Maintaining Sensing Coverage and Connectivity in Large Sensor Networks

Honghai Zhang and Jennifer C. Hou
Department of Computer Science
University of Illinois at Urbana Champaign
Urbana, IL 61801
Email:{hzhang3,jhou}@cs.uiuc.edu

***Abstract*— In this paper, we address the issues of maintaining sensing coverage and connectivity by keeping a minimal number of sensor nodes in the active mode in wireless sensor networks. We investigate the relationship between coverage and connectivity by solving the following two sub-problems. First, we prove that if the radio range is at least twice of the sensing range, a complete coverage of a convex area implies connectivity among the working set of nodes. With such a proof, we can then focus only on the coverage problem. Second, we derive, under the ideal case in which node density is sufficiently high, a set of optimality conditions under which a subset of working sensor nodes can be chosen for full coverage.**

Based on the optimality conditions, we then devise a decentralized and localized density control algorithm, *Optimal Geographical Density Control* (OGDC), for density control in large scale sensor networks. *Ns-2* simulation show that OGDC outperforms the PEAS algorithm [32], the hexagon-based GAF-like algorithm, and the sponsor area algorithm [28] with respect to the number of working nodes needed (sometimes at a 50% improvement), and achieves almost the same coverage as the algorithm with the best result.

I. Introduction

Recent technological advances have led to the emergence of small, low-power devices that integrate sensors and actuators with limited on-board processing and wireless communication capabilities. Pervasive networks of such sensors and actuators open new vistas for many potential applications, such as battlefield surveillance, environment monitoring and biological detection [11], [15], [2], [20].

Since most of these devices have limited battery life and it is infeasible to replenish energy via replacing batteries on up to tens of thousands of sensors in most of the applications, it is well accepted that a sensor network should be deployed with high density (up to 20 nodes/m^3 [26]) in order to prolong the network lifetime. In such a high-density network with energy-constrained sensors, it is neither necessary nor desirable to have all nodes operate in the active mode at the same time. If all the sensor nodes simultaneously operated in the active mode, an excessive amount of energy would be wasted and data thus collected would be highly correlated and redundant. Moreover, excessive packet collision would occur as a result that many sensors intend to send packets especially in the presence of certain triggering events.

One important issue that arises in such high-density sensor networks is *density control* — the function that controls the density of the working sensor set to a certain level [32]. Specifically, density control ensures only a subset of sensor nodes operates in the active mode, while fulfilling the following two requirements: (i) *coverage:* the area that can be monitored is not smaller than that which can be monitored by a full set of sensors; and (ii) *connectivity:* the sensor network remains connected so that information collected by sensor nodes can be relayed back to data sinks or controllers. Under the assumption that an (acoustic or light) signal can be detected with certain minimal signal to noise ratio by a sensor node only if the sensor is within a certain range of the signal source, the first issue essentially broils down to a coverage problem: assuming that each node can monitor a disk (the radius of which is called the *sensing range* of the sensor node) centered at itself on a two dimensional surface, what is the minimum set of nodes that can cover the whole area? On the other hand, the second (connectivity) issue can be studied, in conjunction with the first, if the relationship between coverage and connectivity can be well characterized (e.g., under what condition coverage may imply connectivity and vice versa).

In addition to the above two requirements, it is desirable to choose a minimal set of working sensors in order to reduce power consumption and prolong network lifetime. Finally, due to the distributed nature of sensor networks, a practical density control algorithm should be not only distributed but also completely localized (i.e., relies on and makes use of local information only) [11].

In this paper, we address the above two issues, and based on the findings, propose a fully decentralized and localized algorithm, called *Optimal Geographical Density Control* (OGDC), for density control in large scale sensor networks. Our goal is to maintain coverage as well as connectivity using a minimal number of sensor nodes. We investigate the relationship between coverage and connectivity by solving the following two sub-problems. First, we prove that if the radio range is at least twice of the sensing range, a complete coverage of a convex area implies connectivity among the working set of nodes. With such a proof, we can then focus only on the coverage problem. Second, we explore, under the ideal case in which node density is sufficiently high, a set of optimality conditions under which a subset of working sensor nodes can be chosen for full coverage. Based on the optimality conditions, we then devise a decentralized and localized density control algorithm, OGDC. We also perform *ns-2* simulation to validate OGDC and compare it against a hexagon-based GAF-like algorithm and the PEAS algorithm presented in [31], [32].

Several researchers have addressed the same or similar issues, and we will provide a detailed summary of existing work in Section II. The work reported in [31], [32], [28] comes closest to ours. However, the work reported in [31], [32] does not ensure complete coverage. The work reported in [28], on the other hand, attempts to solve the complete coverage problem, but requires a large number of nodes to operate in the active mode (even more than a simple algorithm based on the idea of GAF does [29]). To the best of our knowledge, we are the first to formally investigate the relationship between coverage and connectivity, and devise a fully decentralized and localized density control algorithm that gives the minimum set of working sensor nodes for full coverage.

The rest of the paper is organized as follows. In Section II we give a detailed summary of existing work. In Section III we investigate the relationship between coverage and connectivity. In Section IV we derive the optimality conditions for full coverage under the ideal case. Following that, we present in Section V the proposed density control algorithm. Finally, we present our simulation study in Section VI and conclude the paper in Section VII.

II. Related Work

Minimizing energy consumption and prolonging the system lifetime has been a major design objective for wireless ad hoc networks. GAF [29] assumes the availability of GPS and conserves energy by dividing a region into rectangular grids, ensuring that the maximum distance between any pair of nodes in adjacent grids is within the transmission range of each other, and electing a leader in each grid to stay awake and relay packets (while putting all the other nodes into sleep). The leader election scheme in each grid takes into account of battery usage at each node. SPAN [6], on the other hand, decides if a node should be working or sleeping based on connectivity among its neighbors. Both algorithms need to perform local neighborhood discovery.

While traditional wired and wireless networks are expected to cater to a variety of user applications, a sensor network is usually deployed to perform surveillance and monitoring tasks. This leads to two key differences. First, algorithms used for wireless ad hoc networks do not address the issue of sensing coverage. Second, although reducing power consumption is a common design objective, algorithms used for wireless ad hoc networks often aim to maximize the life time of each individual node, while those used for sensor networks aim to maximize the time interval of continuously performing some (monitoring) functions. Note that as long as the coverage and connectivity is maintained, a sensor network still functions well even if some sensors die much earlier than others.

Several centralized and distributed algorithms have been proposed for sensing coverage in sensor networks [27], [5], [31], [32], [28]. Slijepcevic *et al.* [27] address the problem of finding the maximal number of covers in a sensor network, where a cover is defined as a set of nodes that can completely cover the monitored area. They proved the NP completeness of this problem, and provided a centralized heuristic solution. They showed that the proposed algorithm approaches the upper bound of the solution under most cases. It is, however, not clear how to implement the solution algorithm in a distributed manner.

Ye *et al.* [31], [32] present PEAS, a distributed, probing-based density control algorithm for robust sensing coverage. In this work, a subset of nodes operate in the active mode to maintain coverage while others are put into sleep. A sleeping node wakes up occasionally to check if there exist working nodes in its vicinity. If no working nodes are within its probing range, it starts to operate in the active mode; otherwise, it sleeps again. The probing range can be adjusted to achieve different levels of coverage redundancy. The algorithm guarantees that the distance between any pair of working nodes is at least the probing range, but does not ensure that the coverage area of a sleeping node is completely covered by other nodes, i.e., it does not guarantee complete coverage.

Cerpa and Estrin [5] present ASCENT, to automatically configure sensor network topologies. In ASCENT, each node measures the number of active neighbors and the per-link data loss rate through data traffic. Based on these two values, it decides whether to sleep or keep awake. ASCENT does not consider the issue of completely covering the monitored region either.

Tian *et al.* [28] provide an algorithm that provides complete coverage using the concept of "sponsored area." Whenever a sensor node receives a packet from one of its working neighbors, it calculates its sponsored area (defined as the maximal sector covered by the neighbor). If the union of all the sponsored areas of the sensor node covers the whole disk covered by itself, it turns itself off. As will be shown in Section VI, this approach may be less efficient than a hexagon based GAF-like algorithm. Moreover, the authors only addressed the coverage problem without investigating the connectivity problem.

The works reported in [21], [18] defines coverage (totally differently) as finding a path through a sensor network, given the location of all sensors. Two coverage problems are studied: the best coverage problem attempts to find the path that minimizes the maximal distance of all points to their closest sensors, while the worst coverage problem attempts to find the path which maximizes the minimal distance of all points on the path to their closest sensors. In particular, Meguerdichian *et al.* [21] presented centralized algorithms for both the best and worst coverage problems, and Li *et al.* [18] gave localized algorithms for both problems. Another related problem is to deploy a minimal number of base stations in cellular networks so as to cover the maximal area. The work reported in [19], [23] approach this problem via devising centralized numerical methods.

In addition to coverage and connectivity, several other issues have also been addressed in the context of sensor networks, such as information dissemination [17], [14], [30], architectural consideration for networked sensor devices [13], [24], localization without the aid of GPS [3], [25], [22]. Those are orthogonal to the work reported in this paper.

III. Relationship Between Coverage and Connectivity

In this section we investigate the relationship between coverage and connectivity. Specifically, we derive the necessary and sufficient condition under which coverage implies connectivity — the radio range is at least twice of the sensing range. We assume the whole area is a convex set, and denote the sensing range and the radio transmission range as, respectively, r_s and r_t.

Theorem 1 *Assume the number of sensors in any finite area is finite. Then the condition that the radio range is at least twice of the sensing range is both necessary and sufficient to ensure that coverage implies connectivity.*

Proof It is easy to see that the necessary condition to ensure connectivity given the full coverage is $r_t \geq 2r_s$. We prove this by devising a scenario in which coverage does not imply connectivity if $r_t < 2r_s$. In Figure 1, a sensor is located at O and has, respectively, a sensing radius r_s and a radio transmission radius $r_t < 2r_s$. Now we place a sufficient number of sensors on the circle centered at O and with radius $r_t + \epsilon < 2r_s$ (where $\epsilon > 0$) such that they together cover the whole disk centered at O and with radius $r_t + \epsilon$. However, this network is not connected since the distance between node O and any other node is more than r_t.

Next we show that $r_t \geq 2r_s$ is also a sufficient condition to ensure that coverage implies connectivity. We assume that the number of sensors in any finite area is finite, and prove this by contradiction. If the network is disconnected, there exists a pair of nodes between which there exists no path. We find a pair of such nodes, (S, D)(Fig. 2), with the minimal distance among all pairs of disconnected nodes. Consider the circle whose center is on the line from node S to node D and the distance between its center and node S is r_s. We claim that there must exist some other node within or on the circle. Otherwise since the number of nodes is finite in any finite area, we can move the circle along $\overline{SD}$ toward node D by a minimum distance ϵ in order for the circle to include another node. Then if we move the circle along $\overline{SD}$ toward node D by a distance $\epsilon/2$, there will be no node within or on the circle. That means the center of the circle is not covered by any node, which violates the condition of coverage. Let node P be such a node that lies within or on the circle (before it is moved). Nodes S and P are connected since their distance is less than $2r_s \leq r_t$. Hence nodes P and D must be disconnected; otherwise nodes S and D are connected. Since $\angle SPD > \pi/2 > \angle PSD$, we have $|\overline{SD}| > |\overline{PD}|$. This contradicts the assumption that nodes S and D has the minimal distance among all the pairs of disconnected nodes. □

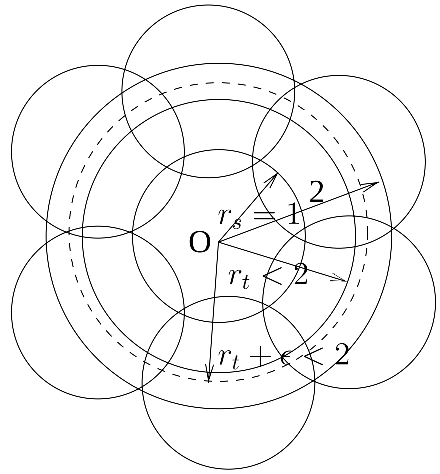


Fig. 1. A scenario that demonstrates $r_t \geq 2r_s$ is a necessary condition that coverage ensures connectivity. All nodes have a sensing range of r_s and a radio range of $r_t < 2r_s$. A sufficient number of nodes are placed on the circle centered at O and with radius $r_t + \epsilon < 2r_s$ (where $\epsilon > 0$), and cover the entire disk centered at O and with radius $r_t + \epsilon$. However, the network is not connected as the distance between node O and any other nodes is more than r_t.

By Theorem 1, if the radio range is at least twice of the sensing range, then if the coverage is satisfied, the connectivity is also satisfied. Although the above derivation is made on a

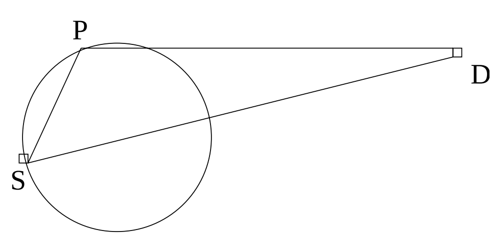


Fig. 2. A scenario that demonstrates $r_t \geq 2r_s$ is the sufficient condition to ensure coverage implies connectivity.

two dimensional surface, both the theorem and its proof apply to the three dimensional space as well. With Theorem 1 we reduce the problem of ensuring both coverage and connectivity to a much simpler problem of ensuring coverage only, and will henceforth consider only the coverage problem.

Note also that if the radio range is too large as compared to the sensing range, the network may be subject to excessive radio interference although its connectivity is ensured. Consequently Theorem 1 also suggests a minimum radio range that ensures connectivity. Most wireless devices can adjust their transmission range by adjusting their transmission power. By Theorem 1, we should set the transmission range to be twice of the sensing range.

IV. OPTIMAL SENSING COVERAGE IN THE IDEAL CASE

Recall that density control aims to find a minimum subset of sensor nodes that completely covers a given area so as to maximize the lifetime of the sensor network. Two requirements are implied here: first, the subset should completely cover the area R. Specifically, given that the coverage area of a sensor node is a disk centered at itself, we define a *crossing* as an intersection point of the circle boundaries of two disks. A crossing is said to be *covered* if it is an *interior point* of a third disk. The following theorem extracted from [12] pages 59 and 181 states a sufficient condition for complete coverage. It is also a necessary condition if we assume that the circle boundaries of any three disks do not intersect at a point. The assumption is reasonable as the probability of the circle boundaries of three disks intersecting at a point is zero, if all sensors are randomly placed. Theorem 2 is one of the important theoretical bases for our distributed density control algorithm in the next section.

THEOREM 2 *Suppose the size of a disk is sufficiently smaller than that of a region R. If one or more disks are placed within the region R, and at least one of those disks intersect another disk, and all crossings in the region R are covered, then R is completely covered.*

The second requirement is that the set of working sensors should consume as minimal power as possible so as to prolong the network lifetime. Under the assumptions that each sensor consumes the same amount of power when it is powered on and has the same sensing range, the requirement of minimizing power consumption broils down to that of minimizing the number of working sensors. Note that if sensors have different sensing ranges (e.g., using different levels of power to sense), a minimal number of working sensors does not necessarily imply minimum power consumption.

To facilitate derivation of conditions under which the second requirement is fulfilled, we first define the *overlap* at a point x the number of sensors whose sensing range can cover the point minus $I_R(x)$, where

$$I_R(x) = \begin{cases} 1 & \text{if } x \in R, \\ 0 & \text{otherwise.} \end{cases} \tag{1}$$

The intuition behind the above definition is that when N sensors can cover a point x, if the point x is in the monitored region R, then $N-1$ of them are redundant; otherwise, all of them are redundant. Next, the overlap of sensing areas of all the sensors is the integral of overlaps at all points in the area covered by all the sensors. In general, the larger the overlap of an area, the more amount of data (for environmental monitoring or target tracking, for example) will be generated (some of which may be redundant) and more power will be consumed. However, an adequate degree of redundancy may be needed to gather accurate, high-fidelity data in some cases. Although our focus in this paper is to ensure that every point is covered by at least one sensor, we will discuss how to extend our work to ensure that every point is covered by at least k sensors in Section VI.

We claim that overlap is a better index for measuring power consumption than the number of working sensors in the coverage problem for two reasons. First, while the number of working sensors is no longer directly related to power consumption in the case where different sensors have different sensing ranges, the measure of overlap is still valid, as a larger value of overlap implies more data redundancy and sensing power consumption. Second, as will be proved in the following theorem, minimizing the overlap value is equivalent to minimizing the number of working sensors in the case that all sensors have the same sensing ranges.

THEOREM 3 *If all sensor nodes (i) completely cover a region R and (ii) have the same sensing range, then minimizing the number of working nodes is equivalent to minimizing the overlap of sensing areas of all the nodes.*

Proof We prove the theorem by showing that given the conditions stated in the theorem, the number of working sensor nodes and the overlap have a linear relationship with a positive slope.

Let the indicator function of a working node i, $I_i(x)$, be defined as

$$I_i(x) = \begin{cases} 1, & \text{if } x \text{ is within the coverage area of node } i, \\ 0, & \text{otherwise.} \end{cases}$$

Let R' be a region that contains R and the coverage areas of all sensor nodes. Then the coverage area of a sensor node i is a disk with the size $\int_{R'} I_i(x)dx \stackrel{\triangle}{=} |S_i|$, where $|S_i|$ denotes the size of the area S_i covered by sensor node i. By condition (ii), $|S_i| = |S|$ for all i. With the definition of $I_i(x)$, the overlap at point x can be written as

$$L(x) = \sum_{i=1}^{N} I_i(x) - I_R(x). \quad (2)$$

If N is the number of working nodes, then the overlap of sensing areas of all the sensor nodes, L, can be written as

$$\begin{aligned} L &= \int_{R'} L(x)dx \\ &= \int_{R'} (\sum_{i=1}^{N} I_i(x) - I_R(x))dx \\ &= \sum_{i=1}^{N} \int_{R'} I_i(x)dx - |R| \\ &= N|S| - |R|, \end{aligned} \quad (3)$$

where condition (i) is implied in the first equality and condition (ii) is implied in the fourth equality. By Eq. (3), we prove that minimizing the number of working nodes N is equivalent to minimizing the overlap of sensing areas of all the sensor nodes L. □

This result is interesting because the total number of working sensor nodes is a global variable that may be difficult to obtain, while the overlapping areas between working nodes can be, in general, easily measured as a local variable. In this paper we focus on the coverage problem under the case that all sensors have the same sensing range and will consider the other case in our future work. As mentioned above, the notion of overlap can be extended to the case in which all sensors have different sensing ranges.

A. Algorithm under the Ideal Case

With Theorems 2–3, we are now in a position to discuss how to minimize the overlap of sensing areas of all the sensor nodes. Our discussion is built upon the following assumptions:

- (**A1**) The sensor density is high enough that a sensor can be found at any desirable point.
- (**A2**) The region R is large enough as compared to the sensing range of each sensor node so that the boundary effects can be ignored.

Assumption (**A2**) is usually valid. Although (**A1**) may not hold in practice, as will be shown in Section V, the result derived under (**A1**) still provides insightful guidance in designing the distributed algorithm.

By Theorem 2, in order to totally cover the region R, some sensors must be placed inside region R and their coverage areas intersect one another. If two disks A and B intersect, at least one more disk is needed to cover their crossing points. Consider, for example, Figure 3. Disk C is used to cover disks A's and B's crossing point O. In order to minimize the overlap while covering the crossing point O (and its vicinity not covered by disks A and B), disk C should also intersect disks A and B at the point O; otherwise, one can always move disk C away from disks A and B to reduce the overlap.

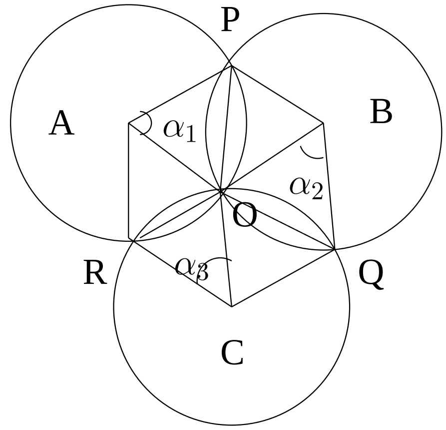


Fig. 3. An example that demonstrates how to minimize the overlap while covering the crossing point O.

Given that two disks A and B intersect, we now investigate the number of disks needed, and their relative positions, in order to cover a crossing point O of disks A and B and at the same time minimize the overlap. Take the case of three disks (Fig. 3) as an example. Let $\angle PAO = \angle PBO \triangleq \alpha_1$, $\angle OBQ = \angle OCQ \triangleq \alpha_2$, and $\angle OCR = \angle OAR \triangleq \alpha_3$. We consider two cases: (i) $\alpha_1, \alpha_2, \alpha_3$ are all variables; and (ii) α_1 is a constant but α_2 and α_3 are variables. Each of the above two cases can be extended to the case in which $k-2$ disks are placed to cover one crossing point of the first two disks (that are placed on the two-dimensional plane), and α_i, $1 \le i \le k$, are defined accordingly. Again, the boundaries of all disks should intersect at point O in order to reduce the overlap. Case (i) corresponds to a global optimization case where we can choose all the node positions, while case (ii) corresponds to a local optimization case where two nodes (A and B) are already fixed and we choose the position of a third node C to minimize the overlap. In the following discussion we assume for simplicity that the sensing range $r = 1$. Note, however, that the results are still valid even in the case of $r \neq 1$.

Case i: α_i, $1 \le i \le k$, are all variables*:* We first prove the following Lemma.

LEMMA 1

$$\sum_{i=1}^{k} \alpha_i = (k-2)\pi, \quad (4)$$

Proof: There are multiple coverage areas centered at C_i's and they all intersecting at point O. We assume that the centers of these coverage areas are labeled as C_i, with the index i increasing clockwise. (Fig. 3 gives the case of $k = 3$, where $C_1 = A$, $C_2 = B$, and $C_3 =$

C.) Now we have $\sum_{i=1}^{k} \angle C_i O C_{(i \mod k)+1} = 2\pi$ and $\angle C_i O C_{(i \mod k)+1} + \alpha_i = \pi$. From the above equations, we have $\sum_{i=1}^{k} \alpha_i = (k-2)\pi$. □

Now the overlap between the i^{th} and $(i \bmod k+1)^{th}$ disks is $(\alpha_i - \sin\alpha_i)$, $1 \le i \le k$. If we ignore the overlap caused by *non-adjacent* disks (note that disks i and $(i \mod k+1)$ are *adjacent*), then the total overlap is $L = \sum_{i=1}^{k}(\alpha_i - \sin\alpha_i)$. Hence the coverage problem can be formulated as

PROBLEM 1

$$\text{minimize} \sum_{i=1}^{k}(\alpha_i - \sin\alpha_i),$$
$$\text{subject to} \sum_{i=1}^{k} \alpha_i = (k-2)\pi. \quad (5)$$

The Lagrangian multiplier method can be used to solve the above optimization problem. The solution is $\alpha_i = (k-2)\pi/k, i = 1, 2, \cdots, k$ and the resulting minimal overlap using k disks to cover the crossing point O is

$$L(k) = (k-2)\pi - k\sin(\frac{(k-2)\pi}{k}) = (k-2)\pi - k\sin(\frac{2\pi}{k}).$$

Note that the overlap per disk

$$\frac{L(k)}{k} = \pi - \frac{2\pi}{k} - sin(\frac{2\pi}{k}) \quad (6)$$

monotonically increases with k when $k \ge 3$. Moreover when $k = 3$ (which means that we use one disk to cover the crossing point), the optimal solution is $\alpha_i = \pi/3$ and there is no overlap between *non-adjacent* disks. No matter whether there exist overlaps between *non-adjacent* disks when $k > 3$, the overlap per disk is always higher than that in the case of $k = 3$. This implies that using one disk to cover the crossing point and its vicinity is optimal in the sense of minimizing the overlap. Moreover, the centers of the three disks should form a equilateral triangle with edge $\sqrt{3}$. We state the above result in the following theorem.

THEOREM 4 *To cover one crossing point of two disks with the minimal overlap, only one disk should be used and the centers of the three disk should form a equilateral triangle with side length $\sqrt{3}r$, where r is the radius of the disks.*

Case 2: α_1 is a constant, while α_i, $2 \le i \le k$, are variables: In this case the problem can still be formulated as in Problem 1, except that α_1 is fixed. The Lagrangian multiplier method can again be used to solve the problem, and the optimal solution is $\alpha_i = ((k-2)\pi - \alpha_1)/(k-1)$, $2 \le i \le k$. Again a similar conclusion can be reached that using one disk to cover the crossing point gives the minimal overlap. We state the result in the following theorem.

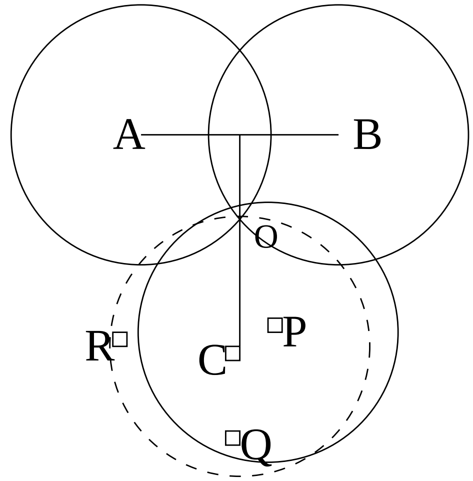


Fig. 4. Although C is the optimal place to cover the crossing O of A, B, there is no sensor node there. The node closest to C, P, is selected to cover the crossing O.

THEOREM 5 *To cover one crossing point of two disks whose positions are fixed (i.e., α_1 is fixed in Fig. 3), only one disk should be used and $\alpha_2 = \alpha_3 = (\pi - \alpha_1)/2$.*

In summary, to cover a large region R with the minimal overlap, one should ensure (i) at least one pair of disks intersects; (ii) the crossing points of any pair of disks are covered by a third disk; (iii) if the positions of any three sensor nodes are adjustable, then as stated in Theorem 4 the three nodes should form an equilateral triangle with side length $\sqrt{3}r$. If the positions of two sensor nodes A and B are already fixed, then as stated in Theorem 5 the third sensor node should be placed on the line that is perpendicular to the line connecting nodes A and B and have a distance r to the intersection of the two circles (e.g., the optimal point in Fig. 4 is C). These conditions are optimal for the coverage problem in the ideal case in which assumptions (**A1**) and (**A2**) hold.

V. OPTIMAL GEOGRAPHICAL DENSITY CONTROL ALGORITHM

In this section, we propose a completely localized density control algorithm, called OGDC, that makes use of the optimal conditions derived in Section IV. Note that as it may not be possible to locate sensor nodes in any desirable position (i.e., assumption (**A1**) may not hold), OGDC attempts to select sensor nodes that are as close to optimal locations as possible to be the working nodes. We first give an overview of OGDC and then delve into its detailed operations. We will also discuss some of its possible extension and limitation.

A. Overview

OGDC is devised under the following assumptions:

(**B1**) The radio range is at least twice of the sensing range. As discussed in Section III, under this assumption complete coverage implies connectivity.

(**B2**) Each node is aware of its own position. This assumption is not impractical, as many research efforts have been made to address the localization problem [25], [21], [7].

(**B3**) For clarity of algorithm discussion, we assume all sensor nodes are time synchronized. We will relax this assumption in Section V-C.

At any time, a node is in one of the three states: "UNDECIDED," "ON," and "OFF." Time is divided into rounds. At the beginning of each round, all the nodes wake up, set their states to "UNDECIDED," and carry out the operation of selecting working nodes. By the end of the execution, all the nodes change their states to either "ON" or "OFF" and remain in that state until the beginning of the next round. The length of each round is so chosen that it is much longer than the time it takes to execute OGDC but much shorter than the average sensor lifetime. Our simulation results show that the time it takes to execute the node selection operation for networks of size up to 1000 nodes in an area of $50 \times 50\mathrm{m}^2$ (and for timer values appropriately set) is usually well below 1 second and most nodes can decide their states (either "ON" or "OFF") in less than 0.2 second from the time instant when at least one node volunteers to be a starting node. The interval for each round is usually set to approximately 1000 seconds, and the overhead of density control is small ($\lesssim$ 1%).

The process of selecting working nodes (in a decentralized manner) in each round commences by randomly selecting a sensor node A to be the starting node (Fig. 4). Then one of its neighbors with an (approximate) distance of $\sqrt{3}r$, B, is selected to be a working node. To cover the crossing point of disks A and B, the node whose position is closest to the optimal position C (e.g., node P in Fig. 4) is then selected, in compliance with Theorem 5, to become a working node. The process continues until all the nodes change their states to either "ON" or "OFF," and the set of nodes with "ON" states forms the working set. As the starting node in each round is randomly selected, the set of working sensor nodes is not likely to be the same in each round. This ensures uniform (and minimal) power consumption across the network, as well as complete coverage and connectivity. In what follows, we give the detailed description of OGDC.

B. Detailed Description of OGDC

Selection of the starting node: At the beginning of each round, every node is powered on with the "UNDECIDED" state. A node volunteers to be a starting node with probability p if its power exceeds a pre-determined threshold P_t. The power threshold P_t is related to the length of the round and in general is set to a value so as to ensure with high probability the sensor can remain powered on until the end of the round.

If a sensor node volunteers, it sets a backoff timer of τ_1 seconds, where τ_1 is uniformly distributed in $[0, T_d]$. When the timer expires, the node changes its state to "ON", and broadcasts a power-on message. If a node hears other power-on messages before its timer expires, it cancels its timer and does not become a starting node. The use of backoff timers avoids the possibility of multiple neighboring nodes volunteering themselves to be the starting node in a round. The selection of T_d is a tradeoff between the performance and the latency. Using a large value of T_d can reduce the number of starting nodes in the network and possibly reduce the overlap. However, with fewer starting nodes, it will take a longer time to complete the operations of working node selection. In our simulation, we select T_d to be about 1.5 times of the transmission time of a power-on packet. The power-on message sent by the starting node contains (i) the position of the sender and (ii) the direction α along which the second working node should be located. This direction is randomly generated from a uniform distribution in$[0, 2\pi]$.

If the node does not volunteer itself to be a starting node, it sets a timer of T_s seconds. T_s should be set to a sufficiently large value such that the operation of selecting working nodes can complete if there is at least one starting node. In our simulation we set it to be $100T_d$. When the timer T_s[1] expires, it repeats the above volunteering process with the value of p doubled until the value reaches 1. The timer is canceled whenever the state of a node is changed to "ON" or "OFF." The value of p is initially set to p_0. We will discuss how to determine the value of p_0 in Section V-C.

Actions taken when a node receives a power-on message: When a sensor node receives a power-on message, if its power is less than the power threshold P_t, it turns itself off and sets its state to "OFF"; otherwise, it checks whether or not all its neighbors completely cover its coverage area. (We will discuss how this can be effectively tested in Section V-C.) If so, it sets its state to "OFF" and turns itself off. If the node is more than $2r_s$ away from the sender node of the power-on message, it ignores the message; otherwise it adds this sender to its neighbor list and takes actions according to the following rules:

R1 *The message is the first power-on message received and is from a starting node.* A node tells whether or not a message was sent from a starting node by the value set in the direction field α: if $\alpha \geq 0$, then the message was originated from a starting node. In this case, the node sets a timer of T_{c1} seconds. When the timer expires, the node sets its state to "ON" and broadcasts a power-on message (with α set to a negative value). If the node hears any other power-on message before its timer expires, it carries out

[1] With a little abuse of symbols, we will use T_s to refer both the timer and the value of the timer. This applies to other timers.

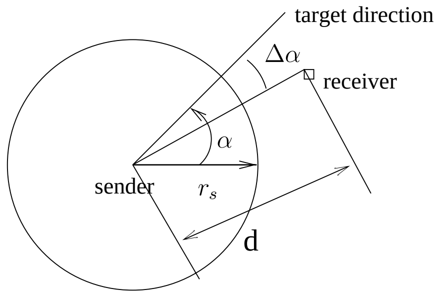


Fig. 5. A scenario that demonstrates how the value, T_{c1}, of the backoff timer is set, when a node receives a power-on message from the starting node.

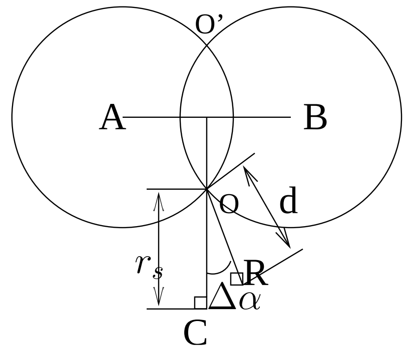


Fig. 6. A scenario that demonstrates how the value T_{c2} are set (in case R2) when a node receives two power-on messages.

the operations specified in rule (**R3**).

The value, T_{c1}, of the backoff timer is set as

$$T_{c1} = \begin{cases} t_0(c((\sqrt{3}r_s - d)^2 + (d\Delta\alpha)^2) + u), \\ \quad \text{if } d <= \sqrt{3}r_s; \\ t_0(c((\sqrt{3}r_s - d)^2 + (d\Delta\alpha)^2 + \ell) + u), \\ \quad \text{otherwise,} \end{cases} \tag{7}$$

where t_0 is the time it takes to send a power-on message, c is a constant that determines the backoff scale and is set to $10/r_s^2$ in our simulation, d is the distance from the sender to the receiver, $\Delta\alpha$ is the angle between α and the direction from the sender to the receiver (Fig. 5), u is a random number drawn from the uniform distribution $[0, 1]$, and ℓ is a constant. Note that by assumption (**B2**), the node can determine the values of d and $\Delta\alpha$.

T_{c1} includes two terms: a deterministic term $(c(\sqrt{3}r_s - d)^2 + (d\Delta\alpha)^2)$ and a random term (u). If the receiver is right in the direction α and its distance to the starting node is $\sqrt{3}r_s$, the deterministic term is 0; otherwise, $c((\sqrt{3}r_s - d)^2 + (d\Delta\alpha)^2)$ roughly represents the deviation from the optimal position and a delay is introduced in proportion of this deviation. The constant ℓ is introduced to discourage a node with distance $d > \sqrt{3}r_s$ from becoming a working node. The random term is introduced to break ties in the case that there exist nodes whose positions yield the same value of the deterministic term.

R2 *The message is the first power-on message received but is from a non-starting node.* If the value carried in the direction field, α, of the power-on message is negative, then the message was originated from a non-starting node. In this case, the node sets a timer of T_e seconds, where the value of T_e is much larger than that of T_{c1}. (In our simulation, we set $T_e = 0.2T_s$.) This is because when a node receives such a power-on message, it expects to receive another power-on message and the coverage areas of the two senders are expected to overlap. If upon timer expiration no new power-on message arrives, this usually indicates there do not exist sufficient nodes with power $\geq P_t$ which can participate density control process. In this case, the node still sets its state to “ON” and broadcast a power-on message. On the other hand, if a new power-on message arrives before its timer expires, the node carries out the operations specified in rule (**R3**). Note that the value of T_e cannot exceed that of T_s; otherwise, before the T_e timer expires, the node may double its value of p and volunteers itself as a starting node.

R3 *The message is the second power-on message received.* If the coverage areas of the two senders do not intersect, the node simply ignores the new message but retains the timer set for the first power-on message; otherwise, it cancels the existing T_e or T_{c1} timer and sets a timer of T_{c2} seconds. The value of T_{c2} is calculated as follows. As shown in Figure 6, let O denote the crossing point of the coverage areas of the two senders, C the optimal location of a third sensor node to cover the crossing point O, R the location of the receiver node, d the distance between the node and the crossing point O, and $\Delta\alpha$ the angle between $\overline{OC}$ and $\overline{OR}$. Then the value of T_{c2} is

$$T_{c2} = \begin{cases} t_0(c((r_s - d)^2 + (d\Delta\alpha)^2)) + u), \\ \quad \text{if } d < r_s; \\ t_0(c((r_s - d)^2 + (d\Delta\alpha)^2 + \ell) + u), \\ \quad \text{otherwise,} \end{cases} \tag{8}$$

where t_0, c, u and ℓ are the same as those defined in Eq. (7). How the value of T_{c2} is calculated can be reasoned in the same manner as T_{c1} and will not be elaborated on again.

When the timer expires, the node sets its state to “ON” to cover the crossing point O and broadcasts a power-on message with α set to a negative value.

If the node receives any other power-on message before its timer expires, it carries out the operations specified in (**R4**).

R4 *More than two power-on messages have been received.* This case can be further divided into four subcases which are also illustrated in Fig. 7:

a) None of the coverage areas of the senders overlaps with each other (Fig. 7(a)): This occurs when multiple starting nodes propagate their power-on messages to the same receiver. In such a case, the receiver will ignore the power-on message received last but retain the the first timer T_{c1} (or T_e) set for the first power-on message.

b) The coverage areas of the previous senders do not overlap but the coverage area of the new sender overlaps with at least one of the previous two coverage areas (Fig. 7(b)): This occurs when there exist several starting nodes and the new sender and at least one other sender receive power-on messages propagated from the same starting node. In this case, the receiver first cancels the timer T_{c1} (or T_e) set for the first power-on message. Then it finds the closest crossing points produced by the new sender and the other intersecting neighbors and takes action as if it only received two power-on messages which create the closest crossing point.

c) The coverage areas of the previous senders overlap. The receiver plans to cover a crossing point upon its timer (T_{c2}) expiration, and the coverage area of the new sender does not cover that crossing point (Fig. 7(c)): This occurs when the first two power-on messages are propagated from the same starting node but the new power-on message originates from a different starting node. In this case the receiver simply ignores the message received last but retains the timer T_{c2}.

d) The coverage areas of the previous senders overlap. The receiver plans to cover a crossing point upon its timer (T_{c2}) expiration but the coverage area of the new sender covers that crossing point; or the receiver does not have a timer T_{c2} (Fig. 7(d)): The first scenario occurs when both the new sender and the receiver attempt to cover the same crossing point but the timer T_{c2} of the new sender expires first and the new sender sends a power-on message. In this case, the receiver will cancel its existing timer T_{c2} if it exists and selects the closest crossing point of the coverage areas of the new sender and any previous sender that has not been covered by any of its existing neighbors. Then it take action as if it only received the two power-on messages that create the closest crossing point that is not yet covered. In the case that no such crossing point exists, it simply waits until it receives a new power-on message or its T_s timer expires. In the latter case, when the next new power-on message arrives, the current receiver will not have a T_{c2} timer, although the coverage areas of the previous two senders overlap.

Recall that every node has a timer T_s which is canceled if and only if its state changes to “ON” or “OFF” from ”UNDECIDED.” Upon the timer expiration of T_s the node will double its probability p and initiates the volunteering process (i.e., with probability p, the node volunteers to be a starting node). The process may repeat until the value of p reaches 1, at which point the node volunteers. Note that although in cases **R4** (a) and **R4** (c) the receiver ignores the power-on message sent by the new sender, the new sender will eventually be covered by the receiver, because as long as the coverage area of the receiver is not completed covered, its state will remain in ”UNDECIDED” and eventually become “ON” due to the timer T_s.

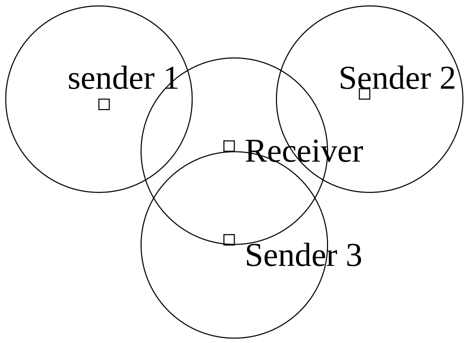
C. Discussion

After describing the operations of OGDC, we are now in a position to elaborate on several implementation and parameter tuning issues:

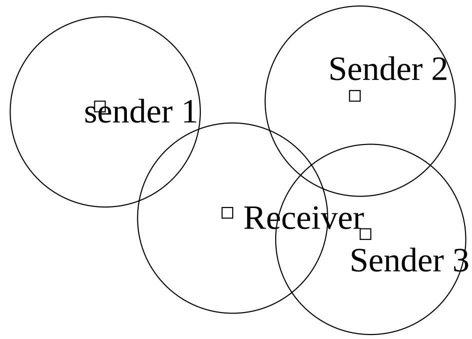
Setting of the initial volunteering probability, p_0: Recall that p_0 is the initial probability that a node volunteers itself as a starting node. In the case that the region to be covered is not large, it is desirable that at any one time only one node determines to be a starting node. To this end, we set $p_0 = 1/N$, where N is the total number of sensor nodes in the network, as this maximizes the probability that exactly one sensor node volunteers itself as a starting node. On the other hand, if the region to be covered is large, it is desirable to have multiple sensor nodes volunteer themselves at the same time. In this case, we set $p_0 = k/N$ as this maximizes the probability that exactly k nodes volunteers themselves. We argue that the number of sensor nodes, N, or at least its order is usually known at the time of network deployment. If not, as the value of p will be doubled every time the T_s timer expires, the value of p_0 may not have a significant impact on the performance.

Guidelines of OGDC parameter tuning: OGDC has several tunable parameters. We have briefly described how to set the value of each parameter when it is mentioned for the first time. Here we outline the set of guidelines for parameter tuning. Table I list the parameters, their functions, and their values used in our simulation study.

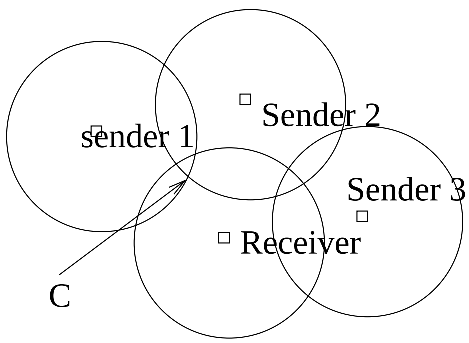
Most timing related parameters such as T_d, T_s, T_e, c and l should be set according to the transmission time of a power-on message t_0. As a rule of thumb, the T_d timer used to suppress surplus starting nodes should be in the same order of t_0. The T_s timer should be set to approximately two orders of magnitude larger than t_0 to allow the density control process to complete before the T_s timer fires if there exist some



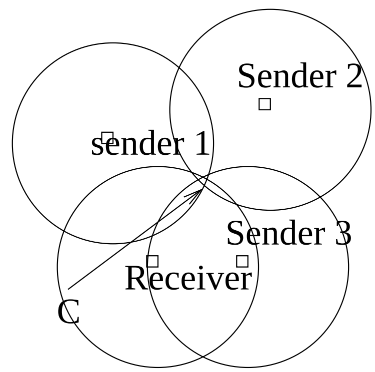
(a) None of the coverage areas of the senders overlap.



(b) The coverage areas of the previous senders (sender 1 and 2) do not overlap but the coverage area of the new sender (sender 3) overlaps with at least one of the previous two.



(c) The crossing point C of the coverage areas of the previous senders (sender 1 and 2) is not covered by the coverage area of the new sender (sender 3).



(d) The crossing point C of the coverage areas of the previous senders (sender 1 and 2) is covered by the coverage area of the new sender (sender 3).

Fig. 7. The subcases in case R4: more than two power-on messages have been received by a node.

TABLE I
PARAMETER VALUES USED IN THE SIMULATION STUDY.

Parameter	Function	Value set
r_s	sensing range	10 m
round time	period for executing OGDC	1000 s
P_t	power threshold for acting as a working node	the level that allows a node to be idle for 900 seconds
T_d	timer value used in volunteering to be a starting node	10 ms
T_s	timer value used in the starting node selection process	1 s
T_e	timer value used when receiving the first power-on message from a non-starting node	200 ms
t_0	the time it takes to send a power-on packet	6.9 ms
c	constant used in Eqs. (7) and (8)	$\frac{10}{r_s^2}$
ℓ	constant used in Eqs. (7) and (8)	$\frac{r_s^2}{4}$
channel capacity		40K bps

starting nodes in the network. The T_e timer should be set to at least one order of magnitude larger than t_0, but much less than T_s. c should be chosen such that T_{c1} and T_{c2} are about one order of magnitude larger than t_0 on average to avoid packet collision. l should be set to a value so that its contribution to the timer value T_{c1} or T_{c2} is around twice of t_0's contribution. The round time should be set to a value that is approximately one order of magnitude less than that of the lifetime of a single sensor.

The value of P_t is dependent on the application requirement. If the application strongly requires that full coverage be continuously maintained, P_t should be set to a value such that a sensor can remain active for at least the duration of a round time. If intermittent, incomplete coverage in each round is acceptable, P_t can be set to a value that is less than the power required to keep the sensor active for the entire round time.

It is worth mentioning that we follow the above guidelines to tune parameters in our simulation and the simulation results are quite promising. Moreover, the performance of OGDC is not particularly susceptible to parameter settings as long as the above guidelines are followed.

Time synchronization: For simplicity of algorithm discussion, we assume that all nodes are time synchronized ((**B3**)). Several time synchronization schemes for sensor networks have been proposed recently ([8], [9], [10]). Elson and Estrin reported in [8] that time synchronization with the precision on the order of $1\mu sec$ can be achieved with very low energy cost. Since all the timers used in OGDC is on the order of $1ms$, time synchronization algorithms reported in the literature can be readily used.

On the other hand, the clock synchronization assumption can be relaxed as follows. In the first round we designate a sensor node to be the starting node. When the starting node sends a power-on message, instead of telling its neighbors what time they should wake up for the next round, it tells until when (δT) they should wake up for the next round. When a non-starting node broadcasts a power-on message, it reduces the value of δT by the time elapsed since it receives the last power-on message and includes the new value of δT in its power-on message. In this fashion, at the beginning of the next round, all the nodes will wake up.

If the monitored region is so large that it is not acceptable to have one starting node in a round, we can synchronize a few nodes before deployment, distribute them evenly in the entire region, and designate them to be the starting node in the first round. In fact it is not unreasonable to assume that multiple synchronized nodes with overlapping coverage areas can serve as reference points of other nodes ([3], [4]).

What if no other sensor nodes volunteer: It may occur that the power of a node is less than the threshold power P_t and yet no power-on message is received even after the node sets the value of p to 1. This indicates that all the nodes do not have sufficient power and cannot volunteer themselves to be starting nodes. In this case, the node resets its power threshold P_t to 0 and restarts the density control process.

How does a node monitor whether or not its coverage area has been completely covered: In order for a node to determine whether or not its own coverage area is completely covered by all the working, neighboring nodes, each sensor node divides its own coverage area into small grids and uses a bitmap to indicate whether the center point of a grid is covered by some other working node(s). Each time when a node hears a power-on message, it updates the bitmap. If every grid center point is covered and its state is "UNDECIDED," it sets its state to "OFF" and turns itself off.

Packet collision: Because of the backoff timer mechanism employed in OGDC, packet collision rarely occurs in a low/medium-density network, but does occur if the node density is high. Packet collision may have an adverse effect on OGDC, as a node will not be able to update its coverage bitmap in the case of packet collision, and may not turn itself off properly. To deal with this problem, we enable a sensor node to turn itself off when it overhears packet collision. This is because the fact that a node overhears packet collision indicates a high likelihood that the node is located in a high-density area. Our simulation results have shown that turning nodes off in the case that they overhear packet collision will not impair full coverage.

Computational efforts: The major computation incurred in OGDC is to calculate the crossing points of the coverage areas of two neighbors. Since each node only maintains the information of its working neighbors that are within twice of its sensing range before it decides to set its state "ON" or "OFF," due to density control most nodes can decides its state in each round after it hears a few power-on messages. In our simulation, it rarely occurs that a node need to maintain the information of more than 5 nodes before it can decide its state. For every pair of neighbors, only if their distance is less than $2r_s$ will their coverage areas intersect. Calculating the crossing point given the locations of two senders requires only a few steps of algebraic operations which we give in in Appendix .

VI. Performance Evaluation

A. Simulation Environment Setup

To validate and evaluate the proposed design of OGDC, we have implemented it in *ns-2* [1] with the CMU wireless extension, and conducted a simulation study.

Schemes for comparison: In addition to evaluating OGDC, we also evaluate as a baseline the performance of the PEAS algorithm proposed by Ye *et al.* [32] and a hexagon-based GAF-like algorithm. The hexagon-based GAF-like algorithm is built upon GAF [29] and operates as follows. The entire area can be divided into square grids and one node is selected to be awake in each grid. To maintain coverage, the grid size must be less than or equal to $r_s/\sqrt{2}$. Thus, for a large area with size $l \times l$, it requires $\frac{2l^2}{r_s^2}$ nodes to operate in the active mode to ensure full coverage. (Note that we ignore the boundary effects as we assume the area is sufficiently large.) As pointed out by [16], hexagonal grids are more "homogeneous" than square grids and thus offer more scaling benefits, e.g., the number of working nodes is significantly less. To maintain coverage in hexagonal grids, the side length of each hexagon is at most $r_s/2$, and it requires $\frac{8l^2}{3\sqrt{3}r_s^2} \approx \frac{1.54l^2}{r_s^2}$ working nodes to fully cover a large area with size $l \times l$ (again the boundary effects are ignored). As will be discussed below, the hexagon-based GAF-like algorithm performs better than the "sponsored area" algorithm proposed in [28], and hence the latter is not included in the comparison.

Parameters used: We use the energy model in [32], where the power consumption ratio for transmitting, receiving (idling) is 5:1. We ignore power consumption in the sleep mode because it is very small. We define one unit of energy (power) as that required for a node to remain idle for 1 second. Each node has a sensing range of $r_s = 10$ meters, and a lifetime of 5000 seconds if it is idle all the time.

The tunable parameters in OGDC are set as follows: the round time is set to 1000 seconds, the power threshold P_t is set to the level that allows a node to be idle for 900 seconds, the timer values are set to, respectively, $T_d = 10$ ms, $T_s = 1$ s, and $T_e = T_s/5 = 200$ ms, t_0 is set to the time it takes to send a power-on packet, $6.9s$ (the wireless communication capacity is 40Kbps, the packet size is 34 bytes). The constants used in Eqs. (7) and (7) are set to, respectively, $c = \frac{10}{r_s^2}$ and $\ell = \frac{r_s^2}{4}$, and p_0 is set to $1/N$ where N is the total number of sensors. Table I give all the parameter values used.

Although OGDC involves tuning of several parameters, we have found that its performance is rather insensitive to the parameter values, as long as they are set in compliance with the guidelines discussed in Section V-C. Each data point reported below is an average of 20 simulation runs unless specified. In each run all the sensors are uniformly distributed in the area to be monitored. The same values of system parameters are used for each node, such as the initial energy of each node, the radio transmission rate, and the energy consumption rate.

B. Performance metrics

The performance metrics of interest are (i) the percentage of coverage, i.e. the ratio of the covered area to the total area to be monitored; (ii) the number of working nodes required to provide the percentage of coverage in (i); and (iii) α-lifetime, defined as the total time during which α portion of the total area is covered by at least one node. The conventionally defined network lifetime is then 100%-lifetime. Note that the lifetime definition used in this paper is slightly different from that in [31], where the lifetime is defined as the time interval until which coverage falls below a pre-determined percentage and never comes back again.

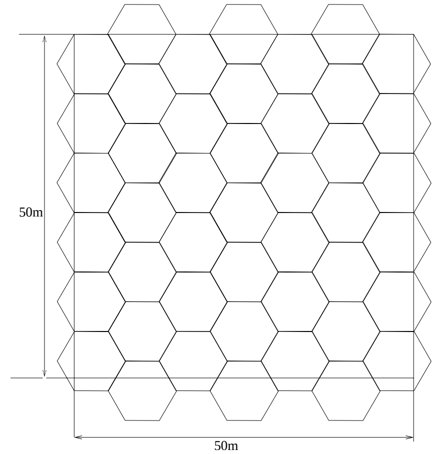
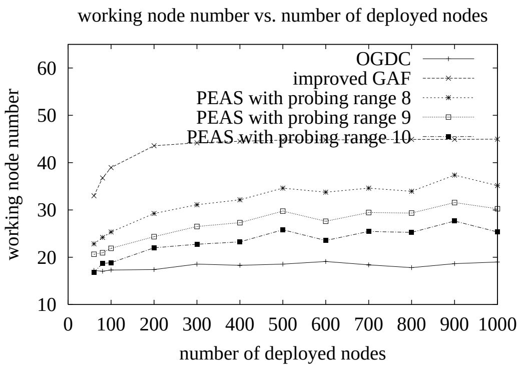


Fig. 8. 45 hexagons are required to cover a 50 × 50 m^2 area.

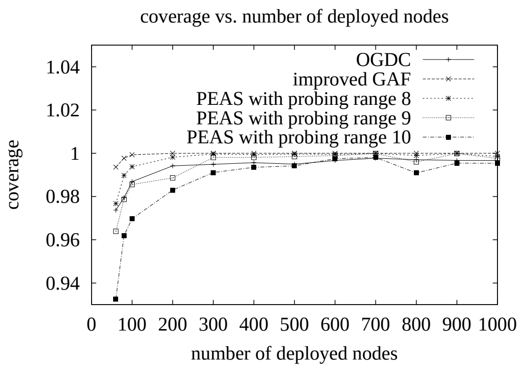
C. Simulation Results

We have conducted our simulation study in a $50 \times 50\mathrm{m}^2$ area. Coverage is measured as follows: we divide the area into 50 × 50 square grids. A grid is considered covered if the center of the grid is covered, and coverage is defined as the ratio of the number of grids that are covered by at least one sensor to the total number of grids. For this $50 \times 50\mathrm{m}^2$ area, 45 hexagon cells are required to cover the entire area if the hexagon-based GAF-like algorithm is used (Fig 8). Hence, the hexagon-based algorithm ensures 100% coverage if at least 45 sensors operate in the active mode in each round, one for each cell. Similarly at least 47 nodes are required to operate in the active mode under the "sponsored area" algorithm proposed in [28] to ensure the complete coverage. When the number of sensor nodes in the sensor network increase, the sponsored area algorithm requires more nodes to cover the entire area. As the sponsored area algorithm performs worse than the hexagon-based, GAF-like method, we do not include the sponsored area algorithm [28] in the following comparison.

Fig. 9 gives the curves of the number of working nodes and coverage versus the number of sensor nodes in the network. Both metrics are measured after the density control process completes. Under most cases, OGDC takes less than 0.2 seconds to perform density control in each round, while the PEAS approach [32] may take more than 100 seconds. As shown in Fig. 9, OGDC needs only half as many nodes to operate in the active mode as compared to the hexagon-based GAF-like algorithm, but achieves almost the same coverage. In most cases OGDC achieves more than 99.5% coverage. The PEAS algorithm, on the other hand, can control the



(a) # of working nodes

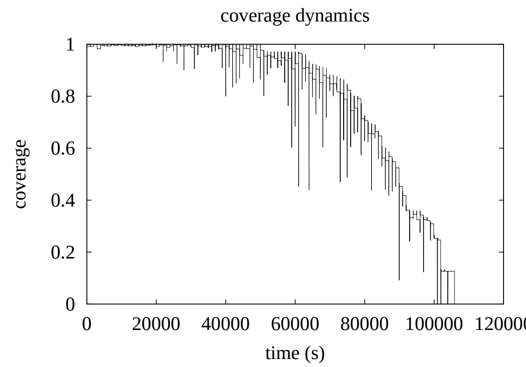


(b) Coverage

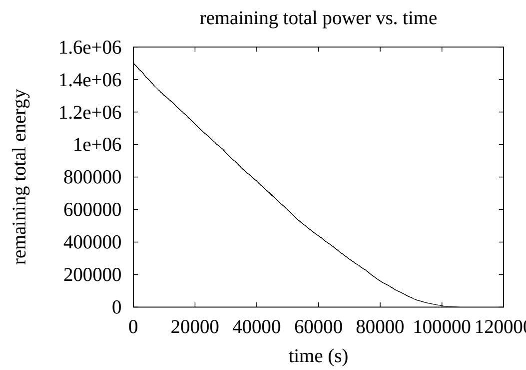
Fig. 9. # of working nodes and coverage versus # of sensor nodes in a 50 $\times$ 50 m^2 area.

number of working nodes by using different probing ranges. We tried three different probing ranges: 8m, 9m and 10m. As shown in Fig. 9, using a small probing range (8m) gives an excessive number of working nodes, while using a large probing range (9m or 10m) results in insufficient coverage (especially under low density cases which are of more interest because under high density cases all the schemes achieve over 99% coverage). Even if a probing range of 9m is used, the resulting coverage is less that that achievable by OGDC while the number of working nodes is up to 50% more than that of OGDC. Moreover, the number of working nodes required under OGDC modestly increases with the number of sensor nodes deployed, while a 50% increase in the number of working nodes is observed under both the PEAS algorithm [32] and the sponsored area algorithm [28], when the number of sensor nodes deployed in the network increases from 100 to 1000.

Fig. 10 gives the dynamics of the coverage and the total remaining power over the time in a typical simulation run for a sensor network of 300 sensor nodes in a 50 $\times$ 50 m^2 area. OGDC can provide over 95% coverage for appropriately 10 times of the lifetime of a single sensor node and the total power of the network decreases smoothly.



(a) Sensing coverage



(b) Total remaining power

Fig. 10. Dynamics of the sensing coverage and total remaining power versus time in a sensor network of 300 sensor nodes in a 50 $\times$ 50 m^2 area.

Fig. 11 compares the α-lifetime achieved by OGDC and PEAS in a sensor network of 300 nodes, where α varies from from 98% to 50%. For the PEAS algorithm we tried three different probing ranges: 8m, 9m and 10m. As shown in Fig. 11, with the probing range of 8m, the α-lifetime of PEAS is always less than that of OGDC, while with larger probing ranges (9m and 10m), the α-lifetime of PEAS is much shorter than that of OGDC for high α and longer than that of OGDC for low α. This is because with a relatively small probing range, PEAS requires an excessive number of nodes to operate simultaneously. Hence, its lifetime is consistently shorter than OGDC. On the other hand, with large probing ranges 9m and 10m, PEAS only guarantees that no two working nodes are in each other's probing range and does not ensure full coverage. Moreover, when a node dies, it may take more than 100 seconds for another node to wake up to take its place. During that period the network is not fully covered. As a result, the low percentage lifetime is prolonged in PEAS. A nice property of OGDC is that during most of the lifetime, the monitored region is covered with high percentage. It is clear that OGDC is preferred to PEAS no matter what probing

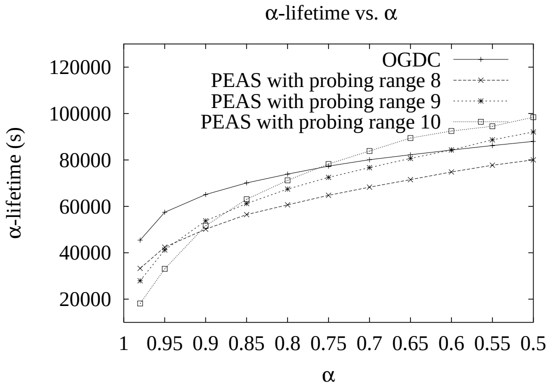


Fig. 11. Comparison of α-lifetime versus α under OGDC and PEAS.

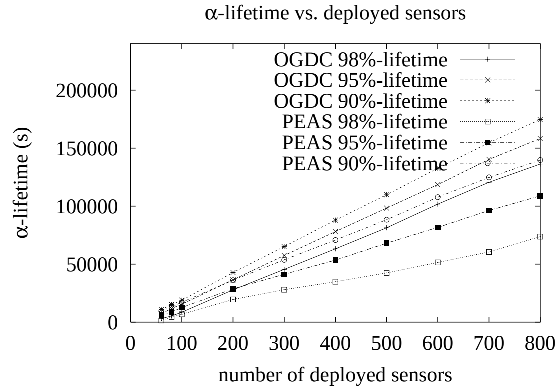


Fig. 12. Comparison of α-lifetime versus number of sensor nodes under OGDC and PEAS (with probing range 9m).

range is used, unless the desired coverage percentage is less than 80%.

Fig. 12, on the other hand, gives the 98%-lifetime, 95%-lifetime and 90%-lifetime under OGDC and PEAS with a probing range of 9m, when the number of sensor nodes deployed in a network varies from 60 to 800. The α-lifetime scales linearly as the number of sensors deployed increases for both algorithms. However, OGDC achieves nearly 100% more 98%-lifetime, 50% more 95%-lifetime and 40% more 90%-lifetime than PEAS does.

For applications that require high levels of accuracy and reliability, it may be desirable to have multiple sensors cover a single point. To this end, we define k-coverage as that each point in an area is covered by at least k nodes. We claim that OGDC can be readily extended to accommodate k-coverage by the following modification: a node is only turned off when each grid point in the node's coverage area is covered by at least k other nodes. Figure 13 shows the curve of 80%-lifetime for 3-coverage versus the number of sensor nodes. Again the 80%-lifetime linearly increases as the number of sensor nodes deployed in the network increases. A more in-depth study on k-coverage is a subject of our future research.

VII. Conclusions and Future Works

In this paper we investigate the issues of maintaining coverage and connectivity by keeping a minimal number of sensor nodes to operate in the active mode in wireless sensor networks. We begin with a discussion on the relationship between coverage and connectivity, and show that if the radio range is at least twice of the sensing range, then complete coverage implies connectivity. Hence, if the condition holds, we only need to consider the coverage problem. Then, we derive, under the ideal case in which node density is sufficiently high, a set of optimality conditions under which a subset of working sensor nodes can be chosen for full coverage. Based on the optimality conditions, we then devise a decentralized and localized density control algorithm, OGDC. *Ns-2* simulation shows that OGDC outperforms the PEAS algorithm [32], the hexagon-based GAF-like algorithm, and the sponsor area algorithm [28] with respect to the number of working nodes needed (sometimes at a 50% improvement), and achieves almost the same coverage as the best algorithm.

In OGDC, each node needs to know its own location. However, we claim that this requirement can be relaxed to that each node knows its relative location to its neighbors. We are in the process of verifying this claim. As mentioned in Section VI, we would also like to look into the issue of k-coverage and its impact on fault tolerance. Also, to better evaluate OGDC (or other density control algorithms), we need to derive the upper bound of the network lifetime in large areas. This problem is a subject of our future research.

References

[1] ns-2 network simulator. *http://www.isi.edu/nsnam/ns*.

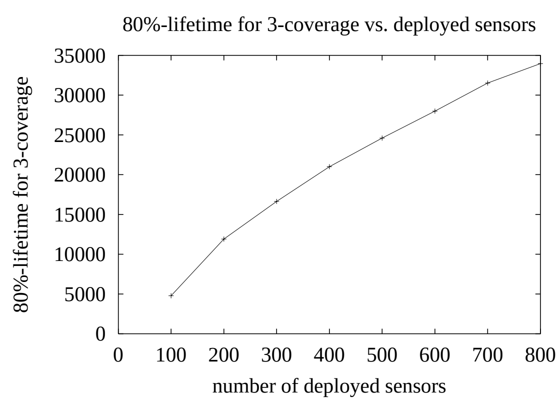


Fig. 13. $80\% - lifetime$ for 3-coverage versus number of sensor nodes under OGDC.

[2] I. F. Akyildiz, W. Su, Y. Sankarasubramaniam, and E. Cayirci. *Wireless Sensor Networks: A Survey, Computer Networks*. March 2002.

[3] N. Bulusu, J. Heidemann, and D. Estrin. GPS-less low cost outdoor localization for very small devices. *IEEE Personal Communications Magazine*, 7(5):28–34, October 2000.

[4] Nirupama Bulusu. *Self-Configuring Localization Systems*. PhD thesis, University of California, Los Angeles, 2002.

[5] A. Cerpa and D. Estrin. Ascent: Adaptive self-configuring sensor networks topologies. In *Proc. of Infocom 2002*.

[6] B. Chen, K. Jamieson, H. Balakrishnan, and R. MOrris. Span: An energy-efficient operation in multihop wireless ad hoc networks. In *Proc. of ACM MobiCom'01*, 2001.

[7] L. Doherty, L. El Ghaoui, and K. S. J. Pister. Convex position estimation in wireless sensor networks. In *Proc. of IEEE Infocom 2001*, Anchorage, AK, April 2001.

[8] J. Elson and D. Estrin. Time synchronization for wireless sensor networks. In *Proc. of IPDPS, workshop on Parallel and Distributed Computing Issues in Wireless Networks and Mobile Computing*, April 2001.

[9] J. Elson, L. Girod, and D. Estrin. Fine-grained network time synchronization using reference broadcasts. Submitted for review, May 2002.

[10] J. Elson and K. Romer. Wireless sensor networks: A new regime for time synchronization. In *First Workshop on Hot Topics in Networks*, Oct. 2002.

[11] D. Estrin, R. Govindan, J. S. Heidemann, and S. Kumar. Next century challenges: Scalable coordination in sensor networks. In *Proc. of ACM MobiCom'99*, Washington, August 1999.

[12] P. Hall. *Introduction to the Theory of Coverage Processes*. 1988.

[13] J. Hill, R. Szewczyk, A. Woo, S. Hollar, D. Culler, and K. Pister. System architecture directions for networked sensors. In *Architectural Support for Programming Languages and Operating Systems*, 2000.

[14] C. Intenagonwiwat, R. Govindan, and D. Estrin. Directed diffusion: A scalable and robust communication paradigm for sensor networks. In *Proc. of ACM MobiCom'01*, 2001.

[15] J. M. Kahn, R. H. Katz, and K. S. J. Pister. Next century challenges: Mobile networking for "smart dust". In *Proc. of ACM MobiCom'99*, August 1999.

[16] S. Kandula and J. C. Hou. Hierarchical clustering for data-centric sensor networks. Technical report, University of Illinois at Urbana, Champaign, August 2002.

[17] J. Kulik, W. Rabiner, and H. Balarkishnan. Adaptive protocols for information dissemination in wireless sensor networks. In *Prod. of ACM MobiCom'01*, 1999.

[18] X. Li, P. Wan, and O. Frieder. Coverage in wireless ad-hoc sensor networks. In *ICC 2002*, New York City, April 28 – May 2nd 2002.

[19] K. Lieska, E. Laitinen, and J. Lahteenmaki. Radio coverage optimization with genetic algorithms. In *IEEE International Sysmpsium on Personal, Indoor and Mobile Radio Communications*, volume 1, pages 318–22, Sep. 1998.

[20] A. Mainwaring, J. Polastre, R. Szewczyk, and D. Culler. Wireless sensor networks for habitat monitoring. In *First ACM International Workshop on Wireless Workshop in Wireless Sensor Networks and Applications (WSNA 2002)*, August 2002.

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

[22] S. Meguerdichian, S. Slijepcevic, V. Karayan, and M. Potkonjak. Localized algorithms in wireless ad-hoc network: Location discovery and sensor exposure. In *Proceedings of the ACM Symposium on Mobile Ad Hoc Networking and Computing (MobiHoc) 2001*, Long Beach, California, USA, October 2001.

[23] A. Molina, G. E. Athanasiadou, and A.R. Nix. The automatic location of base-station for optimized cellular coverage: a new combinatiorial approach. In *IEEE 49th Vehicular Technology Conference*, volume 1, pages 606–10, May 1999.

[24] G. Pottie and W. Kaiser. Wireless integrated network sensors (wins): Principles and practice. In *CACM'00*, 2000.

[25] A. Savvides, C. Han, and M. Strivastava. Dynamic fine-grained localization in ad-hoc networks of sensors. In *Proc. of ACM MOBICOM'01*, pages 166–179. ACM Press, 2001.

[26] E. Shih, S. Cho, N. Ickes, R. Min, A. Sinha, A. Wang, and A. Chandrakasan. Physical layer driven protocol and algorithm design for energy-efficient wireless sensor networks. In *Proc. of ACM MobiCom'01*, Rome, Italy, July 2001.

[27] S. Slijepcevic and M. Potkonjak. Power efficient organization of wireless sensor networks. In *ICC 2001*, Helsinki, Finland, June 2001.

[28] D. Tian and N. D. Georganas. A coverage-preserving node scheduling scheme for large wireless sensor networks. In *First ACM International Workshop on Wireless Sensor Networks and Applications*, Georgia, GA, 2002.

[29] Y. Xu, J. Heidemann, and D. Estrin. Geography-informed energy conservation for ad hoc routing. In *Proc. of ACM MOBICOM'01*, Rome, Italy, July 2001.

[30] F. Ye, H. Luo, J. Cheng, S. Lu, and L. Zhang. A two-tier data dissemination model for large-scale wireless sensor networks. In *proc. of ACM MobiCom'02*, 2002.

[31] F. Ye, G. Zhong, S. Lu, and L. Zhang. Energy efficient robust sensing coverage in large sensor networks. Technical report, UCLA, 2002.

[32] F. Ye, G. Zhong, S. Lu, and L. Zhang. Peas: A robust energy conserving protocol for long-lived sensor networks. In *The 23nd International Conference on Distributed Computing Systems (ICDCS)*, 2003.

APPENDIX

Given the locations of two sender nodes, (x_1, y_1) and (x_2, y_2) $(x_1 \neq x_2)$ and the location of the receiver node (x_0, y_0), the crossing point, (x, y), that is closer to the location of the node (x_0, y_0) can be calculated as follows. Refer Fig. 6, the cross points O and O' both satisfy the equations

$$(x - x_1)^2 + (y - y_1)^2 = r_s^2, \tag{9}$$
$$(x - x_2)^2 + (y - y_2)^2 = r_s^2. \tag{10}$$

Subtracting (10) from (9), we get the linear equation of OO':

$$x = -uy - v, \tag{11}$$

where

$$\begin{aligned} u &= (y_1 - y_2)/(x_1 - x_2), \\ v &= \frac{(x_2^2 + y_2^2) - (x_1^2 + y_1^2)}{2(x_1 - x_2)}, \end{aligned}$$

Substituting x using (11) in (9), we get a quadratical equation about y:

$$ay^2 + by + c = 0, \tag{12}$$

where

$$\begin{aligned} a &= 1 + u^2, \\ b &= 2u(x_2 + v) - 2y_2, \\ c &= y_2^2 + (v + x_2)^2 - r_s^2, \end{aligned}$$

Now we can solve the equation (12), and take the larger solution if R is above the line segment of $\overline{AB}$, vise versa. We get the crossing point (x,y) (the one closer to R):

$$\begin{aligned} \Delta &= \sqrt{b^2 - 4ac}, \\ y &= \begin{cases} \frac{-b+\Delta}{2a}, & \text{if } y_0 > y_1 + \frac{(y_2-y_1)*(x_0-x_1)}{(x_2-x_1)}, \\ \frac{-b-\Delta}{2a}, & \text{otherwise,} \end{cases} \\ x &= -uy - v. \end{aligned} \tag{13}$$

In the case of $x_1 = x_2$, The equation of OO' is simply

$$y = (y1 + y2)/2. \tag{14}$$

Substituting y in Eq. (9) using (14), we will get a quadratic equation about x. Solving the equation we can calculate the crossing point (x, y).