Design and Optimization of Distributed Sensing Coverage in Wireless Sensor Networks

TING YAN
University of Virginia
YU GU and TIAN HE
University of Minnesota
and
JOHN A. STANKOVIC
University of Virginia

For many sensor network applications, such as military surveillance, it is necessary to provide full sensing coverage to a security-sensitive area while, at the same time, minimizing energy consumption and extending system lifetime by leveraging the redundant deployment of sensor nodes. In this paper, we propose a surveillance service for sensor networks based on a distributed energy-efficient sensing coverage protocol. In the protocol, each node is able to dynamically decide a schedule for itself to guarantee a certain degree-of-coverage (DOC) with average energy consumption inversely proportional to the node density. Several optimizations and extensions are proposed to enhance the basic design with a better load-balance feature and a longer network lifetime. We consider and address the impact of the target size and the unbalanced initial energy capacity of individual nodes to the network lifetime. Several practical issues such as the localization error, irregular sensing range, and unreliable communication links are addressed as well. Simulation shows that our protocol extends system lift-time significantly with low energy consumption. It outperforms other state-of-the-art schemes by as much as 50% reduction in energy consumption and as much as 130% increase in the half-life of the network.

Categories and Subject Descriptors: C.2. [**Computer Communication Networks**]: Network Protocols

General Terms: Algorithms, Performance, Design

Additional Key Words and Phrases: Sensor networks, sensing coverage, energy conservation

33

1. INTRODUCTION

Wireless sensor networks have emerged as a new information-gathering paradigm based on the collaborative effort of a large number of sensing nodes. In such networks, nodes deployed in a remote environment must self-configure without any *a priori* information about the network topology or global view. Nodes act in response to environmental events and relay collected and possibly aggregated information through the dynamically formed multihop wireless network in accordance with desired system functionality. These networks can form the basis for many types of applications such as military surveillance [He et al. 2004b], habitat monitoring [Szewczyk et al. 2004], infrastructure protection [Xu et al. 2004], and scientific exploration [Tolle et al. 2005].

Low-cost deployment is one acclaimed advantage of sensor networks, which implies that the resources available to individual nodes are severely limited. Limited processor bandwidth and small memory are two arguable constraints in sensor networks, which will disappear with the development of fabrication techniques. However, the energy constraint is unlikely to be relieved quickly because of slow progress in developing battery capacity. Moreover, the unattended nature of sensor nodes and hazardous sensing environments preclude battery replacement as a feasible solution. On the other hand, the surveillance nature of sensor network applications requires a long lifetime; therefore, it is a very important research issue to provide a form of energy-efficient surveillance service for a geographic area.

This paper addresses the problem of providing a surveillance service for sensor networks that is adaptable to the system requirements. We first propose the main design that provides full coverage and then extend it into a differentiated version. Four major goals are achieved in this paper. First, our new sensing coverage algorithm outperforms other state-of-the-art solutions in terms of energy conservation, energy balance, and communication overhead when it runs in full coverage mode. Second, differentiated surveillance for sensor networks can be supported easily with minor changes to the basic design. Third, this paper proposes three additional optimizations, taking the size/shape of targets, the unbalanced initial energy capacity of individual nodes, and reference distribution into account to enhance the energy conservation and balance of the sensor network further. Fourth, we investigate the impact of the localization error, irregular sensing range, and unreliable communication links on our sensing performance.

The remainder of this paper is organized as follows: Section 2 discusses previous research related to the sensing coverage problem found in sensor networks. Section 3 describes the design of our sensing coverage protocol without differentiation. Section 4 extends the design to provide differentiated

surveillance. Section 5 extends the design to consider the size and shape of targets. Section 6 enhances the design with a technique called reference point scattering. Section 7 gives a brief description of baselines to which we compare our work, including two other state-of-the-art sensing coverage protocols. Section 8 provides a detailed performance evaluation and comparison. Section 9 investigates the implication of reality issues such as the localization error, irregular sensing range, and unreliable communication links. We conclude the paper in Section 10.

2. RELATED WORK

A set of research issues needs to be addressed before surveillance-based applications, such as military tracking and environmental monitoring, become technically feasible and economically practical. Recently, several schemes are proposed to address the sensing coverage problem in sensor networks. In Tian and Georganas [2003], full surveillance coverage is support by a node-scheduling scheme based on off-duty eligibility rules, which allows nodes to turn themselves off as long as the neighboring nodes can cover the area for them. This rule guarantees 100% sensing coverage as long as no void exists. However, this rule underestimates the area that the neighbor nodes can cover, which leads to excess energy consumption. The coordinated sleep algorithm proposed in Hsin and Liu [2004] improves the aforementioned algorithm by providing continuous coverage and compares the approach with a random sleep algorithm. In Ye et al. [2003], surveillance coverage is achieved by a probing mechanism. In this solution, after a sleeping node wakes up, it broadcasts a probing message within a certain range and waits for a reply. If no rely is received within a timeout, it takes the responsibility of surveillance until it depletes its energy. In this solution, the probing range and wakeup rate can be adjusted to affect the degree of coverage indirectly. However, this probing-based approach has no guarantee on sensing coverage and blind points can occur. Co-Grid [Xing et al. 2004] is a distributed protocol for sensing coverage maintenance based on coordinating fusion groups on virtual grids. It maintains probabilistic detection guarantees based on probabilistic detection models and data fusion among multiple nodes. Our solution is different from these solutions in the sense that it can not only guarantee sensing coverage to a certain geographic area, but it can also adapt the degree of coverage to that area, up to the limitation imposed by the number of sensor nodes present.

Energy balancing is another research issue addressed in this paper. Tian and Georganas [2003] and Ye et al. [2002] only consider the metric in terms of the total amount of energy consumed, regardless of the distribution of the energy among the nodes. We argue that unbalanced energy dissipation causes some nodes to die much faster than others do, therefore, the half-life of the network is dramatically reduced in the unbalanced approach. Research has addressed the energy balance issue from different aspects of sensor networks. SPEED [He et al. 2003b] balances the traffic by nondeterministically forwarding the packet through multiple routes. GAF [Xu et al. 2001] performs leader rotation among the nodes inside a virtual grid, in order to balance energy consumption.

Research on network topology control such as ASCENT [Cerpa and Estrin 2002], SPAN [Chen et al. 2001], LEACH [Heinzelman et al. 2000], and GAF [Xu et al. 2001] addresses the problem of providing communication coverage within an energy conservation context. These have a similar flavor as the surveillance coverage problem. For example, LEACH [Heinzelman et al. 2000] partitions a network into clusters and randomly rotates the cluster leader in order to evenly distribute the energy consumption among the sensors. SPAN [Chen et al. 2001] is another randomized algorithm where nodes make local decisions on whether to sleep or to join a backbone network in order to reduce energy consumption. The major difference between those and this work is that communication coverage considers only connectivity between the nodes. In contrast, surveillance (sensing) coverage addresses the coverage problem to every physical point in the terrain. As a result, new scheduling is required to force some nodes to stay awake for surveillance purposes, even though they are not participating in data forwarding.

It also helps network communication to provide a full sensing coverage. The impact of full sensing coverage of a sensor network on greedy forwarding algorithms is studied in Xing et al. [2006]. An upper bound of network dilation is derived so that the sum of lengths of transmission hops along the routing paths is bounded, which leads to high communication efficiency. Geometric analysis is conducted on the relationship between sensing coverage and communication connectivity in Xing et al. [2005]. A coverage configuration protocol (CCP) is proposed for energy-efficient sensing coverage and is integrated with SPAN for connectivity. Compared with CCP, which requires broadcasts of join/withdraw signals for each state rotation, the schedule of each node in our approach is generated at the beginning of the network operation and does not require more communication at each rotation. In addition, our protocol provides more flexible control over degree of coverage for the system.

Work/sleep schedules also attract research attention in the area of power-conserving tracking applications of wireless sensor networks. Several publications [Gui and Mohapatra 2004; Ren et al. 2005; Cao et al. 2005] address various aspects, such as theoretical analysis and algorithm designs, for tracking applications. These approaches are based on a partial sensing coverage and their main interests are on the impact of the incomplete sensing coverage on the tracking performance metrics.

Besides the aforementioned work in communication and sensing coverage that conserves energy, work in energy conservation for general sensor networks has been considered at various levels of the communication stack. From the bottom to the top, special hardware [Min et al. 2000] is designed with multiple energy dissipation settings. MAC layer protocols developed for energy savings mostly take advantage of overhearing and scheduling to allow nodes to sleep while they are not transmitting or receiving messages [Guo et al. 2001; Heinzelman et al. 2000]. At the network and routing layers, solutions are diversified. Data-placement schemes [Bhattacharya et al. 2003] minimize energy along the transmission path through data caching. In Ramanathan and Rosales-Hain [2000], Ramanathan et al. adjust communication range dynamically, based on the node density to conserve energy consumed in transmission.

MFR [Takagi and Kleinrock 1984] by Takagi et al. uses a minimal hop path to reduce the total number of transmissions. Xue and Li [2001] sets routes according to the energy remaining at nodes along that path, and [He et al. 2003b] uses mechanisms to save energy through the distribution of messages among various paths from source to destination. At the application layer, the protocols incorporate routing semantics to form groups and rotate leadership responsibilities, allowing nonleader nodes to sleep and conserve their energy [Chen et al. 2001]. Finally, data aggregation techniques inside the network and application layers also provide energy conservation features in both application-independent [He et al. 2004a] and dependent fashions [Intanagonwiwat et al. 2002; Krishnamachari et al. 2002]. All of these protocols tackle the energy issue from different aspects of sensor networks. Thus, we consider them complementary to our work.

Research has also been conducted to address QoS issues to support differentiated services for sensor networks. SWAN [Ahn et al. 2002] uses feedback information from the MAC layer to regulate the transmission rate of non-real-time traffic in order to sustain real-time traffic. RAP [Lu et al. 2002] uses velocity monotonic scheduling to prioritize real-time traffic and enforces such prioritization through a differentiated MAC layer. In Bhatnagar et al. [2001], the delivery ratios of packets at different priority levels are affected by the forwarding probability at intermediate nodes. To date, to the best of our knowledge, no algorithm has been specifically designed to address how to provide differentiated surveillance for sensor networks.

3. BASIC PROTOCOL DESIGN

In this and Section 4, we introduce the design of an adaptive-sensing coverage scheme for sensor networks. The basic design without differentiation is first introduced in Section 3. This is then the basis for the extension for differentiated surveillance to support scenarios in which a partial coverage ($<100\%$) is sufficient or a high degree of coverage ($>100\%$) is desired (Section 4).

3.1 Design Goals

The basic design goal for this work is to provide energy-efficient sensing coverage for a geographic area covered by sensor nodes. Though it has been addressed in previous work [Tian and Georganas 2003; Ye et al. 2002], our work distinguishes itself from previous solutions in following subgoals that we achieve.

—Reduce total amount of energy consumed.
—Reduce energy variation among nodes.
—Reduce communication overhead in establishing nodes' work schedules.
—Support a new extension to the service for differentiated surveillance with small overhead.
—Provide communication connectivity as an auxiliary benefit.

In the remaining parts of the paper, we support these claims with analytical analysis and simulation results.

3.2 Assumptions

First of all, we assume the deployed sensor nodes can fully cover the sensing field if they are all awake. Second, we assume that each node knows its own location [He et al. 2003a] and nodes are not moving. The node location information does not need to be precise when we are using conservative sensing ranges to decide the schedules (see Section 3.4). These are common assumptions for many sensor network applications. For simplicity and convenience of protocol description in the rest of the paper, we refer to the sensing area of a node as a circle with a nominal radius r centered at the location of the node itself. We later show that we can also deal with irregular and/or nonuniform sensing areas as long as the neighboring nodes are aware of each other's sensing area. In the protocol description, we deploy the sensor nodes in a two-dimensional (2D) Euclidean plane. However, the protocol can be extended to a three-dimensional (3D) space or a curved surface without much difficulty. We also assume that the neighboring nodes should be roughly time-synchronized on the order of seconds, which can be easily achieved by current millisecond-level synchronization solutions, such as found in Elson et al. [2002]. The tolerance for small synchronization skews (see Section 3.4.2) also allows the resynchronization process to happen even less frequently, so that the cost can be neglected.

The last assumption we make is that nodes can directly communicate with the neighboring nodes within a radius larger than $2r$ (r is nominal-sensing radius). This is a typical case in all systems we experienced. For example, MICA II [CrossBow Technology, Inc.] has a open-space communication range of about 500 ft, while the sensing range is within 100 ft even with long-range motion sensors. Actually, this is an optional assumption. The protocol works so long as the nodes are able to communicate directly or indirectly with each other within the distance of $2r$ and the communication range does not need to be regular. We make this assumption for simplicity in the protocol description and to avoid routing overhead for any two nodes that can sense a common area. This assumption also helps provide a connectivity guarantee when full sensing coverage is achieved.

3.3 Basic Design without Differentiation

Each node in the sensor network is either in sleeping or in working mode. Our goal for the basic design without differentiation is to have as many nodes as possible go to sleep to save energy and extend the lifetime of the sensor network while guaranteeing 100% sensing coverage to the target area.

3.3.1 *A Node's Working Schedule.* The lifetime of a sensor network is composed of an initialization phase and a sensing phase. During the initialization phase, each sensor node finds its own position [He et al. 2003a] and synchronizes [Elson et al. 2002] time with neighboring nodes. After that, nodes enter into a sensing phase and start to sense environmental events. The sensing phase of nodes is divided into rounds with equal duration and nodes are synchronized with each other. Each node establishes a working schedule through our algorithm, which tells it when to sleep and when to work for each round. When a node goes to sleep, its sensing, communication, and computation components
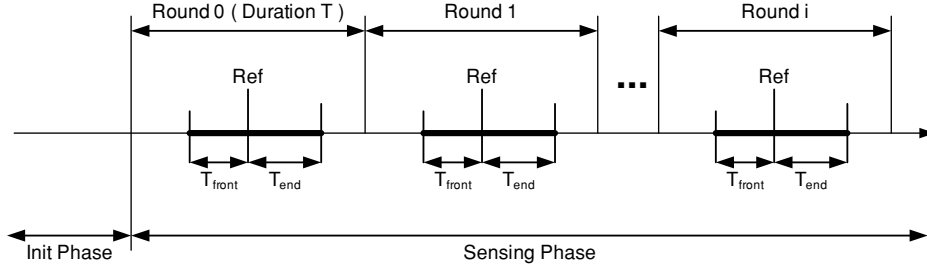
Fig. 1. Node working schedule.

can all be asleep and only a timer needs to work and wake up all components according to its schedule.

We denote the duration of each round as T. The schedule for a node is determined by a tuple with four parameters: (T, Ref, T_{front}, T_{end}). As shown in Figure 1, Ref is a random time reference point chosen by a node within [0, T). T_{front} is the duration of time prior to the reference point Ref, and T_{end} is the duration of time after reference point Ref. We describe how to decide these parameters in Section 3.3.2. For any given tuple (T, Ref, T_{front}, T_{end}), a node executes the following working schedule:

A node wakes up at time $(T \times i + Ref - T_{front})$ *and goes to sleep at time* $(T \times i + Ref + T_{end})$, *where i is the round sequence number*.

It should be noted that the following three rules hold for the schedule:

1. The tuple (T, Ref, T_{front}, T_{end}) for each node is chosen during the initialization phase and does not change during the whole sensing phase, unless rescheduling is required for fault-tolerance purposes.
2. Time duration T_{front} and T_{end} should be in the interval [0, T) and, furthermore, the sum of T_{front} and T_{end}, which is the time duration for a node to work during each round, should be less than or equal to T.
3. The wake-up schedule of a node can cross the boundary of two consecutive rounds T.

3.3.2 *Setting Tuple Parameters* (T, Ref, T$_{front}$, T$_{end}$). In our approach, we cover the target area with a virtual square grid. We describe how to choose the unit grid size later in Section 3.4.1. Now we describe the protocol that decides the schedule for each sensor node to guarantee that each grid point in the target area is covered by at least one working node at any time and with minimum energy consumption. Here we only consider the energy consumed by the sensing function, so the energy consumed by a node is proportional to its working time for each round, which is ($T_{front} + T_{end}$). Parameter T, which is the duration of each round, is predetermined and kept constant across all the nodes inside the sensor network. We discuss how to set this parameter in more detail in Section 3.4.2.

After the nodes complete localization and time synchronization, they each broadcast a beacon (a random jitter in transmission time is added to avoid collisions). This beacon has the information of its location and a randomly
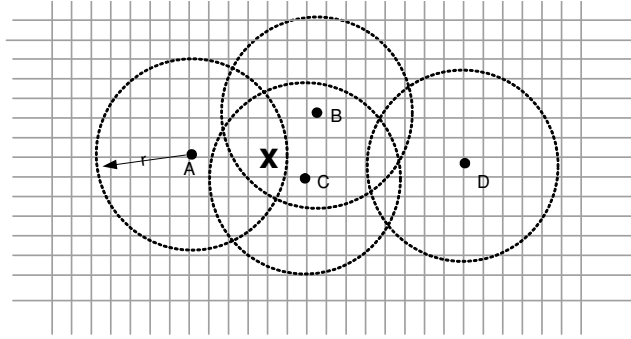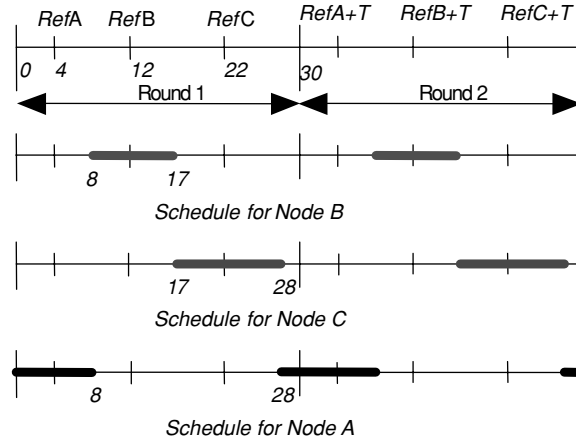
Fig. 2.   Sensing coverage to single grid point X.

Fig. 3.   Calculating node schedules for a single grid point.

chosen time reference point *Ref*. The reference time should be uniformly chosen among $[0, T)$ and follows an identical independent distribution (iid). According to the assumption of direct communication within $2r$, each node within the distance of $2r$ of the sending node should receive the beacon. For example in Figure 2, nodes A, B and C hear each other. After every node beacons the time reference point *Ref*, each node maintains a neighbor table with the information of locations and time reference points of its neighbors within $2r$'s distance. For example, node A keeps information about nodes B, C, and itself in a table. D's beacon is ignored, since it is outside the $2r$'s distance of A.

Before discussing the sensing coverage of the entire area, we first focus on how to provide sensing coverage for a single grid point. Here we consider node A and grid point x shown in Figure 2. Obviously, all the sensor nodes whose sensing areas cover grid point x are in the $2r$'s distance of node A. Therefore, node A has the information of their locations and time reference points in its neighbor table. With such information, node A sorts those time reference points in ascending order and lays out all N reference times, including its own reference time point (see Figure 3). (Suppose that the total number of sensor nodes, which can cover grid point x is N).

Without loss of generality, suppose node A has the ith reference time $Ref(i)$, node B has the (i-1)th reference time $Ref(i - 1)$, and node C has the (i + 1)th reference time $Ref(i + 1)$. Then the time interval T_{front} and the time interval T_{end} for grid point x are set by our algorithm to $[Ref(i) - Ref(i - 1)]/2$ to $[Ref(i + 1) - Ref(i)]/2$, respectively. There are two special cases for the T_{front} and T_{end} calculations:

1. If node A has the smallest reference time among all N nodes that can cover grid point x, then the time interval T_{front} is $(T + Ref(1) - Ref(N))/2$ and the time interval T_{end} is $(Ref(2) - Ref(1))/2$.
2. If node A has the largest reference time among all N nodes that can cover grid point x, then the time interval T_{front} is $(Ref(N) - Ref(N - 1))/2$ and the time interval T_{end} is $(T + Ref(1) - Ref(N))/2$.

To understand the procedure better, we explain how to decide node schedules for a certain grid point x through an example shown in Figure 3. Suppose the duration of each round T is 30 min and only nodes A, B, and C can provide coverage to grid point x. Nodes A, B, and C choose reference time Ref values 4, 12, and 22, respectively. According to our algorithm, node B sets the T_{front} value to (12-4)/2 = 4 and set the T_{end} value to (22-12)/2 = 5 to cover the one-half gap between the reference time points. Consequently, the parameter tuple (T, Ref, T_{front}, T_{end}) for B is (30, 12, 4, 5), which means node B stays awake for (4 + 5) min out of the 30 min period (4 min before and 5 min after time reference point 12). Similarly, the parameter tuples for A and C are (30, 4, 6, 4) and (30, 22, 5, 6).

It should be noted that such a schedule for a single-grid point is optimal in terms of energy consumption, since the total awake time for all three nodes is equal to the duration of the round (30 min in the example). Any other schedule, which has less total awake time, introduces a blind time on that grid point. Since there is no overlap among the node's working schedules for grid point x and total coverage time equals the duration of the round, it is easy to conclude that at any time, grid point x is covered by at least one node's schedule.

The same procedure is carried out for every grid point. In the end, every node has a set of schedules for all the grid points it can cover. Since it is impossible for a node to simultaneously execute multiple schedules, it is necessary to create a single integrated schedule. Now, we describe how a node combines this set of schedules for grid points into a single schedule for the area it can cover.

After each node calculates its schedules for all grid points it can cover, it creates an integrated schedule (T, Ref, T_{front}, T_{end}). The integrated T_{front} of node A is the largest one among all T_{front} time intervals of node A for all the grid points that node A can cover. Similarly, the integrated time interval T_{end} of node A is the largest one among the T_{end} time intervals of node A for all grid points that node A can cover. Therefore, the integrated schedule of each node is the union of its schedules for all the grid points it can cover. Previously, we have concluded that a given grid point is covered fully in time by at least one neighboring node's schedule. Thus, with the calculated integrated schedules of all the sensor nodes, at any time any grid point is covered by at least one
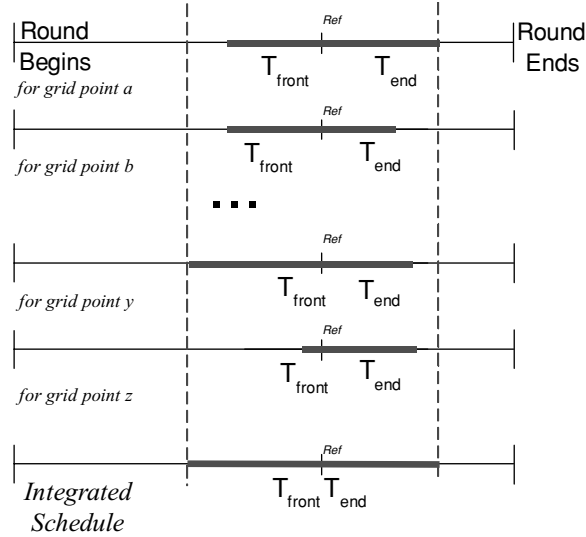
Fig. 4.   The process of schedule integration on a node.

awaken sensor node. To clarify this integration process further, Figure 4 shows how to calculate the integrated schedule on a node according to its schedules for a set of grid points.

Note that many sensor networks are deployed with a reasonable density so that when every node is working, the whole target area is sensed by the sensing nodes without blind holes and our approach guarantees full sensing coverage at any time during the sensing phase. Another noteworthy point is that, in order to prevent the nodes at the edge of the network from working all the time, when we specify the target area, we need to exclude the edge of the network. Then, with enough density, it is highly probable that any grid point in the specified target area can be sensed by at least two nodes when all nodes are awake.

One might argue that intuitively such integration through the union of schedules for each grid point would lead to a pessimistic result with very long working duration. However, this is not the case. In fact, a node usually has the same schedule for grid points near to each other, which does not increase the length of integrated T_{front} and T_{end} when the union operation takes place. This has been shown in the simulation.

Our approach has several major advantages for sensor network deployment:

1. Communication overhead in our approach is minimized. For the approach described above, each node needs only to beacon one message without consideration of collisions. Compared to approaches that a node beacons each time it wakes up [Ye et al. 2002] or for each round [Tian and Georganas 2003], the energy consumed by our approach's communication is much less.
2. The random time reference point with uniform distribution contributes to the energy balance among the nodes. An optimization on this (discussed later) can enhance this energy-balancing feature even more.

3. Because of the optimal energy consumption property for a single grid point, the total energy consumption of our approach increases very slowly with the increase of node density. As a result, the system lifetime increases proportionally with the increase in node density. We show this in the evaluation section.

Although the length of working time for each node may differ as a result of randomness, the protocol is still fair in the sense that no node is given priority to work longer or shorter.

It can be easily seen from the above description that even when at the initialization phase some node is not discovered by its neighbors, once there are enough nodes to cover the target area, the 100% sensing coverage is still guaranteed. However, it may cause other nodes to work unnecessarily longer.

The protocol just described provides a guarantee of 100% sensing coverage at any time for the target area. Acknowledging that in some situations, full coverage by single sensor nodes is not enough to obtain high confidence in detection, our basic design can be easily extended to provide a guaranteed degree of coverage for a geographic area with very small overhead (see Section 4).

3.4 Design Issues

This section completes our approach with several design issues concerning grid granularity, clock skew, irregular sensing patterns, and node failures.

3.4.1 *Dealing with Possible Blind Points in Space Because of the Large Granularity of the Grid Size.* One may argue that in our protocol description, there can be small sensing holes, because we only guarantee grid point coverage, instead of a grid area. Here we explain why this approach still makes sense. We denote the unit grid size as d. We also define a conservative sensing area of a sensor node as a smaller concentric circle than its actual sensing area in order to handle this problem and other issues.

From Figure 5, it can be easily seen that the target area is fully covered by the sensor nodes' actual sensing areas as long as (1) each grid point is inside the conservative sensing area of at least one sensor node, and (2) the difference between the actual sensing radius r_{actual} and the conservative sensing radius $r_{conservative}$ is greater than $d/\sqrt{2}$. The problem of covering the whole target area with actual sensing areas then reduces to that of covering the grid points with conservative sensing areas. With such observations, we decide the upper bound of the grid size according to the following rules.

1. If we use the actual sensing radius to calculate the nodes' working schedules, the grid size d should be smaller than the size of the targets we want to detect, so that these targets do not fit in the gap among the grid points.

2. If we use a conservative sensing radius, the grid size d should be smaller than $\sqrt{2}(r_{actual} - r_{conservative})$.

3. At the same time, a lower bound for the grid size should be set with consideration of computational cost (see Section 3.7.1).
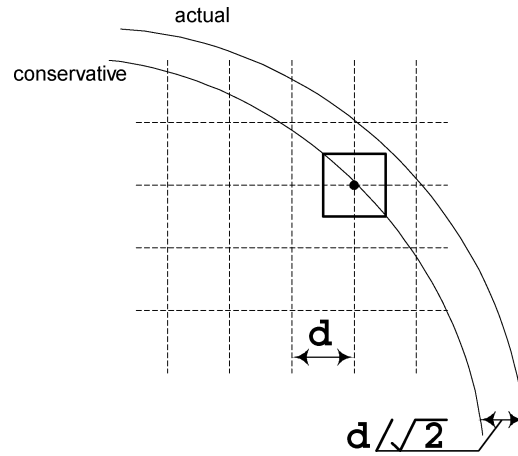
Fig. 5. Grid size granularity issues.

4. Other factors like the imprecision of location information and irregularity of the sensing areas should also be considered when we decide the conservative sensing radius.

In the remaining protocol description, we do not distinguish between concepts of grid point and area coverage.

3.4.2 *Dealing with Possible Blind Points in Time Because of Synchronization Skew.* Because of the precision limits of synchronization algorithms for real sensor network systems, there is always some synchronization skew. When nodes try to trigger some events at the same time, the actual time the event is triggered on each node is different as a result of the synchronization skew. Our protocol is based on synchronization, so we must compensate for the synchronization skew to make the schedules seamless. We define the maximum skew of the whole sensor network as T_s. In order to bridge the gaps caused by the synchronization skew, we only need to extend the schedule of each node for $T_s/2$ in both directions. This compensation affects the performance of the protocol. In order to minimize the impact of this performance degradation, it is favorable to increase the time duration T for each round. The reason is that when T_s is fixed, the longer a round is, the lower is the compensation overhead compared to the duration of a round or the schedules for a round. For example, suppose $T_{front} + T_{end}$ is 30 min, if we allow synchronization skew to be 1 s, the performance is degraded less than 0.1%. However, T should not be very large, which may lead to unbalanced energy consumption.

3.4.3 *Dealing with Possible Blind Points in Space Because of Irregularity in Sensing Patterns.* In the description of the protocol, we assume that the sensing area of each node is uniformly a regular circle and the nodes are deployed in a 2D Euclidean plane. However, if we review the protocol description, it is clear that these assumptions are not mandatory. We define a node's sensing neighbors as the nodes that can sense at least a subarea of its sensing area. Our protocol works as long as the sensing neighbors are aware of each other's sensing area,
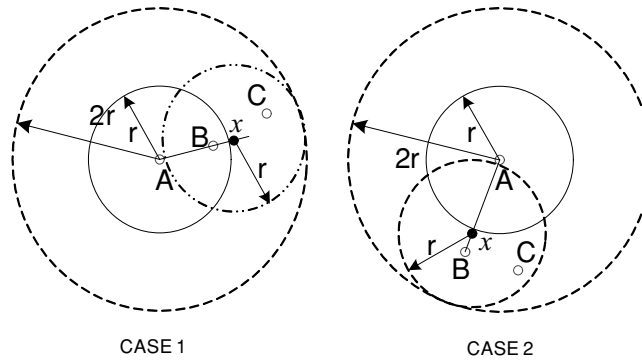
CASE 1   CASE 2

Fig. 6.   A wakeup example for rescheduling.

through preexisting knowledge or communication with small overhead, no matter where the nodes are deployed, and whether the sensing areas of nodes are regular or uniform. Even without precise information about sensing areas or node locations, we can use a conservative sensing range to make sure of sensing coverage.

3.4.4 *Reschedule for Fault Tolerance.* In this subsection, we propose a mechanism to provide fault tolerance with the basic protocol that guarantees 100% coverage. We assume any two nodes do not fail at the same time and when some node fails, it just stops working silently. After each node calculates its integrated schedule, it broadcast this schedule to its $2r$ neighbors. When a node is working, it periodically sends a heartbeat signal to its working neighbors within the $2r$ range. Each working node knows the schedules of its $2r$ neighbors and expects heartbeat signals from the working ones among such neighbors. After a node detects the failure of one of its neighbors through heartbeat timeout, it wakes up all its neighbors that are also the $2r$ neighbors of the failed node. The wakeup mechanism can be achieved through either a hardware wakeup circuit [Gu and Stankovic 2004] or a software solution [He et al. 2006]. These $2r$ neighbors then recalculate the schedules with the knowledge that the problematic node failed. With the new schedules, the target area is guaranteed to be fully covered by working nodes' sensing areas. In exchange for the fault tolerance feature, the communication overhead increases significantly because of the heartbeat signals.

Figure 6 gives a simple proof to show that all sleeping $2r$ neighbor of a failed node A can be awakened up by at least one working node that is also node A's $2r$ neighbor. There are two cases: (1) suppose a sleeping node B is inside the sensing area of node A. We extend the line segment that connects A and B. A point x on the extended line segment is very close, but outside of node A's sensing area. There must be some working node C that is sensing point x. It can be easily seen that node C is in the $2r$ neighborhood of node A and the distance of C and B is less than $2r$ so that node C can wake up node B. (2) Suppose a sleeping node B is outside of node A's sensing area, we connect A and B with a line segment. We choose point x so that x is on the line segment and is very

close, but outside of A's sensing coverage. There must be some working node C that covers point x. It can be easily seen that C is in the $2r$ neighborhood of A and the distance of C and B is less than $2r$ so that C can wake up B.

3.5 Optimizations and Extensions

This section discusses optimizations and extensions built upon our basic design. These techniques achieve better performance in exchange for more beacons required.

3.5.1 *Second-Pass Optimization for Lower Energy Consumption.* Each node's integrated schedule is the super set of its schedules for many grid points. It is possible that the integrated schedules are more than sufficient to provide the coverage guarantee. Therefore, we make a second-pass optimization to reduce this kind of redundancy. In order to simplify the description, we only describe the optimization for a 100% degree of coverage. After a sensor node calculates the integrated schedule, it sends this integrated schedule to its neighbors within the distance of $2r$. When some node finds out that it is the one with the longest schedule, which has the largest $T_{front} + T_{end}$ value compared to its $2r$ neighbors, it recalculates a new schedule. The new schedule shrinks T_{front} and T_{end} values as much as possible, while still guaranteeing 100% coverage for all the grid points the node is able to sense. Then it beacons the new schedule to its neighbors within $2r$'s distance. Each node that has the longest schedule among the nodes in the $2r$ range that have not updated their schedules recalculates the schedule and beacons its new schedule. When there is a conflict of beacons, the conflicting nodes make use of a random back-off scheme to send the beacons again and avoid conflicts. In the evaluation section, we show that the system performance improves with this optimization.

3.5.2 *Multiround Extension for Energy Balance.* A sensor network's lifetime is affected by both the total energy consumption of all the sensor nodes and the variation of energy consumption per unit time by each node. In a sensor network that has higher variance in energy consumption, some nodes run out of energy much earlier than others do. Then blind holes appear in the target area, or another scheduling process should be applied to eliminate the blind holes. Given the same total energy consumed, the less is the variation of energy consumption per unit time, the longer the network lasts with full sensing coverage.

Energy consumption variance in our protocol can be attributed to at least two reasons. First, because of the randomness of node deployment, some nodes may have fewer neighbors than others in the range of $2r$, so they have to work longer than others do. This imbalance is intrinsic for a certain deployment and little can be done to deal with this problem. The second reason results from the randomness of reference time. As an example, if several sensor nodes can all sense a grid point, and their reference times are selected very close to each other, there must be an extraordinarily long schedule of one of the nodes for this grid point compared to others. Here, we propose an extension to smooth the energy consumption variation caused by the randomness of reference time selection.

Instead of calculating a single schedule, we calculate **M** schedules for each node according to **M** independently selected random reference times *Ref* for each node. At the initialization phase, each node beacons its **M** reference times *Ref* to its $2r$ neighbors. The ith (i is between 1 and **M**) schedule is calculated with the ith reference time of each node. Then, at each round in the sensing phase, the nodes consecutively choose one schedule. We show in the evaluation section that this extension decreases the variation in energy consumption.

3.5.3 *Enhancement For Different Initial Node Energy*.   In the previous sections, we assume that all nodes have the same initial energy, however, this is not always the case in the real applications. Therefore, we propose a solution to deal with this issue.

First, intuitively, we need to modify the equations for deciding T_{front} and T_{end}. In the previous design, the time interval T_{front} and the time interval T_{end} for grid point x are set to $[Ref(i) - Ref(i - 1)]/2$ and $[Ref(i + 1) - Ref(i)]/2$, respectively, based on the assumption that all nodes have same initial energy and each node should work one-half time between two reference points. In the case of different initial node energy, we should consider this factor when deciding T_{front} and T_{end}. We modify the existing formulas into $[Ref(i) - Ref(i - 1)] \times$ [(initial energy for i)/(initial energy for i + initial Energy for i − 1)] and $[Ref(i + 1) - Ref(i)] \times$ [(Initial Energy for i)/(Initial Energy for i + Initial Energy for i + 1)], respectively. By changing the formulas, for each grid point, the amount of time that each node works is according to the portion of its energy to the total energy of two neighboring nodes and, therefore, balances the energy consumptions among nodes for a grid point.

However, we found that modifying the equations for T_{front} and T_{end} alone cannot ensure the extension of network lifetime, because the integrated T_{front} and T_{end} for a node are the largest ones among all T_{front} and T_{end}, respectively. If, on a specific grid point, two low-energy capacity nodes are neighbors, their T_{front} or T_{end} benefits little from the revised formulas discussed above, according to the algorithm for calculating integrated T_{front} and T_{end}, the optimized T_{front} and T_{end} then may be shaded by those unoptimized T_{front} and T_{end}, resulting in small balancing gain in node energy consumption. Therefore, we propose, instead of selecting reference points for nodes completely randomly, we separate the domain for reference points by the node density of network and generate random reference points for high-energy capacity nodes and low-energy capacity nodes in certain intervals so as to increase the chance that the reference points of nodes with a significant energy difference are neighboring to each other.

For example, when the node density is 4 per r^2, we assign the reference points for high-energy capacity nodes to be the random numbers in the first and third quarter of a round, while setting low-capacity ones to the second and fourth. Thus, when we decide T_{front} and T_{end} for a grid point, the chance that high-capacity ones are next to low-capacity ones is much higher than what we can obtain by completely selecting reference points randomly. In the evaluation section, we show this extension improves the lifetime of the network by 18% in the case that the nodes in the network have different initial energy.
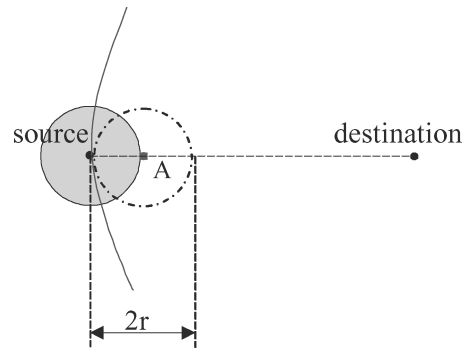
Fig. 7. Communication connectivity.

3.6 Effect on Communication Connectivity

Sensing coverage and communication connectivity are two major targets for energy-efficient approaches for sensor networks. For quite a few routing protocols like He et al. [2003b], the communication connectivity is based on the existence of a next hop closer to the destination. Say we want to send a message from a source to a destination and the source cannot reach the destination in one hop (Figure 7) in a sensor network whose 100% sensing coverage is guaranteed with our protocol. Let us consider the location A that is very close to, but out of the sensing circle of the source node and on the line segment connecting the source and the destination. In order to cover this location, there must be another node in the dashed circle with the radius r. We can see that any node inside the dashed circle is closer to the destination than the source node so that it can serve as the next hop. Thus, the existence of a next hop closer to the destination is guaranteed when a 100% sensing coverage is provided and the communication range is at least twice as much as the sensing range, which is a good feature that supports communication connectivity. For more information, the reader can refer to the CCP [Xing et al. 2005] protocol, which provides a very thorough analysis on the relationship between sensing coverage and communication connectivity.

3.7 Protocol Analysis

This section provides the theoretical analysis of our algorithm for a better understanding of the performance issues.

3.7.1 *Complexity Analysis.* The major complexity involved in the proposed protocol is the computational complexity. In the following discussion, we analyze the steps the protocol takes to calculate the integrated schedule for a single node.

The first step is to calculate the schedules for each grid point. We know that the number of grid points in the node's sensing area is about $\pi r^2/d^2$ and suppose that a single grid point can be sensed by N nearby nodes, on average, which depends on node density. For each grid point, a node should decide if it is in the sensing area of each node in the $2r$ range, which involves $4N\pi r^2/d^2$ distance

computations. For each grid point, all these nodes' reference times are sorted, which takes cNlog N basic steps, where C is a small constant. The schedule for a grid point is decided in negligible time. Last, it takes $2\pi r^2$ comparisons to get the integrated schedule for this node. For instance, in a certain sensor network, we choose d as r/10 and N as 10, the basic steps it takes to calculate the integrated schedule is in the order of magnitude of ten thousand. In a typical hardware platform [CrossBow Technology, Inc.], the energy consumed for tens of thousands of basic calculations is in the same order of magnitude of transmitting a single bit, so it is negligible compared to communication overhead.

The memory space occupied for calculating the integrated schedule is composed of (1) 4N entries on average for the neighborhood table, (2) N memory units for sorting the reference times on average, and (3) $2\pi r^2$ memory units for storing the schedules for grid points. Typically, the memory usage of the protocol on each node is in the order of magnitude of Kilobytes, which is suitable for resource bounded sensor nodes like CrossBow Technology, Inc.

3.7.2 *Communication Overhead Analysis.* In the basic design, each node only beacons once (without consideration of collisions) in the initialization phase, disseminating the information about its reference time and location. During the sensing phase, there is no need to send extra messages for coordination. In the second-round optimization, each node beacons three messages indicating the reference time, the integrated and the updated schedule. Typically, as mentioned, the energy consumption for sending or receiving a single bit is on the same order of magnitude as running tens of thousands of instructions. In real sensor networks, the retransmissions used to deal with contention make the energy consumption for sending a message even higher. Therefore, the very few number of messages transmitted in our protocol is preferable for minimizing energy consumption and extending system lifetime. Any extra energy consumed by complicated computation is negligible compared to the energy saved by decreasing communication overhead.

4. ENHANCED DESIGN WITH DIFFERENTIATION

Previous research focuses on how to provide full or partial sensing coverage in the context of energy conservation. In such an approach, nodes are put into a dormant state as long as their neighbors can provide sensing coverage for them. These solutions regard the sensing coverage to a certain geographic area as binary, either it provides coverage or not. However, we argue that in most scenarios, such as battlefields, there are certain geographic sections, such as the general command center that are much more security sensitive than others. Based on the fact that individual sensor nodes are not reliable and subject to failure and single sensing readings can be easily distorted by background noise and cause false alarms, it is simply not sufficient to rely on a single sensor to safeguard a critical area. In this case, it is desired to provide a higher degree of coverage in which multiple sensors monitor the same location at the same time in order to obtain high confidence in detection. On the other hand, it is overkill and energy consuming to support the same high degree of coverage for some noncritical area. As a result, a differentiated version of basic design is necessary.

It is easy to modify our basic protocol to deal with differentiated sensing coverage when there are different requirements, statically or dynamically. In the basic algorithm, the middle point between two reference times is selected as the start/end point of a schedule for a grid point. That is to say, T_{front} and T_{end} are chosen as one-half of the intervals between two reference times, respectively. If we want to adjust the degree of sensing coverage to an arbitrary degree α, the only change in the protocol required is to extend or shrink T_{front} and T_{end} proportionally, that is, T_{front} and T_{end} should be multiplied by α. It can be easily seen that with all the schedules for each grid point multiplied by α, the aggregate schedule of a sensor node is also multiplied by α. There is one constraint for the extension: if the sum of the intervals multiplied by α is greater than the round duration T, we should shrink the multiplier so that $T_{front} + T_{end}$ does not exceed the round interval.

More formally, a node's working schedule for differentiated coverage is determined by the tuple $(T, Ref, T_{front}, T_{end}, \alpha)$ and a node executes the following working schedule:

A node wakes up at time $(T \times i + Ref - T_{front} \times \alpha)$ and it goes to sleep at time $(T \times i + Ref + T_{end} \times \alpha)$, where i is the round sequence number.

The intended degree of coverage α provides a guarantee that the average number of working nodes that sense each grid point in each round is at least α, up to the limitation imposed by the number of sensor nodes present. Specifically, when α is 2, at any time, each grid point is sensed by at least two working nodes. When any single node fails, the target area is still fully covered. For an α value other than 1 and 2, the number of working nodes that sense a specific grid point is only guaranteed in the average sense. This feature is of great importance for a highly secured area. In fact, this differentiation mechanism can be generalized to the situations where α is smaller than 1. By doing so, our solution provides partial coverage in time to the target area and further reduces the total energy consumed.

It should be noted that this differentiation mechanism can be applied with very small overhead. After the sensor network is deployed, the sensor nodes are programmed so that any grid point of the target area is covered. When an emergency event happens, sensor nodes in the whole sensor network or part of the network should immediately raise the degree of coverage. For example, when a sensor network used for surveillance detects an intruder, more nodes should be awakened to achieve a more precise and reliable tracking of the intruder. The obvious advantage of our scheme is that no rescheduling through beaconing is required. One can disseminate the degree of coverage α to the target area and nodes simply expand their schedules according to this parameter to provide a higher degree of coverage without further beacon exchange.

5. ENHANCED DESIGN CONSIDERING THE TARGET SIZE AND SHAPE

In the basic protocol, we assume targets are 1D points for the sake of simplicity. However, this assumption, unfortunately, imposes an unnecessary requirement that the area should be fully covered. Since, in reality, targets are not points, if we consider the target size and shape, we can guarantee the detection with a

(a) Polygon parti-
tioned into 4 chains

(b) Two corners of
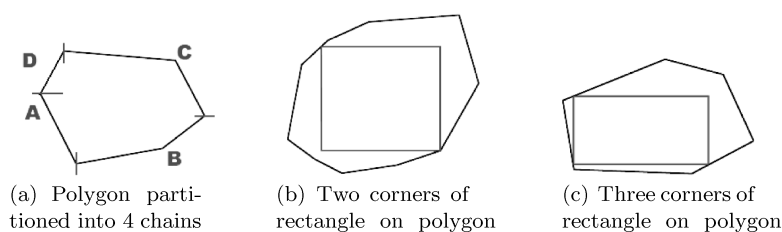rectangle on polygon

(c) Three corners of
rectangle on polygon

Fig. 8.   Process for finding the largest inscribed rectangle in a polygon.

partial coverage, which leads to a better energy conservation. In the enhance-
ment described in this section, we assume that the types of targets are known
a prior, i.e., the minimal target size is one of the inputs into our coverage al-
gorithm. We also assume that the shape of sensed targets can be described by
arbitrary polygons. The requirement for the partial coverage enhancement is
to ensure that (static or dynamic) targets do not fall into any sensing hole in
the networks.

   In the basic protocol, nodes are scheduled to cover the grid points. If the
gaps between the grid points are very large, a target denoted by a polygon can
exist within the gap and, hence, can not be detected. If the gaps between grid
points are very small, we consume excessive energy. Ideally, we need to find the
largest inscribed rectangle of a polygon and use this rectangle to build the grid.
Finding an inscribed rectangle is a typical problem in the field of computational
geometry, where several solutions using the prune-and-search technique have
been proposed [Alt et al. 1995; Kirkpatrick and Snoeyink 1995; Daniels et al.
1997]. To illustrate the idea without getting into too much detail, we give an
example here. As shown in Figure 8a, first, we identify the minimum X_{min},
Y_{min} and maximum X_{max}, Y_{max} values among the vertexes of the polygon. The
vertexes with these values are called extreme vertexes (e.g., vertexes with a
crossing bar in Figure 8a). Second, by breaking at extreme vertexes, we decom-
pose the boundary of a polygon into four pieces (e.g., A, B, C, and D as shown
in Figure 8a). Third, we consider two cases where an inscribed rectangle has
either two (Figure 8b) or three corners (Figure 8c) on the boundary of this poly-
gon. We can obtain the largest inscribed rectangle by searching the possible
corner positions within divided pieces. As for the two-corner case, one of the
corners must be located at a polygon vertex, as shown in Figure 8b. We omit
the searching method for the more complex three-corner case, which can be
found in Alt et al. [1995]. By finding the largest inscribed rectangle, we are
sure that as long as the four corners of inscribed rectangle could be covered
by the nodes, then the polygon could also be covered. As proven in Alt et al.
[1995], Kirkpatrick and Snoeyink [1995], and Daniels et al. [1997], there could
be multiple largest inscribed rectangles for a polygon. In that case, we choose
an inscribed rectangle that has the smallest difference between the longer and
shorter sides to maximize the sensing range and grid size discussed below.

   To make minimal change to the basic protocol, instead of changing the full-
coverage algorithm to a partial-coverage algorithm, we *virtually* extend the
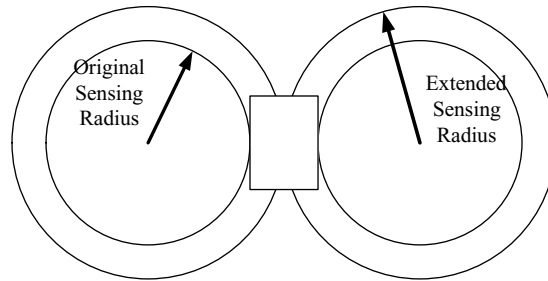physical sensing range by one-half of the shorter side of the rectangle in our

Fig. 9. The method used to extend the sensing range.

coverage algorithm, as shown in Figure 9. From this figure, we can see that even if the target is between the sensing circles of nodes, it still can be detected. We note that this enhancement is very simple. A change of the sensing range parameter according to the target size is all we need. This enhancement is also very flexible. If the assumption about the target size changes, all we need to do is to recalculate the schedule with a different sensing range. This enhancement is orthogonal to other aspects of our protocol. For example, the differentiation scheme can be still applied together with this enhancement, offering more flexibility to the system design.

6. ENHANCED DESIGN WITH REFERENCE POINT SCATTERING

In the basic protocol, the nodes select their reference points in a totally random and independent way. Since intuitively the ideal scheduling results should end up with the scenario in which nodes in a neighborhood have short and relatively uniform working schedules, their reference points should also be distributed in a relatively uniform way. When the distance between two nodes is short and these reference points are very close to each other, the schedules may be longer than necessary. Therefore, the randomness of choosing the reference points may also hurt the total energy consumption.

Based on the above observations, we propose an enhanced design named "reference point scattering" to more evenly scatter the reference points. The idea of reference point scattering is rather simple. At the initialization phase, each node selects its reference point based on the reference points of other nodes in its circular neighborhood with radius R_{rs} that have already broadcasted their beacons. For instance, if a node has received the reference points from k other nodes: Ref_1, Ref_2, $\ldots$, and Ref_k, sorted in ascending order. Then, a period $[Ref_1, Ref_1 + T)$ is divided by the reference points into k gaps $[Ref_1, Ref_2)$, $\ldots$, $[Ref_k, Ref_1 + T)$. The node chooses the maximum gap among them and selects the middle point as its own reference point and broadcasts it.

One issue here is the selection the radius of neighborhood R_{rs}. We show the impact of R_{rs} on performance through simulations in Section 8.

7. BASELINES FOR COMPARISON

To enable a detailed evaluation, theoretical baselines and baselines from recent publications [Tian and Georganas 2003; Ye et al. 2002] are considered in our evaluation.

7.1 Theoretical Upper Bound and Lower Bound

Here we give the upper and lower bound of energy consumption per unit time for 100% coverage of a target area. We use them as baselines. The upper bound is trivial and happens when all the sensor nodes are working all the time. For different hardware platforms, the energy consumed by a working node per unit time varies significantly. For generality, we denote it as one unit of energy consumption U_e. Thus, the upper bound of energy consumption by a sensor network composed of N nodes is trivially NU_e. The lower bound of energy consumption of a sensor network differs according to the deployment of the sensor nodes. However, we are able to obtain a theoretical lower bound, although it may not be reachable by a certain sensor network. The lower bound is based on the following theorem [Williams 1979].

Theorem: To cover a unit area with equivalent circles, the lower bound of the number of circles used is $2/\sqrt{27}r^2$. The corresponding lower bound for energy consumption is $2AU_e/\sqrt{27}r^2$, where A is the size of the target area. Details of this theorem can be found in Williams [1979].

7.2 Comparison to Sponsored Coverage Scheme

Tian and Georganas [2003] and Ye et al. [2002] are the first works to address the sensing coverage problem in sensor networks. Ye et al. [2002] proposes a mechanism based on probing for energy-efficient sensing coverage for sensor networks. In the mechanism, sleeping nodes wake up according to a dynamically changing wakeup rate. When a node wakes up, it broadcasts a probing message. If there is any working node in a radius of r centered at the wakeup node, the working node broadcasts a reply and the awakened node receives it. If the awakened node does not receive any reply in a certain amount of time, the node switches into working mode. The node in working mode continuously senses its sensing area until it fails or runs out of energy. Tian and Georganas [2003] shows that such a probing-based mechanism is not able to ensure the original sensing coverage and avoid blind holes. It also points out that although by decreasing the probing area, the probing-based mechanism can minimize the area of blind holes; it keeps more nodes working, compared to the sponsor coverage-based scheme in Tian and Georganas [2003]. Hence, more energy consumption is required in Ye et al. [2002]. Therefore, in the evaluation section, we only compare our protocol with Tian and Georganas [2003].

Tian and Georganas [2003] is referred to as the sponsored coverage scheme in this paper. It presents a node-scheduling mechanism based on an eligibility rule. At the beginning of each round, each node calculates its eligibility for going to sleep. A back-off mechanism is used to avoid sensing holes caused by simultaneous actions of multiple nodes. Our approach has two advantages over this one. First, the sensor nodes only need to send a few messages for scheduling in the initialization phase, while the scheme in the paper by Tian and Georganas [2003] requires broadcasting to the neighbors at the beginning of each round. Although the exact results depend on the system settings, it can be easily concluded that the per round-based broadcast messages cause more overhead than ours. Second, when calculating eligibility for going to sleep, the
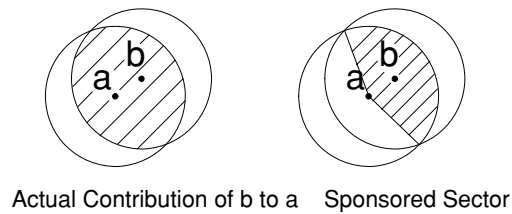
Actual Contribution of b to a   Sponsored Sector

Fig. 10.   Understanding of sensor coverage by neighboring nodes.

contribution of a neighbor node's sensing area to the sensing area of the node itself is simplified to a "sponsored sector," which is much smaller than the actual contribution, especially when the neighbor node is very close to the node itself (Figure 10). Thus, compared with our protocol, theirs need more nodes to be awake; therefore, more energy consumption occurs.

8. EVALUATION

In this evaluation section, we demonstrate the improved performance generated by our scheme in terms of (1) total amount of energy consumed, (2) energy variation among nodes, (3) half-life of the network, and (4) sensing coverage over time. Moreover, we also demonstrate how we optimize the performance by considering the target size and unbalanced initial power that are not supported by previous schemes.

In the evaluation, we do not include communication cost resulting from data transfer because it is highly application specific. Also for some applications as intruder detection, data transfer only occurs when some rare events are issued. For such systems, energy spent on data transfer is relatively insignificant.

8.1 Simulation Configuration

We run our basic protocol and extensions on a special purpose simulator. In our simulation, the sensor nodes are distributed in a 160 m × 160 m square field. The sensing range is 10 m and the communication range is 25 m. The sensor nodes are deployed with a uniform distribution into the square field, unless otherwise stated. For our protocol, the target area is the 140 m × 140 m square in the center of the square field to prevent the nodes at the edge from working all the time. We only do statistics on the central 100 m × 100 m field to eliminate the edge effect. All experiments are repeated ten times with different random seeds and different node deployments. The 95% confidence intervals of the results are about 5–10% of the means.

8.2 Total Energy Conservation for the Basic Design

In this experiment, we investigate the energy conservation performance in terms of total energy consumed per unit of time with the energy drain rate of one unit per round, if the sensor node is awake all the time. We collect results from our basic design, the second-pass optimization of our design and Tian and Georganas's [2003] sponsored coverage scheme. We also compare the simulation results with the lower and the upper bounds.
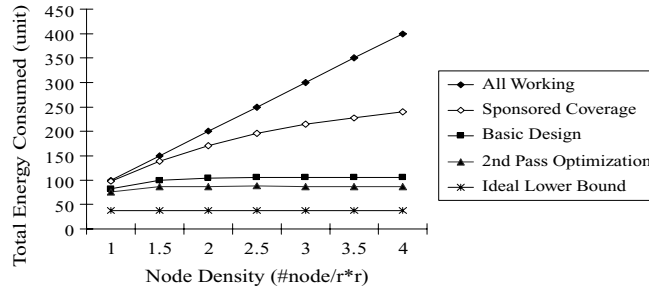
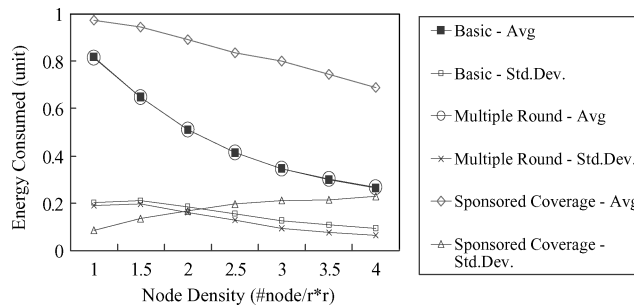Fig. 11.   Total energy consumption per unit of time.

Fig. 12.   Single node energy consumption: Average and standard deviation.

From Figure 11, we can see that our protocol consumes much less energy than Tian and Georganas's [2003] sponsored coverage scheme. This is because of the fact that in the sponsored-coverage scheme, the contribution of one working node to another node's sensing coverage is not fully considered. To be more specific, the sponsored coverage scheme is much less than the actual contribution (see Section 7.2). While in our protocol, the full contribution is considered so the redundancy is reduced. Another distinguished feature that our protocol provides is that with the increase of node density, total energy consumed by our protocol changes very little. In contrast, the sponsored-coverage scheme suffers when density increases. This is because the closer nodes are to each other, the more severe is an underestimation of sensor coverage by neighboring nodes. As a result, our scheme outperforms sponsored coverage scheme by as much as a 50% reduction in energy consumption when the density is about three nodes per r^2. Our scheme ensures that the system lifetime increases nearly proportionally with the node density of the sensor network, which is desirable for long lifetime surveillance applications.

8.3 Balancing the Energy Consumption

In this simulation, we investigate performance of energy balance. We measure the standard deviation of energy consumed by each node in our basic design and in the multiple round extension with M = 10. Results are compared with the sponsored coverage scheme.

Figure 12 shows that in both the sponsored-coverage and our scheme, the average energy consumption for a single node reduces when the network
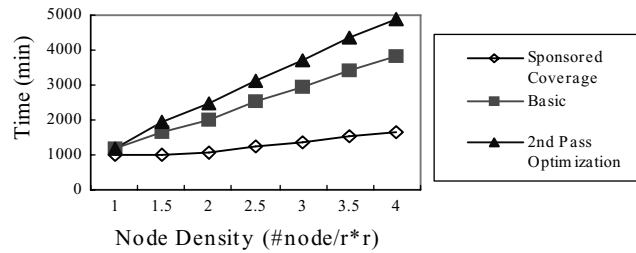
Fig. 13.   Half-life of the network.

density increases. However, our scheme reduces at a much faster rate than the sector-based scheme. When the node density is reasonably high (> 2 node/r^2), our protocol outperforms the sponsored-coverage scheme with regard to energy consumption balance among nodes. From Figure 12, we can also see that the multiple round extension can effectively reduce a certain amount of the variation in energy consumption.

8.4 Half-Life of the Network for the Basic Design

In order to measure the system lifetime extension because of our protocol, we define the half-life of a sensor network as the time from the beginning of the deployment until exactly one-half of the sensor nodes are still alive. We assume that the lifetime of a node when always working is 1000 min. The round duration we used in simulation is 10 min.

We see from the Figure 13 that the distinguishing feature of our approach is that the system half-life increases nearly linearly as the node density increases, while the sponsored-coverage scheme increases slowly when the node density increases. For example, our approach increases the half-life of the network by 130% when node density is 4 per r^2. There are two reasons contributing to this phenomenon. First, the sponsored coverage scheme consumes more energy, on average, than our approach. Second, the standard deviation among nodes in the sponsored coverage scheme increases significantly when the node density increases, as shown in Figure 13. This causes some nodes dying faster than others.

8.5 Coverage Over Time for the Basic Design

One of the most important things that the designers of sensor networks are interested in is, after a certain amount of time from when the network is deployed, how much of the target area is covered by the working sensors. In order to answer this question, we simulate the percentage of coverage after certain time intervals. The percentage of coverage takes care of the coverage in both time and space. For each grid point in the target area, we choose several sampling times during a round. If for some (grid point, time) pair, the grid point is sensed by some working node, we call this pair to be "valid." We count the ratio of the valid pairs among all the pairs with both our proposed protocol and the sponsored-coverage scheme described in Tian and Georganas [2003]. Different curves shown in Figure 14 are obtained under different node density. The
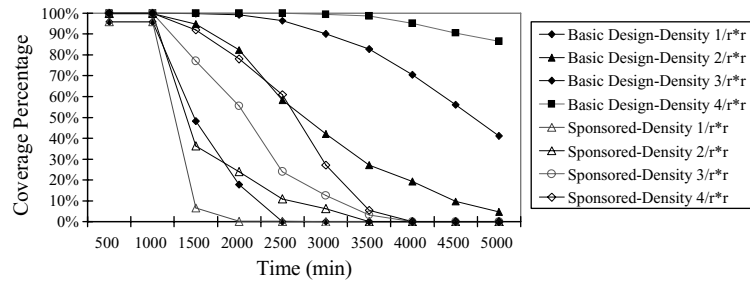
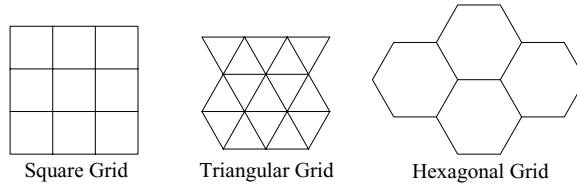Fig. 14.   Sensing coverage over time.

Fig. 15.   Three types of grids.

curves with the hollow markers are the results from the sponsored-coverage scheme and the curves with the solid markers are the results from our scheme.

From the simulation results, we see that over time with the same node density, the target area is covered more with our protocol than with the sponsored coverage scheme. For example, at density of 2 per r^2, it take about 5000 min for our scheme to degrade to zero-sensing coverage, while the sponsored-coverage scheme degrades to zero-sensing coverage in just 3500 min.

8.6 Comparison with Regular Deployments

The above simulation results show the efficiency of the basic protocol when the nodes are deployed with a uniform random distribution. In this case, the variation among energy consumption is because of two random factors. First, the random selection of reference points contributes to the difference of energy consumption among nodes. Second, the randomness in the deployment is another reason of the variance. It is, therefore, interesting to explore through simulations how the latter factor, i.e., the deployment randomness, affects the energy consumption variance among nodes.

We lay out the nodes based on three regular grid types, namely, square, triangular, and hexagonal grid, as shown in Figure 15. For each density and grid type, we calculate the corresponding distance between two neighboring grid points and use it for simulation deployments. The means and standard deviations of energy consumption are obtained from the simulations and shown in Figure 16.

From Figure 16, we can see that the average energy consumption for all deployments differ a small amount. This is because regardless of the deployment type, sensors need to cover the whole area. The only exception is for the triangular grid deployment when the density is 1 node per r^2. The reason is that the
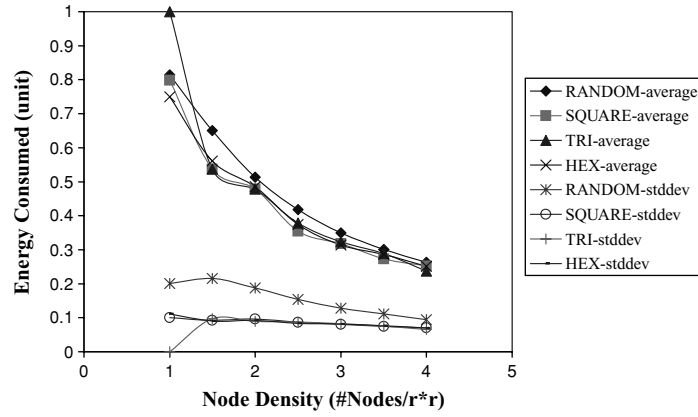
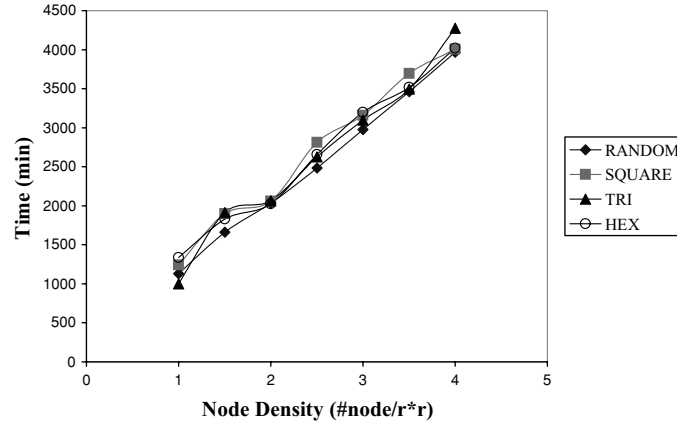Fig. 16.　Energy consumed for grid deployments.

Fig. 17.　Half-life for grid deployments.

distance between two neighboring nodes is $\sqrt{2\sqrt{3}}r$, which is greater than the sensing range r. Each node has to be awake all the time to guarantee the full sensing coverage, because the small area around a node can only be covered by the node itself. On the other hand, the standard deviations of the energy consumption for regular deployments are significantly smaller than those for random deployments. This result indicates that the randomness in deployment significantly affects the variance of energy consumption among nodes. Another interesting observation is that both the average values and standard deviations of energy consumption are very close for all three kinds of grid deployments. The only exception is for triangular deployment when the node density is 1 per r^2. With the density increasing, the difference of the standard deviations between random deployments and regular deployments are decreasing.

We also compare the half-lives of the networks for random and regular deployments. The results are shown in Figure 17. From the figure, we can see that half-life differs insignificantly under four types of deployment.
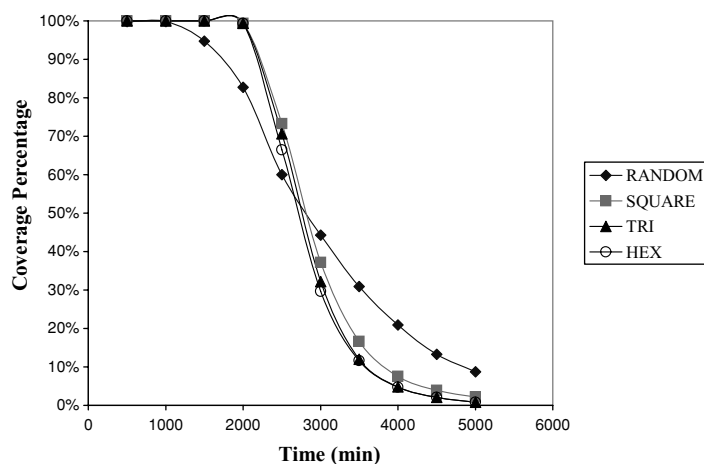
Fig. 18.   Coverage over time for grid deployments.
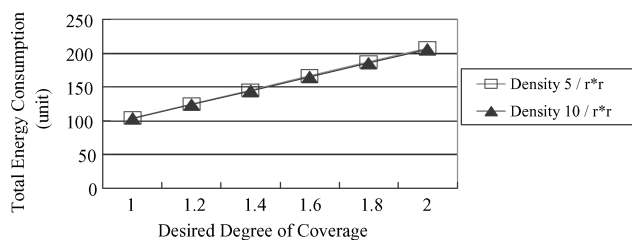
Fig. 19.   Total energy consumed for differentiated surveillance.

We compare sensing coverage over time for the various deployments. The node density we use is 2 per r^2. The results are shown in Figure 18. From Figure 17 we know that at this density all four kinds of deployment have almost the same system half-live, about 2000 min. Figure 18 is very interesting: compared with the random deployment. Regular grid deployments give a significantly higher sensing coverage (almost 100%) before the system half-life. After the system half-life, the coverage under regular deployments decreases much faster than that under random deployment. After 3000 min, the coverage provided by regular deployments are smaller than that provided by the random deployment. The reason is that for regular deployments, the standard deviation of energy consumption among nodes is significantly smaller than the random deployment case. Therefore, for regular deployments, the time when nodes are depleted of energy is more uniform, which causes more nodes to run out of energy soon after the system half-life.

8.7 Total Energy Consumed for Differentiated Surveillance

Unlike previous schemes, our basic design can be easily extended to provide differentiated surveillance by proportionally increasing the T_{front} and T_{end} interval values. Figure 19 shows the simulation results of differentiated coverage with two different node densities.
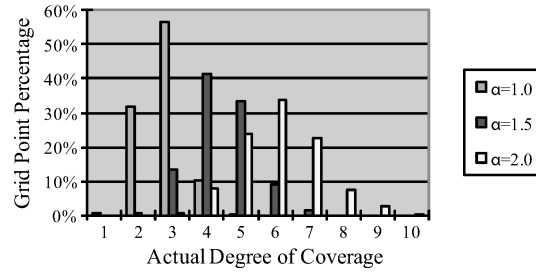
Fig. 20. Actual degree of coverage for differentiated surveillance.

From the results, we conclude that (1) the energy consumed per unit time increases linearly when the parameter of desired degree of coverage α increases and (2) the energy consumed under different node densities changes very little when the node densities are high enough. This is shown in Figure 19, where the two curves nearly overlap each other. This independence between the physical node density and actual energy consumed is quite desirable.

8.8 Actual Degree of Coverage for Differentiated Surveillance

In this simulation, we set the node density to be 5 per r^2 and sample the actual degree of coverage for 100×100 grid points in the center. In addition to the proof from Section 4, Figure 20 shows that our scheme can guarantee a degree of coverage to the desired value α for the target area. For example, if α is 2, then we guarantee that all grid points are covered by more than two sensor nodes and a single failure node does not affect the full coverage. However, the guaranteed α is the minimum possible degree of coverage for each grid point. Because of the protocol's redundancy, most grid points are covered by a higher degree than the desired degree of coverage α. Figure 20 shows the distribution of actual degree of coverage for the grid points when we apply different desired degrees of coverage α. For example, when $\alpha = 2$, we guarantee that no grid point has an actual degree of coverage of less than 2. We acknowledge that the current solution only guarantees a sensor coverage lower bound and future work will continue to investigate the possibility to provide a sensor coverage upper bound for better energy conservation.

8.9 Energy Consumption for Considering Target Size and Shape

In this experiment, we study the improved energy conservation when considering the target sizes and shapes. Figure 21 shows the simulation results of total energy consumed per unit time for four different node densities. Figure 22 denotes the average energy consumption and standard deviation for a single node. Figure 23 demonstrates the impact of target sizes on total energy consumption.

From the results, we can conclude that this enhanced scheme outperforms second-pass optimization for all node densities and, as node density increases, the ratio of conserved energy also increases. At node density of 4 nodes per r^2 and target size of 2, the total energy consumed and average node energy consumption of enhanced scheme are less than 81% of the second-pass optimization
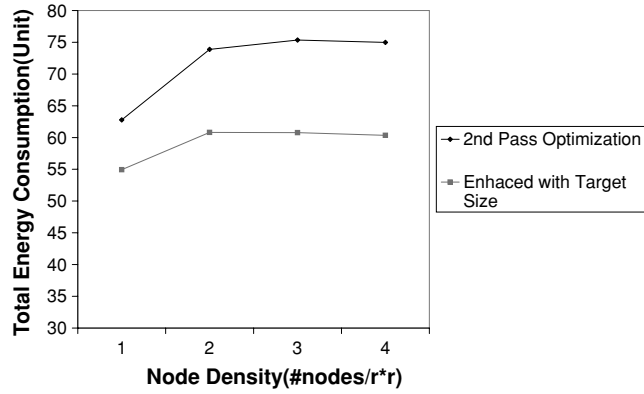
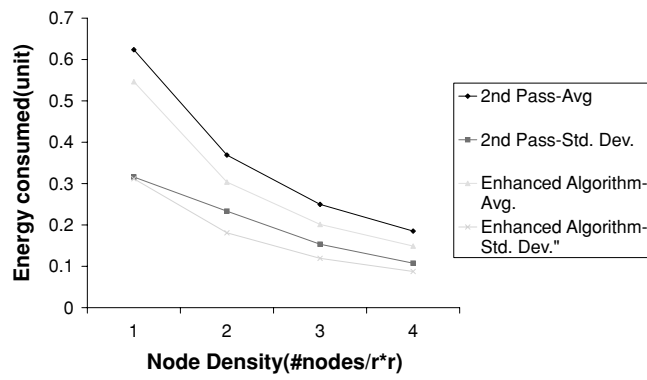Fig. 21.   Total energy consumption per unit of time (target size = 2).

Fig. 22.   Single node energy consumption: Average and standard deviation (target size = 2).
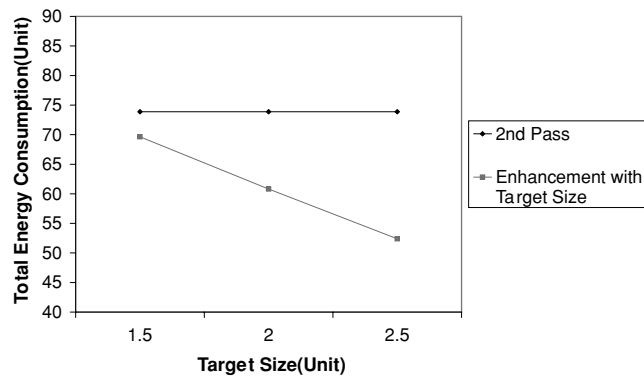

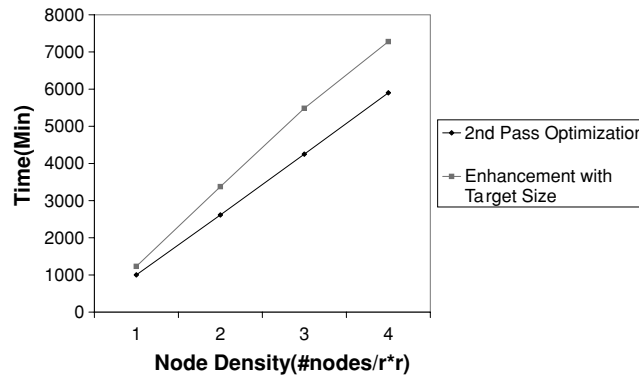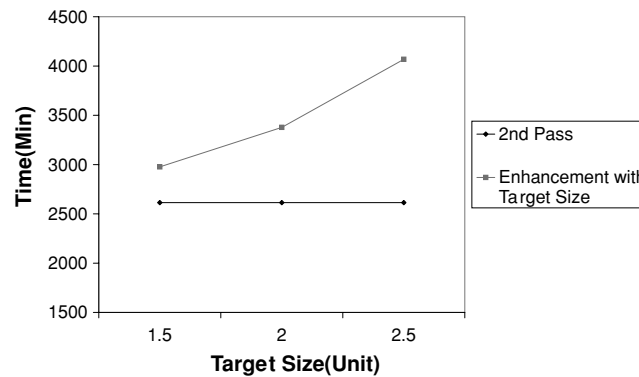Fig. 23.   Total energy consumption versus target size (node density = $2/r * r$).

Fig. 24.　Half-life of the network (target size = 2).

Fig. 25.　Half-life of the network versus target size (node density = $2/r*r$).

scheme, while the standard deviation for single node energy consumption also reduces. As target size increases, the total energy consumption drops almost linearly. The reason for these phenomena is that the new scheme increases grid size based on target size and virtually extends the sensing range. Therefore, as node density increases, the chance that a single node has more neighboring nodes increases dramatically and, consequently, leads to less per-node and total energy consumption.

8.10 Half-Life of the Network for Considering Target Size and Shape

To further measure system lifetime extension with the scheme that assigns grid size based on target size and shape, we compared the half-life of enhanced scheme with original second-pass optimization scheme under different node densities.

From Figure 24, we can see that the half-life of both new and original algorithm grow almost linearly, but with different slopes. The new scheme has a relatively larger slope, which implies a longer half-life for all node densities. Moreover, the rise of target size is also accompanied by the linear improvement on system half-life, which is demonstrated in Figure 25. In the simulation, at
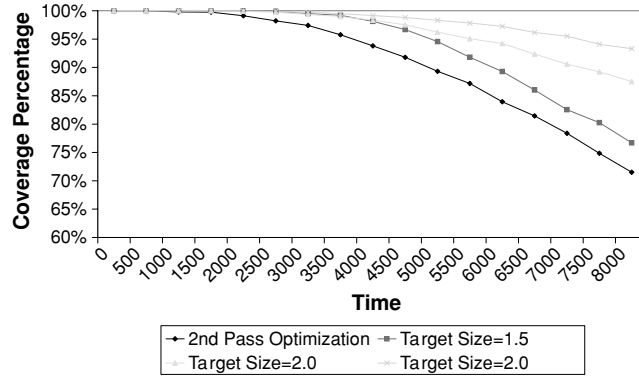
Fig. 26.   Sensing coverage over time $(3/r * r)$.

target size of 2, the new scheme has approximately 1.2 times longer half-life than the second-pass optimization scheme, because of the reduced average node energy consumption.

8.11 Coverage Over Time for Considering Target Size and Shape

In real applications, the coverage over time is one of the most important issues that designers of sensor networks need to consider. In Figure 26, we show the coverage over time at node density of 3.

The figure shows that the new scheme always has better coverage than the original design under node density of 3 (We get similar results under the node densities of 1, 2, and 4). During the whole network lifetime, the gap between the coverage in the new algorithm and the original algorithm is increasing over time. This is because of the relatively lower standard deviation for average node energy consumption in the enhanced algorithm.

8.12 Energy Consumption for Reference Point Scattering

In this subsection we simulate the average values and standard deviations of energy consumption of nodes with various densities when the reference point scattering enhancement is applied. In the simulation, we choose various values for the neighborhood radius R_{rs} and explore the impact of the R_{rs} values on the performance. We compare the results in Figure 27 with the basic design performance. From the figure, we can see that the average energy consumption when $R_{rs} = 0.5r$ is lower than that of the basic design. The average value is even lower when $R_{rs} = 1.0r$. However, when $R_{rs} = 1.5r$, the average energy consumption increases and when $R_{rs} = 2.0r$, the average value is greater than that of the basic design. This shows that when $R_{rs} = 1.0r$, the average energy consumption reaches its optimum value among those with other R_{rs} values and the basic design. The reason that $R_{rs} = 2.0r$ does not give a better result is that when the distance between two nodes is close to two times the sensing range. The overlapping coverage area is very small and, therefore, it is not helpful to consider one reference point for the decision of another. The improvement on the variance of the energy consumption is not significant.
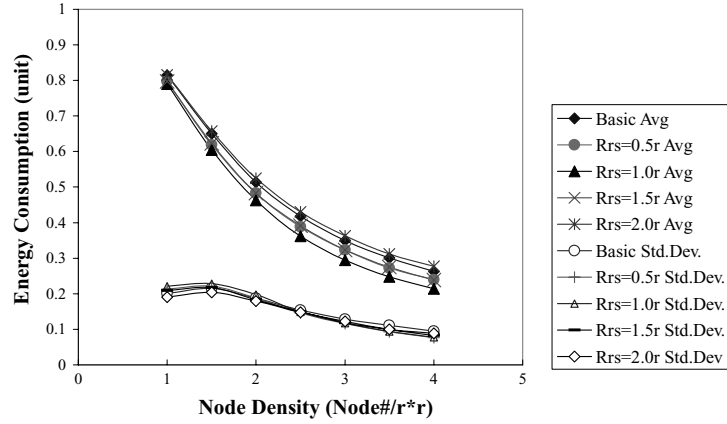
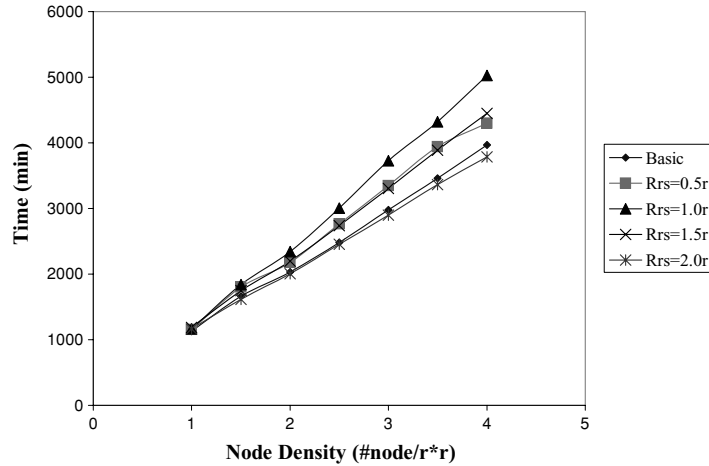Fig. 27. Total energy consumed for reference point scattering.

Fig. 28. Half-life for reference point scattering.

8.13 Half-Life for Reference Point Scattering

With the same simulation setting as the above, we obtain the half-lives of the system. From the results shown in Figure 28, we can see that when $R_{rs} = 1.0r$, the system half-life increases over that of the basic design by as much as 25%. When $R_{rs} = 0.5r$ and $R_{rs} = 1.5r$, there is still a gain in the system half-life, but not as much as that of $R_{rs} = 2.0r$; the system half-life is shorter than that of the basic design. These results are consistent with the results of simulations for the average energy consumption.

8.14 Coverage Over Time for Reference Point Scattering

In this set of simulations, we choose the node density as 2 per r^2 and the other settings are the same as the above. We obtain the sensing coverage over time and show the results in Figure 29. From the figure, we can see that coverage for the case that $R_{rs} = 1.0r$ is significantly greater than that of the basic design.
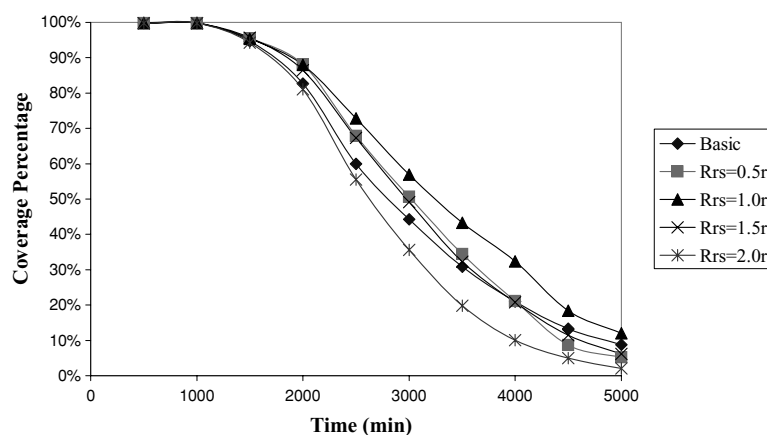
Fig. 29.   Coverage over time for reference point scattering.
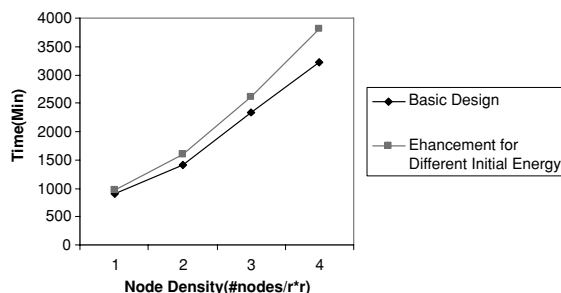

Fig. 30.   Half-life of the network.

Coverage for the cases that $R_{rs} = 0.5r$ and $R_{rs} = 1.5r$ is better than that of the basic design before 4000 min, but worse after that. When $R_{rs} = 2.0r$, the coverage is always worse than that of the basic design. Again, these results are consistent with the results of simulations for the average energy consumption.

From the above three sets of simulations, we can conclude that $1.0r$, or the sensing range itself, is a good choice for R_{rs} when reference point scattering is applied.

9. PRACTICAL ISSUES

In this section, we study the impact of several practical issues caused by different initial node energy, localization errors, irregular sensing ranges, irregular communication ranges, and node failures.

9.1 Impact of Different Initial Node Energy

In this experiment, we randomly assign one-half of the nodes with the initial energy of a 500-min life-time capacity and the other one-half with a 1000-min life-time capacity.

As shown in Figure 30, when the system has different initial energy, the enhancement proposed in Section 3.5.3 contributes a longer half-life of the net-
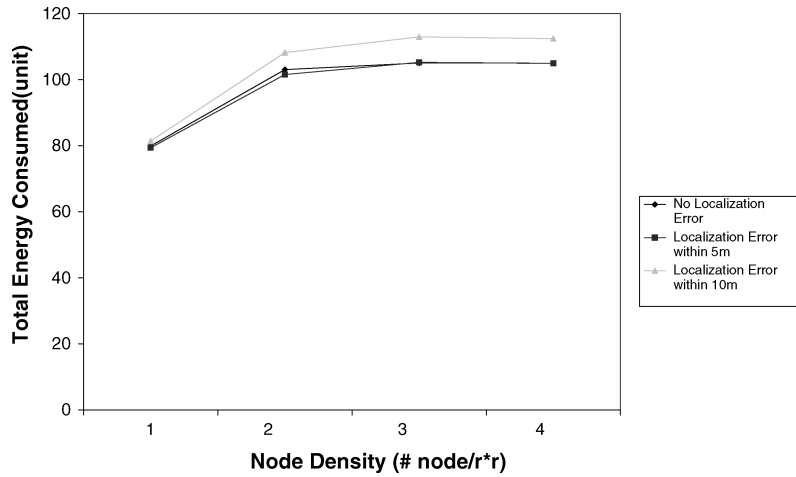
Fig. 31.   Total energy consumption per unit of time.

work, compared with the basic design under all node densities. As node density increases, the benefits gained by the enhancement become more significant. For instance, at the node density of 4 per r^2, the enhancement outperforms the basic design by 18% on the network half-life. The reason for this result is that by optimizing reference point locations and modifying the expressions for calculating T_{front} and T_{end}, the reference points for nodes with high-energy capacity now are more likely adjacent to the reference points of the nodes with low-energy capacity. Therefore, the enhancement effectively balances the energy among nodes, regardless of initial energy distribution.

9.2 Impact of Localization Errors

In this experiment, we investigate the impact of localization errors on the energy consumption and sensing coverage for our proposed schemes. Figure 31 shows the comparison of total energy consumption per unit of time for the scenario with no, local, 5-m, and 10-m localization error under different node densities. From the figure, we can see within 5-m localization error, the energy consumption differs little from the case with no localization error. Since many existing localization schemes can provide meter-level localization accuracy, we believe our sensing coverage algorithm is compatible with existing localization schemes.

In addition, in Figure 32 shows the coverage over time for cases with no, 5-m, and 10-m localization error under different node densities. From Figure 32, we can see even for the localization error of 10 m, the coverage over time is almost identical to the case when there is no localization error. According to the above results, we conclude that our proposed schemes are very robust under moderate localization errors.
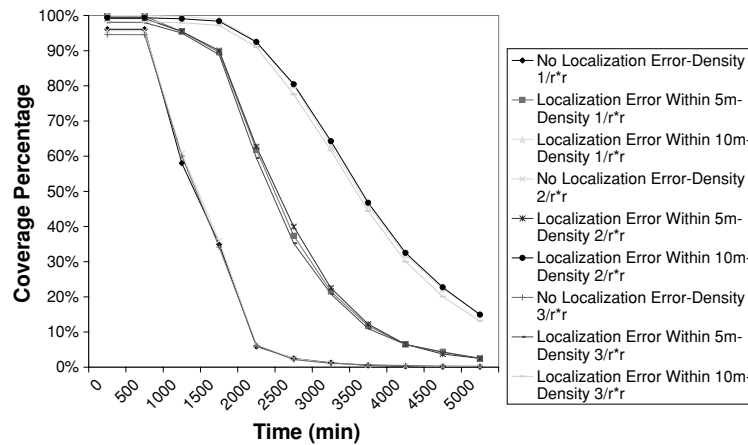
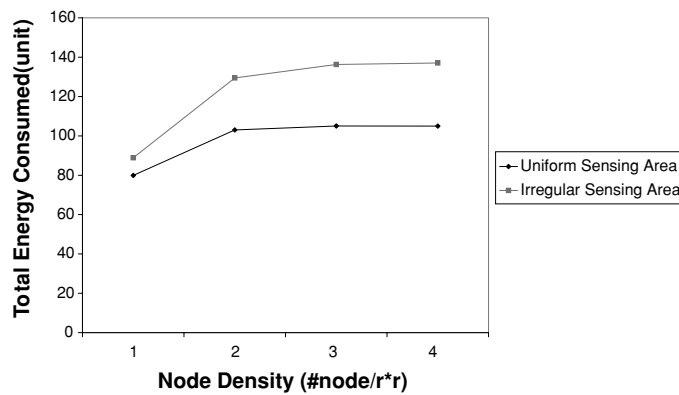Fig. 32.   Sensing coverage over time.

Fig. 33.   Total energy consumption per unit of time.

9.3 Impact of Irregular Sensing Range

In order to understand the impact of the irregular sensing range, we model the single-node sensing area as circular sensing area of 10 m radius, with random unsensed points inside. Figure 33 shows total energy consumption per unit of time for cases of the regular and the irregular sensing range. From Figure 33, we can observe about a 30% increase in total energy consumption for the case of the irregular sensing range. This is because in the irregular sensing range case, the average area sensed by a single sensor node is decreased. Consequently, the number of sensor nodes covering a single grid point reduces, leading to higher energy consumption. Figure 34 displays the coverage over time for the regular and irregular sensing range cases. Consistent with energy consumption results, the nodes with irregular sensing ranges run out of energy earlier than the nodes that have regular sensing ranges.
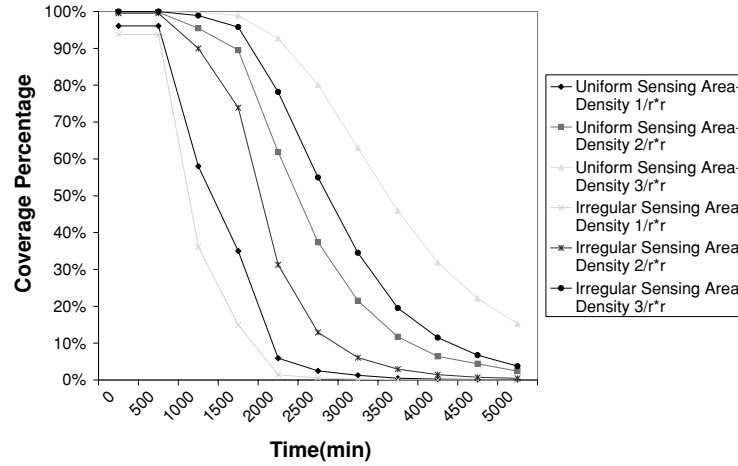
Fig. 34.   Sensing coverage over time.

9.4 Impact of Irregular Communication Range

In the process of setting up nodes schedules, the quality of wireless links among sensor nodes also impacts coverage performance. In this experiment, we use the TOSSIM lossy link model to study the impact of irregularity of wireless communication. Similar to the case of the irregular sensing range, Figure 35 shows the total energy consumption for the irregular communication case increases approximately 15% over the regular communication case. This is because of the packet loss during the reference point broadcasting phase, which potentially reduces the number of known neighbors for each single sensor node. Consequently, it introduces higher energy consumption since the number of nodes used to calculate T_{front} or T_{end} for each covered grid point is decreased. With such increased energy consumption, we also observe similar trend of irregular sensing range for coverage over time in Figure 36, the nodes with irregular communication ranges deplete their energy faster than the nodes with perfect communication, leading to a worse quality of coverage over time. According to this observation, in order to minimize the impact of irregular communication range, during the phase of broadcasting reference points, we suggest increasing the number of retransmissions of reference point packets, in order to reduce the chance that a node is unaware of its existing neighbors.

9.5 Impact of Node Failures

In this experiment, we study the performance of the proposed failure tolerance scheme with a realistic lossy link model. First, Figure 37 shows the half-life of network with node failure rate up to 20% of total deployed sensor nodes. From Figure 37, we can see the system half-life drops slowly as the node failure rate increases. For example, when there are 20% of nodes failures, at node density of 3, the half-life decreases about 15% compared to the network without faulty nodes. Such a result indicates that the proposed fault-tolerance scheme is robust enough to fight against the scenario when there is a large portion of
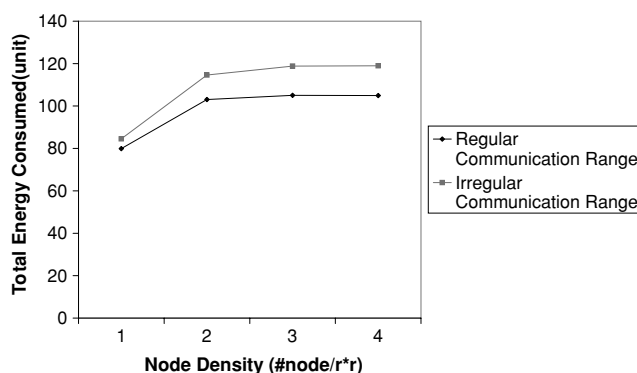
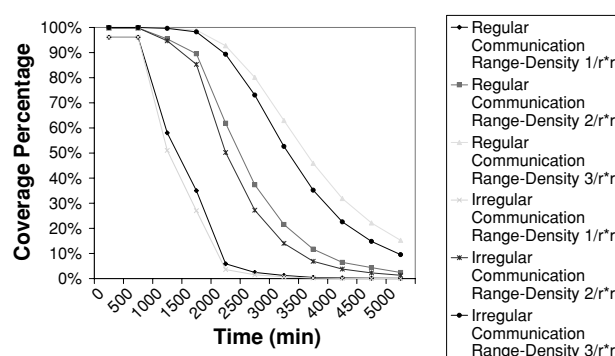Fig. 35. Total energy consumption per unit of time.

Fig. 36. Sensing coverage over time.

node failures during the operation.

Figure 38 shows the coverage from 0 to 20% failure node case under different node densities. From Figure 38, we can see for node density above one, our proposed fault-tolerance scheme effectively recovers the coverage of the network. We note that there is an obvious drop of coverage in node density of one, which is because the original network node density is so low that many grid points are covered by only a single node. If this single node happens to be faulty, no other node can physically take over the coverage, consequently leading to the blind points with no coverage.

10. CONCLUSIONS

In this paper, we introduce an adaptable sensing coverage mechanism for differentiated surveillance in sensor networks. Unlike previous schemes, our scheme guarantees not only full sensing coverage to a certain geographic area, but also the degree of coverage up to the limit imposed by the density of nodes available. The distinguishing advantages of our scheme are a much longer network half-life through energy conservation and balancing among sensor nodes, and a small communication overhead required to establish a working duty sched-
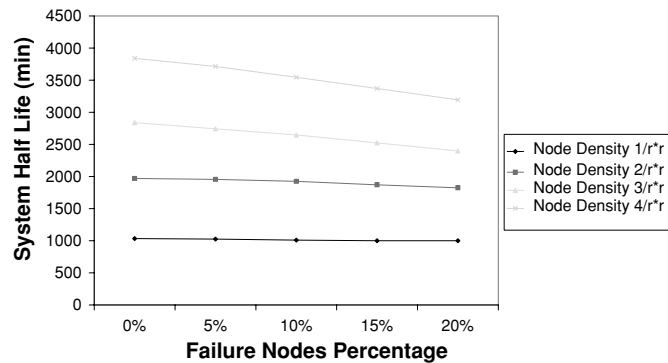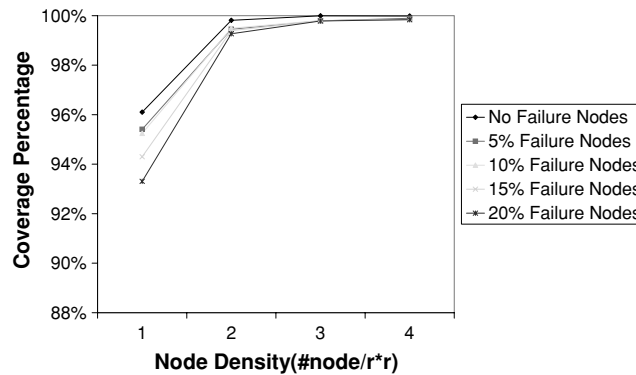
Fig. 37.   Half-life of the network.

Fig. 38.   Sensing coverage.

ule among nodes. Several optimizations and extensions are proposed to provide even better performance in terms of network lifetime and energy balance. In addition, with the enhancement considering the target size and shape, we could gain more benefits in energy conservation. With the enhancement considering the initial energy discrepancy among nodes, we achieve a better energy balance among sensor nodes and, hence, a longer network half-life. Simulations show that our protocol accomplishes differentiated surveillance with low-energy consumption and significantly outperforms other state-of-art schemes in terms of energy consumption and half-life of the network.

REFERENCES

AHN, G.-S., CAMPBELL, A. T., VERES, A., AND SUN, L.-H. 2002. SWAN: Service differentiation in stateless wireless ad hoc networks. In *IEEE INFOCOM*.

ALT, H., HSU, D., AND SNOEYINK, J. 1995. Computing the largest inscribed isothetic rectangle. In *Proceeding of 7th Canadian Conference on Computational Geometry* 67–72.

BHATNAGAR, S., DEB, B. R., AND NATH, B. 2001. Service differentiation in sensor networks. In *International Symposium on Wireless Personal Multimedia Communications*.

BHATTACHARYA, S., KIM, H., PRABH, S., AND ABDELZAHER, T. 2003. Energy-conserving data placement and asynchronous multicast in wireless sensor networks. In *The First International Conference*

on Mobile Systems, Applications, and Services (MobiSys).

CAO, Q., YAN, T., STANKOVIC, J. A., AND ABDELZAHER, T. F. 2005. Analysis of target detection performance for wireless sensor networks. In *International Conference on Distributed Computing in Sensor Networks (DCOSS)*.

CERPA, A. AND ESTRIN, D. 2002. ASCENT: Adaptive self-configuring sensor networks topologies. In *Proceedings of the IEEE Computer and Communications Societies (INFOCOM)*.

CHEN, B., JAMIESON, K., BALAKRISHNAN, H., AND MORRIS, R. 2001. Span: An energy-efficient coordination algorithm for topology maintenance in ad hoc wireless networks. In *6th ACM MOBICOM Conference*.

CrossBow Technology, Inc. CrossBow Technology, Inc. Available at `http://www.xbow.com/Products/Product_pdf_files/wireless_pdf/6020-0042-0%1_A_MICA2.pdf`.

DANIELS, K., MILENKOVIC, V., AND ROTH, D. 1997. Finding the largest area axis-parallel rectangle in a polygon. *Computational Geometry: Theory and Applications*, 125–148.

ELSON, J., GIROD, L., AND ESTRIN, D. 2002. Fine-grained network time synchronization using reference broadcasts. In *Symposium on Operating Systems Design and Implementation*.

GU, L. AND STANKOVIC, J. A. 2004. Radio-triggered wake-up capability for sensor networks. In *Proceedings of RTAS*.

GUI, C. AND MOHAPATRA, P. 2004. Power conservation and quality of surveillance in target tracking sensor networks. In *MobiCom*.

GUO, C., ZHONG, L. C., AND RABAEY, J. M. 2001. Low power distributed MAC for ad hoc sensor radio networks. In *IEEE GlobeCom*.

HE, T., BLUM, B. M., STANKOVIC, J. A., AND ABDELZAHER, T. F. 2004a. AIDA: Adaptive application independent data aggregation in wireless sensor networks. *ACM Transactions on Embedded Computing System, Special Issue on Dynamically Adaptable Embedded Systems*.

HE, T., HUANG, C., BLUM, B. M., STANKOVIC, J. A., AND ABDELZAHER, T. 2003a. Range-free localization schemes in large-scale sensor networks. In *Proceedings of the International Conference on Mobile Computing and Networking (MOBICOM)*.

HE, T., KRISHNAMURTHY, S., STANKOVIC, J. A., AND ABDELZAHER, T. 2004b. An energy-efficient surveillance system using wireless sensor networks. In *The Second International Conference on Mobile Systems, Applications, and Services (MobiSys)*.

HE, T., STANKOVIC, J., LU, C., AND ABDELZAHER, T. 2003b. SPEED: A stateless protocol for real-time communication in ad hoc sensor networks. In *Proceedings of International Conference on Distributed Computing Systems (ICDCS)*.

HE, T., VICAIRE, P., YAN, T., CAO, Q., ZHOU, G., GU, L., LUO, L., STOLERU, R., STANKOVIC, J. A., AND ABDELZAHER, T. 2006. Achieving long-term surveillance in VigilNet. In *IEEE Infocom*.

HEINZELMAN, W. R., CHANDRAKASAN, A., AND BALAKRISHNAN, H. 2000. Energy-efficient communication protocol for wireless microsensor networks. In *Proceedings of the International Conference on System Sciences*.

HSIN, C.-F. AND LIU, M. 2004. Network coverage using low duty-cycle sensors: Random & coordinated sleep algorithms. In *3rd International Symposium on Information Processing in Sensor Networks (IPSN'04)*, Berkeley, California.

H.TAKAGI AND L.KLEINROCK. 1984. Optimal transmission ranges for randomly distributed packet radio terminals. *IEEE Trans. Commun. 32,* 3.

INTANAGONWIWAT, C., ESTRIN, D., GOVINDAN, R., AND HEIDEMANN, J. 2002. Impact of network density on data aggregation in wireless sensor networks. In *International Conference on Distributed Computing Systems*.

KIRKPATRICK, D. AND SNOEYINK, J. 1995. Tentative prune-and-search for computing fixed-points with applications to geometric computation. *Fundamental Informatic*. 353–370.

KRISHNAMACHARI, B., ESTRIN, D., AND WICKER, S. 2002. Impact of data aggregation in wireless sensor networks. In *Proceedings of the International Workshop on Distributed Event-Based Systems*.

LU, C., B. M. B., ABDELZAHER, T. F., STANKOVIC, J. A., AND HE, T. 2002. RAP: A real-time communication architecture for large-scale wireless sensor networks. In *IEEE RTAS*.

MIN, R., BHARDWAJ, M., CHO, S.-H., SINHA, A., SHIH, E., WANG, A., AND CHANDRAKASAN, A. 2000. An architecture for a power-aware distributed microsensor node. In *IEEE Workshop on Signal Processing Systems*.

RAMANATHAN, R. AND ROSALES-HAIN, R. 2000. Topology control of multihop wireless networks using transmit power adjustment. In *IEEE INFOCOM*.

REN, S., LI, Q., WANG, H., CHEN, X., AND ZHANG, X. 2005. Analyzing object tracking quality under probabilistic coverage in sensor networks. *ACM Mobile Computing and Communications Rev. 9,* 1.

SZEWCZYK, R., MAINWARING, A., ANDERSON, J., AND CULLER, D. 2004. An analysis of a large scale habit monitoring application. In *SenSys'04*.

TIAN, D. AND GEORGANAS, N. 2003. A node scheduling scheme for energy conservation in large wireless sensor networks. *Wireless Comm. Mobile Comput. J. 3*, 2, 271–290.

TOLLE, G., POLASTRE, J., SZEWCZYK, R., TURNER, N., TU, K., BURGESS, S., GAY, D., BUONADONNA, P., HONG, W., DAWSON, T., AND CULLER, D. 2005. A macroscope in the redwoods. In *Sensys'05*.

WILLIAMS, R. 1979. *Geometrical foundation of natural structure: A source book of design*. Dover, New York.

XING, G., LU, C., PLESS, R., AND HUANG, Q. 2006. Impact of sensing coverage on greedy geographic routing algorithms. *IEEE Transactions on Parallel and Distributed Systems, Special Issue on Localized Communication and Topology Protocols for Ad Hoc Networks*.

XING, G., LU, C., PLESS, R., AND O'SULLIVAN, J. A. 2004. Co-Grid: An efficient coverage maintenance protocol for distributed sensor networks. In *3rd International Symposium on Information Processing in Sensor Networks (IPSN'04)*.

XING, G., WANG, X., ZHANG, Y., LU, C., PLESS, R., AND GILL, C. 2005. Integrated coverage and connectivity configuration for energy conservation in sensor networks. *ACM Trans. Sensor Netw. 1,* 1.

XU, N., RANGWALA, S., CHINTALAPUDI, K. K., GANESAN, D., BROAD, A., GOVINDAN, R., AND ESTRIN, D. 2004. A wireless sensor network for structural monitoring. In *SenSys 2004*.

XU, Y., HEIDEMANN, J., AND ESTRIN, D. 2001. Geography-informed energy conservation for ad hoc routing. In *MobiCom*.

XUE, Y. AND LI, B. 2001. A location-aided power-aware routing protocol in mobile ad hoc networks. In *IEEE GlobeCom*.

YE, F., ZHONG, G., LU, S., AND ZHANG, L. 2003. PEAS: A robust energy conserving protocol for long-lived sensor networks. In *Proceedings of the International Conference on Distributed Computing Systems (ICDCS)*.

YE, F., ZHONG, G., LU, S., AND ZHANG, L. 2002. Energy efficient robust sensing coverage in large sensor networks. Tech. Rept., UCLA.