coverage for surveillance and target location in distributed sensor networks. *IEEE Transactions on Computers*, 51(12):1448–1453, December 2002.

[3] T. Clouqueur, V. Phipatanasuphorn, P. Ramanathan, and K. K. Saluja. Sensor deployment strategy fortarget detection. In *The First ACM International Workshop on Wireless Sensor Networks and Applications(WSNA'02)*, Sep 2002.

[4] S. Dhillon, K. Chakrabarty, and S. S. Iyengar. Sensor placement for grid coverage under imprecise. In *Proc. International Conference on Information Fusion (FUSION 2002)*, 2002.

[5] M. Duarte and Y.-H. Hu. Distance based decision fusion in a distributed wireless sensor network. In *The 2nd International Workshop on Information Processing in Sensor Networks (IPSN 2003)*, Palo Alto,CA, April 22-23 2003.

[6] R. V. Dyck. Detection performance in self-organized wireless sensor networks. In *IEEE Int. Symp. on Information Theory*, June-July 2002.

[7] I. Y. Hoballah and P. K. Varshney. Distributed bayesian signal detection. *IEEE Trans. Inform. Theory*, Sept. 1989.

[8] D. Li, K. Wong, Y. H. Hu, and A. Sayeed. Detection, classification and tracking of targets in distributed sensor networks. *IEEE Signal Processing Magazine*, Mar 2002.

[9] M. Longo, T. D. Lookabaugh, and R. M. Gray. Quantization for decentralized hypothesis testing under communication constraints. *IEEE Trans. on Inform. Theory*, 36(2):pp.241–255, Mar 1990.

[10] M. L. T. D. Lookabaugh and R. M. Gray. Quantization for decentralized hypothesis testing under communication constraints. *IEEE Trans. Inform. Theory*, March. 1990.

[11] A. Mainwaring, J. Polastre, R. Szewczyk, and D. Culler. Wireless sensor networks for habitat monitoring. In *ACM International Workshop on Wireless Sensor Networks and Applications*, Atlanta, GA, Sep. 2002.

[12] S. Meguerdichian, F. Koushanfar, M. Potkonjak, and M. B. Srivastava. Coverage problems in wireless ad-hoc sensor networks. In *INFOCOM*, pages 1380–1387, 2001.

[13] C. Rago, P. Willett, and Y. Bar-Shalom. Censoring sensors: A low-communication-rate scheme for distributed detection. *IEEE Transactions on Aerospace and Electronic Systems*, vol.32(2), 1996.

[14] D. Tian and N. Georganas. A coverage-preserved node scheduling scheme for large wireless sensor networks. In *Proceedings of First International Workshop on Wireless Sensor Networks and Applications (WSNA'02)*, pages 169–177, Atlanta, USA, Sep 2002.

[15] J. Tsitsiklis and M. Athans. On the complexity of decentralized decision making and detection problems. *IEEE Transactions on Automatic Control*, May 1985.

[16] J. N. Tsitsiklis. Decentralized detection. *Advances in Statistical Signal Processing*, vol.2, 1993.

[17] P. Varshney. *Distributed Detection and Data Fusion*. Spinger-Verlag, New York, NY, 1996.

[18] X. Wang, G. Xing, Y. Zhang, C. Lu, R. Pless, and C. D. Gill. Integrated coverage and connectivity configuration in wireless sensor networks. In *The First ACM Conference on Embedded Networked Sensor Systems(Sensys 03)*, 2003.

[19] T. Yan, T. He, and J. A. Stankovic. Differentiated surveillance for sensor networks. In *The First ACM Conference on Embedded Networked Sensor Systems(Sensys 03)*, 2003.

[20] F. Ye, G. Zhong, S. Lu, and L. Zhang. Peas: A robust energy conserving protocol for long-lived sensor networks. In *The 23rd International Conference on Distributed Computing Systems (ICDCS'03)*, pages 169–177, May 2003.